

66475

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

BANYO TASARIMI İÇİN KURAL TABANLI UZMAN SİSTEM

YÜKSEK LİSANS TEZİ

Mimar Çelik İNGİN

Tezin Enstitüye Verildiği Tarih : 3 Mart 1997

Tezin Savunulduğu Tarih : 6 Mart 1997

Tez Danışmanı : Prof. Dr. Nigan BAYAZIT

Diğer Jüri Üyeleri : Prof. Dr. Gülen ÇAĞDAŞ

Y. Doç. Dr. Alaattin KANOĞLU

M. Bayazıt

Gülen Çağdaş

Akan

28.03.97

Y.E. YÜKSEKÖĞRETİM KURULU
DOKÜMANİASYON MERKEZİ

Mart 1997

ÖNSÖZ

Problemlerin gelişim hızı bilginin üretiliş hızıyla koşut gitmektedir. Bunun sonucu olarak gittikçe karmaşıklaşan problemler tasarımcıları ve uygulayıcıları sorunların karmaşıklığıyla boy ölçüşebilecek araçların geliştirilmesine yöneltmektedir. Bu noktada tasarımda uzmanlaşma ve bilgi sistemlerinden faydalanma gündeme gelmektedir.

Sorunların çözümünde bilgisayarlardan faydalanma konusunda son yıllarda önemli gelişmeler kaydedilmiştir. Çalışmalar bilgisayarların tasarımdaki katkısını da ilerletmektedir. Bu amaçla yeni yaklaşımlar geliştirilmektedir. Bu yaklaşımlardan biri olan yapay zeka araştırmaları ve onun bir alt dalı olan uzman sistemler bu çalışmanın konusunu oluşturmaktadır. Bu çalışmada konut tasarımının bir alt dalı olan banyo tasarımı konusunda, uzman sistem yaklaşımını ve kural tabanlı çıkarım mekanizmasını kullanarak sonuca giden bir bilgisayar programı geliştirilmeye çalışılmıştır.

Her zaman yanımda olan ve beni destekleyen aileme, ve çalışmanın her aşamasında fikirlerini ve kaynaklarını esirgemeyen sayın Prof. Dr. Nigan Bayazıt'a teşekkür ediyorum.

Mimar Çelik İnçin

Mart 1997

İÇİNDEKİLER

ÖNSÖZ.....	ii
İÇİNDEKİLER	iii
ŞEKİL LİSTESİ.....	vi
TABLO LİSTESİ	viii
ÖZET.....	x
SUMMARY	xi
BÖLÜM 1 GİRİŞ.....	1
1.1 Yapay Zeka (YZ).....	1
1.2 Uzman Sistemler (US)	3
1.2.1 US Geliştirme Aşamaları	4
1.2.1.1 Tanımlama (identification) Aşaması	4
1.2.1.2 Kavramları Geliştirme (conceptualization) Aşaması	5
1.2.1.3 Biçimlendirme (formalization) Aşaması	5
1.2.1.4 Uygulama (implementation) Aşaması	5
1.2.1.5 Sınama (testing) Aşaması.....	5
1.2.2 US Tipleri.....	6
1.2.2.1 Yorum (interpretation) Sistemleri.....	6
1.2.2.2 Tahmin (prediction) Sistemleri	6
1.2.2.3 Tanı (diagnosis) Sistemleri.....	6
1.2.2.4 Tasarlama (design) Sistemleri	7
1.2.2.5 Planlama (planning) Sistemleri	7
1.2.2.6 İzleme (monitoring) Sistemleri.....	7
1.2.3 US Uygulamaları ve Geliştirme Araçları.....	8
1.2.3.1 DENDRAL.....	8
1.2.3.2 META-DENDRAL.....	9
1.2.3.3 SAINT	9
1.2.3.4 EXPERT ve CASNET	9
1.2.3.5 MYCIN, EMYCIN, PUFF, ve CADUCEUS	10
1.2.3.6 RITA ve ROSIE.....	10
1.2.5 Genel Amaçlı US Programlama Dilleri.....	10
1.2.5.1 PROLOG	11
1.2.5.2 LISP.....	12

1.3 Banyo.....	14
1.3.1 Tarihçe	14
1.3.2 Başlıca Eylemler, Elemanlar, ve Boyutları	18
1.3.2.1 Vücut Yıkama.....	18
1.3.2.2 Lavabolar.....	24
1.3.2.3 Tuvalet.....	28
1.3.2.4 Sifon ve Süzgeçler.....	32
1.3.2.5 Bataryalar	33
1.3.2.6 Aksesuarlar.....	34
1.3.2.7 Dolaplar	35
1.3.2.8 Sıcak Su	35
1.3.2.9 Isıtma	36
1.3.3 Boyutlar ve Alanlar	37
1.3.4 Kaplama Malzemeleri.....	37
1.3.4.1 Döşeme Kaplamaları	38
1.3.4.2 Duvar Kaplamaları	39
1.3.4.3 Tezgahlar.....	40
BÖLÜM 2 METOD	42
2.1 Programlama Dili ve Ortamı	42
2.1.1 AutoLISP	43
2.2 Programlama Metodu.....	45
2.3 Kural Tabanı ve Değerlendirme	54
2.3.1 Birinci Kural Bölümü	55
2.3.2 İkinci Kural Bölümü.....	57
2.3.3 Kuralların Açıklanması.....	60
2.3.3.1 Küvetle İlgili Kurallar	60
2.3.3.2 Duş Teknesiyle İlgili Kurallar	61
2.3.3.3 Çamaşır Makinesiyle İlgili Kurallar.....	63
2.3.3.4 Termosifonla İlgili Kurallar	64
2.3.3.5 Şofbenle İlgili Kurallar	64
2.3.4 Kural Uygulanması için Bir Örnek.....	65
BÖLÜM 3 YAZILIM	69
3.1 Programın Çalıştırılması.....	69
3.2 Programın Çıktıları	78
SONUÇLAR ve ÖNERİLER.....	85
KAYNAKLAR.....	87
EKLER.....	91
EK A Kombinezon Tablosu	91
EK B Programın Metni	97
EK C Programlanabilir Diyalog Kutuları	156
EK D Banyo.krl Dosyası.....	176
EK E Banyo.dta Dosyası.....	180
EK F Banyo.def Dosyası.....	184



ŞEKİL LİSTESİ

Şekil 1.1 M.Ö. 5. yy a ait bir Girit vazo resmi.	15
Şekil 1.2 Roma Titus dönemine ait bir hamam kesiti.	15
Şekil 1.3 Ortaçağ Avrupa'sında yıkanma.	16
Şekil 1.4 Zenginlerle ve sıradan insanlara ait banyolar.	16
Şekil 1.5 Toplu yıkanma.	17
Şekil 1.6 Modern havuzlar.	17
Şekil 1.7 Antik duş figürleri	19
Şekil 1.8 İlk duş tasarımları.	19
Şekil 1.9 Çevresi kapalı küvet uygulaması.	20
Şekil 1.10 Ayaklı küvet uygulaması.	20
Şekil 1.11 Gömme küvet uygulaması ve tipik bir kesit.	21
Şekil 1.12 Gömme küvetler için kullanım şekilleri ve ölçüler.	22
Şekil 1.13 Köşe küvetlere ait bir uygulama ve çok kullanılan ölçüleri.	22
Şekil 1.14 Standart duş boyutları.	23
Şekil 1.15 Kullanımda bulunan bazı duş çeşitleri ve çok kullanılan duş aksesuarları.	23
Şekil 1.16 Mimar William Burges (1827-81) tarafından gerçekleştirilmiş olan Viktorya tarzındaki el yıkama masası.	25
Şekil 1.17 Soldaki iki resim Chippendale tarafından 1754 tasarlanan el yıkama masalarına aittir; sağdaki resim ise Sheraton tarafından 1803 de tasarlanan el yıkamalı giyinme dolabına aittir.	25
Şekil 1.18 Sırasıyla ayaksız (duvara asılan), yarım ayaklı, tam ayaklı, ve köşe lavabolara ait örnek çizimler.	26

Şekil 1.19 Sırasıyla ayaksız, ayaklı, ve tezgaha gömülen lavabolara örnek uygulamalar.....	27
Şekil 1.20 Roma dönemine ait tuvalet kalıntıları ve yine Roma dönemi tuvalet kullanımını anlatan temsili resim görülmektedir.	29
Şekil 1.21 Solda 17 yy a ait bir klozet, sağda ise 1778 de tasarlanan ve 1890 yıllarına başarıyla kullanılan bir klozet tasarımı görülmektedir.	29
Şekil 1.22 Solda 1880 yılına ait yunus şekilli bir klozet, sağda ise 1900 yılına ait özel bir tasarım görülmektedir.	29
Şekil 1.23 Çeşitli özelliklere sahip klozet ve bideler.	30
Şekil 1.24 İtalya'dan bir uygulama.	30
Şekil 1.25 Gideri alttan ve arkadan olan klozetler.....	31
Şekil 1.26 Standart alturka boyutları.....	31
Şekil 1.27 Sifonlar ve süzgeçler.	32
Şekil 1.28 Bataryalar.	33
Şekil 1.29 Çeşitli aksesuar çizimleri.....	34
Şekil 1.30 Carlo Bartoli den bir aksesuar serisi.	34
Şekil 2.1a Programın akış diyagramı.....	47
Şekil 2.1b Programın akış diyagramı.....	48
Şekil 2.1 Tek duvarlı olasılıklar: a). TS LA KL KU*, ve b). *DT LA KL.....	55
Şekil 2.2 İki duvarlı olasılıklar: a). KL*TS LA KU, ve b): LA*KL DT.	57
Şekil 2.2a Küvetle İlgili Kurallar.	60
Şekil 2.2b Küvetle İlgili Kurallar.	60
Şekil 2.2c Küvetle İlgili Kurallar.....	61
Şekil 2.3a Duş Teknesiyle İlgili Kurallar.....	61
Şekil 2.3b Duş Teknesiyle İlgili Kurallar.....	62
Şekil 2.3c Duş Teknesiyle İlgili Kurallar.	62

Şekil 2.4a Çamaşır Makinesiyle İlgili Kurallar.....	63
Şekil 2.4b Çamaşır Makinesiyle İlgili Kurallar.....	63
Şekil 2.5 Termosifonla İlgili Kurallar.....	64
Şekil 2.6 Şofbenle İlgili Kurallar.	64
Şekil 2.7 Kural Uygulaması için Bir Örnek : DT LA*KL TS CM.	65
Şekil 3.1 Birinci Ekran : Ana Menü.	71
Şekil 3.2 İkinci ekran : Firmalar Ana Menü.	72
Şekil 3.3 Üçüncü ekran : Firmalar Ara Ekranı.....	72
Şekil 3.4 Kural Tabanı Arabirimi	74
Şekil 3.5 Kural Tabanı Arabirimi : Silinecek Kural için Uyarı	74
Şekil 3.6 Altıncı Ekran: Olasılıkların Değerlendirilmesi.....	76
Şekil 3.7 Yedinci Ekran : Puanı Uygun Olan Olasılıkların Çizilmesi.....	76
Şekil G.1 Beş elemanlı çözüm; LA KL TS*CM KU.....	78
Şekil G.2 Beş elemanlı çözüm; CM LA TS KL KU*.....	78
Şekil G.3 Beş elemanlı çözüm; CM KL*TS LA KU.....	79
Şekil G.4 Dört elemanlı çözüm; LA KL*TS KU.....	79
Şekil G.5 Dört elemanlı çözüm; *KU KL LA TS.....	80
Şekil G.6 Dört elemanlı çözüm; *TS KL LA KU.....	80
Şekil G.7 Dört elemanlı çözüm; TS KL LA KU*.....	81
Şekil G.8 Dört elemanlı çözüm; KL*TS LA KU.....	81
Şekil G.9 Dört elemanlı çözüm; LA TS*KL KU.....	82
Şekil G.10 Üç elemanlı çözüm; *DT KL LA.....	82
Şekil G.11 Üç elemanlı çözüm; LA*KL DT.....	83
Şekil G.12 Üç elemanlı çözüm; KL LA DT*.....	83
Şekil G.13 İki elemanlı çözüm; KL*LA	84
Şekil G.14 İki elemanlı çözüm; LA KL*	84

TABLO LİSTESİ

Tablo 1.1 US uygulamaları	8
Tablo 1.2 US programlama dilleri.	11
Tablo 2.1 Programın akış şeması.....	46
Tablo 2.2a Küvetle İlgili Kurallar.	60
Tablo 2.2b Küvetle İlgili Kurallar.	60
Tablo 2.2c Küvetle İlgili Kurallar.	61
Tablo 2.3a Duş Teknesiyle İlgili Kurallar.....	61
Tablo 2.3b Duş Teknesiyle İlgili Kurallar.....	62
Tablo 2.3c Duş Teknesiyle İlgili Kurallar.....	62
Tablo 2.4a Çamaşır Makinesiyle İlgili Kurallar.	63
Tablo 2.4b Çamaşır Makinesiyle İlgili Kurallar.	63
Tablo 2.5 Termosifonla İlgili Kurallar.	64
Tablo 2.6 Şofbenle İlgili Kurallar.....	64
Tablo A.1 Kombinezon tablosu.....	96
Tablo B.1 Program Kodunun harf, sözcük, ve satır cinsinden uzunluğu.	97
Tablo B.2 Programda Yer Alan Sözcük ve Satır Uzunluklarının Ayrıntısı.	97
Tablo B.3 Programda Yer Alan Fonksiyon ve Değişkenlerin Dökümü.	97
Tablo B.4 Programda Yer Alan Fonksiyon ve Değişkenlerin, Uzunlukları ve Kullanım Miktarları.	97

ÖZET

Bilgi teknolojileri birçok alanda olduğu gibi, mimarlık mesleğinde de kendine önemli bir yer bulmuştur. İlk aşamada, çizim sürecini optimize edip hız kazandırmak amacıyla kullanılmışlardır. Fakat kısa bir süre sonra bilgisayarların mimarlar için yapabileceklerinin bununla sınırlı olmadığı anlaşılmıştır. Bir adım ileride, çeşitli Yapay Zeka yaklaşımlarını kullanan ve bir miktar “öngörüye” sahip olan yazılımlardan söz edilebilir. Bu çalışmanın amacı bu tür bir yazılımın gerçekleştirmektir.

Problem alanı olarak orta düzeyli konut banyosu tasarımı seçilmiştir. Banyolarda en sık kullanılan 5 eleman grubu probleme dahil edilmiştir: Lavabo, klozet, yıkanma, çamaşır yıkama, ve su ısıtma. Bu çalışmada banyo tasarımı dikdörtgen bir mekanla sınırlandırılmıştır.

Yazılım geliştirmede kullanmak üzere Autodesk firmasına ait olan AutoCAD programının, Windows işletim sistemi altında çalışan bir sürümü olan, AutoCAD sürüm 13 seçilmiştir. AutoCAD programı üzerinde yazılım geliştirmeye olanak veren AutoLISP isimli bir programlama diline sahiptir. Bu dil, birçok sürümü olan, ve Uzman Sistem uygulamalarında sıkça kullanılan LISP dilinin AutoCAD ortamına bir uyarlanmış bir türevidir.

Yazılım üç aşamadan oluşmaktadır. İlk aşamada kullanıcıyla grafik arabirimi aracılığıyla iletişim kurulur. Kullanılacak markalar, modeller, tasarıma dahil edilen elemanlar, kullanılacak duvar sayısı, vb. gibi bilgiler bu aşamada programa verilir. Seçilen eleman kombinezonu doğrultusunda olasılıkların türetilmesi de bu bölümde yapılır. Kural tabanıyla kullanıcının ilişkisi yine bu bölümde sağlanmaktadır. Kullanıcı bu bölümde, kural tabanına yeni bir kural ekleyebilir, bir kuralı silebilir ya da etkisizleştirebilir.

İkinci bölüm, kullanıcı tarafından algılanmayan fakat önemli görevlere sahip algoritmaların bulunduğu bir bölümdür. İlk bölümde kullanıcıdan alınan veriler ışığında oluşan kombinezondan türetilen tüm olasılıklar bu bölümde kural tabanı tarafından sınanırlar.

Bu bölümde “çizim motoru” bulunmaktadır. İkinci bölümdeki sınamayı geçen permütasyonlar burada çeşitli boyutsal analizlerden geçtikten sonra yine birinci bölümde kullanıcı tarafından bildirilen sayıda çizilirler. Çizimin dışında bu bölüm, veri tabanından aldığı bilgileri kullanarak üretilen her alternatifi maliyetini hesaplayıp kullanıcıya sunar.

SUMMARY

A RULE-BASED EXPERT SYSTEM FOR BATHROOM DESIGN

The information technologies have a very important place in Architecture like other fields. They have been used in early times in order to optimise and speed the drawing process. This is not the all that the computers can do for the Architects. In a step ahead of this, there are softwares which utilise the Artificial Intelligence approaches. Rule-Based Expert Systems are one of the most important field of Artificial Intelligence approaches. Building a Rule-Based Expert System for bathroom design is the main objective of this study.

In order to realise this, designing the middle class home bathroom is chosen as the problem area. The most common five components of bathroom are included in the problem. These are; washbasins, lavatories, showers, baths, washing machines, and hot water appliances. The response to the problem of bathroom design is limited with to rectangular solutions.

The AutoCAD release 13 for Windows from Autodesk is chosen as development environment for this study. This is a very powerful CAD software package which has millions of users all over the world. This software has AutoLISP programming language that is a variant of XLISP developed by David Betz. The LISP is a very well known programming language in Artificial Intelligence approaches because of its very open and powerful architecture.

The Rule-Based Expert System for Bathroom Design is a software which coded in AutoLISP programming language. The software has three main steps:

In the first part, the software communicates with user within a Graphical User Interface. The interface uses the Programmable Dialogue Boxes (DCL) of AutoLISP. The user chooses brands, models, elements and amount of the walls that will use in design, and the other optional item with this interface. Pushing the "Permutation" button the user can produce all the possible permutations for the selected combination. After that pushing the

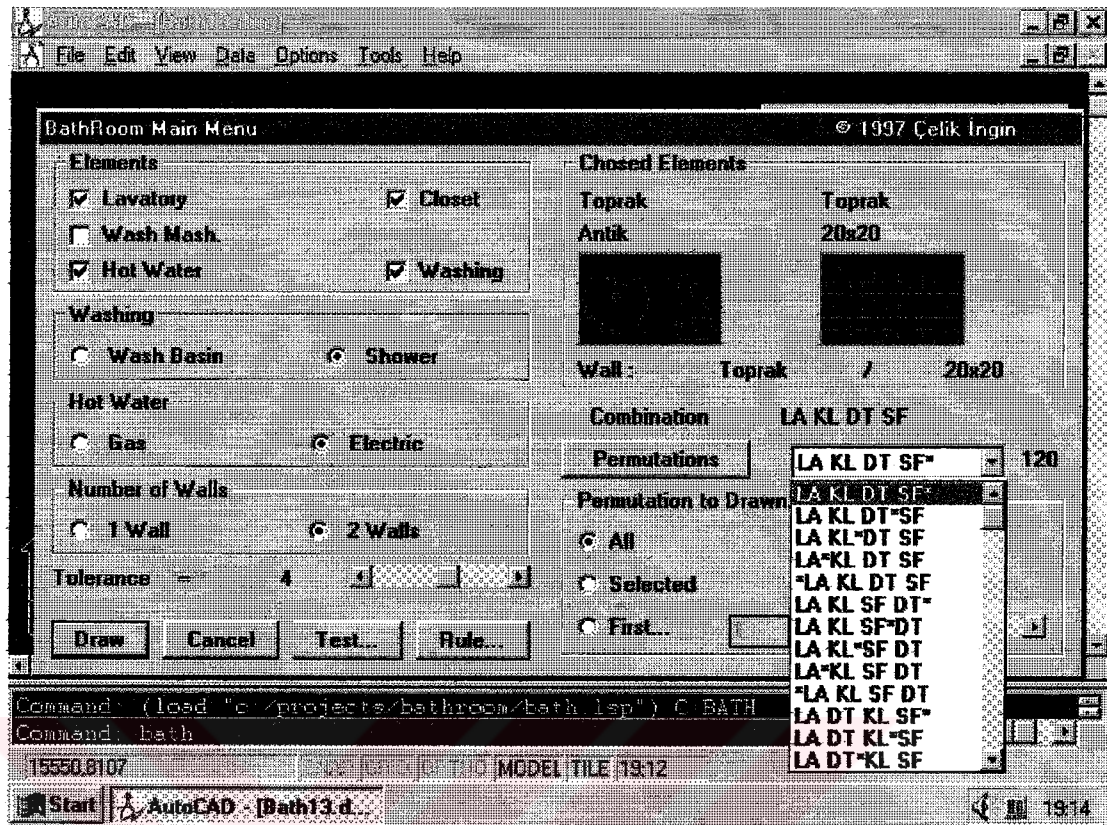


Figure 1. The Graphical User Interface

"Draw" button program will send these permutations to the further part of this application. User can manipulate rule-base within this first part. A new rule can be added, an existing rule can be removed or disabled. Figure 1. shows the first screen of this user interface.

The rule-base is placed in second part. The permutations created in the first part are tested here by the rules related to the rule-base. The software takes the rules that related to the selected combination from the file named Banyo.krl. After taking the related rules, the software converts them into AutoLISP programming codes. Evaluation of these rules generates a process that gives a set of results of failure points. Zero point means no failure, and 8 and above means inapplicable. The results will send to the next part as a LISP list.

The third part has the "drawing engine". The result matrix which contains the permutations and their failure points, is drawn here. The amount of the drawing outputs are depended on the options that user pointed out in the first part and the results of the test are indicated in the second part. The software also gives the approximate cost of the every possible permutation.

The cost is calculated by using the values in the file named Banyo.dta which has been set by the user.

The thesis is organised in to three main chapters. In first chapter, the basic concepts of the artificial intelligence and the expert systems, and the organisational principles of bathroom design are explained and discussed. In the second chapter, the programming language, the programming method of system, and the rule-base are explained. In the last chapter, a limited user guide of the software and some outputs of the system are given.

In appendixes, complete combination table, whole documentation of the AutoLISP and DCL program code, contents of the auxiliary files, and the rule-base are documented.

First Chapter:

Firstly, some of definition of artificial intelligence has been given. The fields which employ successfully artificial intelligence, like robotics, expert systems, learning, have been listed. Second part of this chapter includes Expert Systems. Expert systems have the most successful application of artificial intelligence science. The development stages of expert systems are explained here. These are:

1. Identification,
2. Conceptualisation,
3. Formalisation,
4. Implementation,
5. Testing.

After that, general types of expert systems are listed, like these;

1. Interpretation systems,
2. Prediction systems,
3. Diagnosis systems,
4. Design systems,
5. Planning systems,
6. Monitoring systems.

Very well-known and important applications of expert systems, like ; DENDRAL, META-DENDRAL, SAINT, EXPERT, CASNET, MYCIN, EMYCIN, PUFF, CADUCEUS, RITA, and ROSI, have been explained.

The short information's about general purpose expert system programming languages, like PROLOG, and LISP, have been given in next part.

The next part consists a detailed explanation of bathrooms and their most common components. This part started with a brief history of bathroom, and continue with its elements. In the next part, the dimensions of the elements are given. The last theme of this part is about the materials used in bathrooms commonly.

Second Chapter:

In next chapter the programming method and environment are explained. The reasons, why the AutoCAD software package and its programming language AutoLISP are chosen for the development is explained.

Following that, the programming method is explained step by step (procedure by procedure). The explanations of procedures that include permutation generator, inference machine, and the rule-base-to-AutoLISP rule translator have been given in this part.

The next part contains the detailed explanations about the rule-base and the evaluation process. System does not use a real data-base format or file to save the rule information. An ordinary text file contains the formatted rule primitives. The software reads the primitives and converts them to real `if ... then ...` clauses.

The rule-base has three sections. In the first section, the rules for one-wall-using-combinations (the term one-wall is used to explain one wet wall for installation) are placed. The second section contains the rules for the combination which use two-walls. These rules are applied to the elements which are placed in the intersection of two wet walls. The two-walls combinations need to be applied by the one-wall rules.

The last part of the rule-base is special. It is reserved for the advanced users. In this section the advanced users can write independent LISP rules in order to evaluate information. User can reach the internal variables of the program or define a variable in order to use further lines of this part.

Third Chapter:

This chapter includes a small user guide. The software is explained screen by screen. The guide directs the user by explaining the function of the buttons, and purpose of the screens.

A small performance benchmark is placed in third chapter. System gives its own hardware's performances in order to compare with other systems and hardware's. Finally, some drawing outputs of system are also given in this part.

In appendix A, there is a full explanation of combination table (this is not a real table, this is a set of calculations) the system employs. The table gives total amount of the possibilities (totally; 5.508) the system can generate.



BÖLÜM 1

GİRİŞ

Hayatın birçok alanına olduğu gibi, mimarlık sektörüne de bilgi sistemleri dönüşü olmayacak şekilde girmeye başlamıştır. Bu alanda bilgisayar kullanımı ilk olarak çizim ve sunum sürecini hızlandırmak amacıyla gerçekleşmiştir. Bilgisayarların daha üstün grafik yeteneklerini ve işlem gücünü daha uygun fiatlarla sunmasıyla birlikte, etkili grafik sunum alanı da, bilgisayarların mimarlara ve tasarımcılara hizmet verdiği alanlar arasına girmiştir. Bilgisayarların tasarım ya da sunum sürecinde hızlı ve doğru çizen bir kalemde daha etkin görevler alması, yapay zeka yaklaşımı ve uzman sistemlerin bu alana girmesiyle, geleceğe ait tahminler olmaktan çıkmıştır. Bilgi mühendisliğiyle derlenen uzman tecrübesi, ilişkileri ve kuralları belirlenen deneyim birikimi, artık tasarım sürecine tavsiyeleriyle deneysel katkılarda bulunacak adımlar atmaya başlamıştır.

“Banyo tasarımı için kural tabanlı uzman sistem” yukarıda sözü edilen yaklaşımlardan ve gelişmelerden yola çıkarak geliştirilmiş bir yazılımdır. Bu yazılımla, amaçlanan banyo tasarımında söz sahibi bir çalışma ortaya koymak değildir. Bu çalışmanın asıl çıkış noktası doğru çalışan, esnek ve gelişmeye açık, kural tabanlı bir uzman sistem yazılımı ortaya koymaktır. Bununla beraber yazılım, gerek eğitim amaçlı gerekse ticari amaçlı banyo tasarımının söz konusu olduğu bir çok durumda gündeme gelebilir.

1.1 Yapay Zeka (YZ)

YZ terimi ilk defa 1956 yılında John McCarty tarafından ABD’de “Machine Intelligence” kongresinde ortaya konmuş bir kavramdır. Tanımladığı olgularla birlikte düşünüldüğünde tam olarak doğru bir isimlendirme olmadığı söylenebilirse de yerleşmiş bir kavramdır. Winston (1987) YZ yi şöyle tanımlar; “YZ, durumların bilgisayar aracılığıyla karmaşık ve insan benzeri davranışlarla ilişkilendirilmesiyle

ilgili bir alandır". Winston bu tanımı yaparken vurguyu "karmaşık" ve "insan benzeri" nitelermeleri üzerinde yapar ve bu özelliklerin YZ alanını geleneksel bilgisayar bilimlerinden ayırdığını söyler. Trapple (1986) tarafından yapılan tanıma göre "Yapay Zeka a) Bilgisayarları çekici kılmak, b) İnsan zekasına sahip bilgisayarlar geliştirmek, c) İnsan davranışını benzeştiren makinalar (robotbilim) yapmaktır."

Makinalara, insana ait yetileri kazandırma, fikri kökenleri çok eskilere dayanan bir fikirdir. Bu fikrin uygulamalarına, makinelerin "sadece mekanik" oldukları çağlardan itibaren rastlanmaktadır. Bu girişimlere verilebilecek en iyi örnek İngiliz matematikçi Charles Babbage'in (1791-1871, Londra) tasarlamış olduğu çözümleyici makinedir. Bu tasarım mekanik parçaların hata toleranslarının hiçbir zaman istenilen düzeye düşmemesi yüzünden gerçekleştirilememiştir. Bu ve benzeri, çağının teknolojisinin sunabildiklerinden daha fazlasına ihtiyaç duyan birçok tasarım uzun süre kağıt üzerinde kalmıştır. Ancak 1940'li yıllardan itibaren, önce elektronik, sonra bilgisayar ve son olarak da yazılım geliştirme araçları sektöründe yaşanan hızlı gelişmelerle birlikte bu tür düşünceler tekrar canlanmıştır. Özellikle bilgisayarların çalışma hızlarındaki artış ve yeni nesil YZ programlama dillerindeki gelişmeler YZ araştırmalarını ve uygulamalarını ivmelendirmiştir.

Winston (1987) YZ tekniklerinin kullanıldığı alanları şöyle sıralar:

- bilgisayar destekli robot bilimi,
 - görme,
 - yönlendirme,
 - mekan algılama,
 - nesne ilişkilerini algılama,
- algıların yorumu,
 - görsel algılama,
 - ses algılama,
 - işaretleri sembollere çevirme,
- uzman sistemler,
 - kural tabanlı çıkarım,
 - benzeşim tabanlı çıkarım,
- doğal dil anlama sistemleri,
 - konuşma,
 - sözdizimsel çözümleme,
 - soru anlama,
 - söylev anlama,
- öğrenme,
 - şekil okuyarak öğrenme,
 - örnekleri açarak öğrenme,
- sağduyulu çıkarım,
 - değişken-doğruluk çıkarımı,

- nicelikler üzerine biteliksel çıkarım,
- süper bilgisayar donanım ve yazılımı,
 - büyük boyutlu paralel mimari,
 - çok yüksek düzeyli programlama dilleri.

Bu kullanım alanlarının yanında, İşletme Mühendisliği, Bilgisayar Bilimleri, Kontrol Mühendisliği, Elektrik Mühendisliği, Dilbilim, Matematik, Makine Mühendisliği, Kavrayış Psikolojisi, Algısal Psikoloji, Motor Psikoloji, ve Felsefe gibi birçok akademik disiplin tarafından da araştırma konusu olarak kabul edilmektedir.

1.2 Uzman Sistemler (US)

Uzman Sistemler, Yapay Zeka araştırmalarının ticari açıdan en çok uygulama olanağı bulan ve üzerinde en çok çalışılan bir alt dalıdır. YZ literatüründe US için yapılmış bir çok tanım bulunmaktadır:

“US ler, sınırlı bir problem alanında, yüksek düzeyde bir performans elde etmek için uzman bilgisini kullanan bilgisayar programlarıdır” (Waterman, 1986).

“US, bir problem alanındaki bir veya daha fazla uzmanın bilgi ve deneyimini kullanan bir bilgisayar programıdır. US ler çıkarım (inference) yapmak için alan uzmanlarının sorun çözme deneyimlerini kullanırlar” (Gevarter, 1985).

“US ler, insan uzmanın tasarım, oluşturma, planlama, tanı koyma, yorumlama, özetleme, ve tavsiyede bulunma gibi yeteneklerini benzeştirmek için oluşturulmuş bilgisayar programlarıdır” (Madni, 1988).

Uzmanlık, yıllar süren meslek pratiğinden çıkarılan bilgidir. Bu bilginin iki türü vardır; ya nesnenin bilgisine doğrudan sahip olurnur, ya da bu bilginin nereden bulunabileceği bilinir. Bir uzmanın her iki türlü bilgiye de sahip olması gerekir (Rosenman, 1987).

Yukarıda yapılan tanımlamalar ışığında bir US in, belirli bir alana ait bilgiyle yoğun bir şekilde desteklenmiş, çıkarım (reasoning) yapabilen bir bilgisayar programı olduğu söylenebilir.

Bir US'i diğer YZ uygulamalarında ayıran en önemli özelliği, sahip olduğu bilgi tabanıdır. Aynı şekilde, bir US oluşturulması sürecinin en önemli adımlarından biri uzmandan bilgi çıkarılması (knowledge acquisition) işlemidir.

Uzman Sistemleri yapay zeka uygulamalarından ayıran temel farkı anlatmak için "kaba kuvvet" (brute force) kullanan satranç programları örnek verilebilir. Bu tür programlar rakibin hamlesine karşılık verebilmek için, mevcut durumda oynanabilir olan bütün hamleleri denerler, ve bunların arasından o an için en iyi olanı uygularlar. Burada seçilecek hamlenin kalitesi büyük oranda kullanılan makinenin ve programlama dilinin hızına bağlı olduğu için bu tarz programlara kaba kuvvet kullanan programlar denir. Bu yöntemle programlanmış satranç programları YZ programları olmalarına karşın, sonuca varırken sınırlı miktarda uzman ya da alan bilgisi kullanırlar, bu nedenle US sınıfına girmezler.

1.2.1 US Geliştirme Aşamaları

Yüksek performans elde etmek için uzman bilgisine ihtiyaç duyulduğundan, US ler aşama aşama geliştirilirler. Bu evrimsel geliştirme tarzı, US sistem geliştirme yöntemleri arasında en yaygın kullanılanıdır. Uzmandan bilgi alınması ve bu bilginin bilgisayar programının bilgi tabanına kaydedilmesi işlemine bilgi çıkarımı (knowledge acquisition) denir. Uzmanlık bilgisinin bir bilgi kaynağından alınarak programa aktarılması ve dönüştürülmesi süreci, US geliştirme sürecinin en önemli adımıdır (Hayes-Roth, 1983).

US geliştirme sürecinin diğer adımları şu şekilde sıralanabilir:

1.2.1.1 Tanımlama (identification) Aşaması

Bu aşamada alan uzmanı ve bilgi mühendisi problem alanını netleştirmek ve sınırlarını koymak için birlikte çalışırlar. Aynı ekip geliştirme sürecine katılacak diğer kişileri (uzmanları), ihtiyaç duyulan kaynakları (zaman, donanım ihtiyacı, vb.), ve US geliştirmenin hedeflerini ya da amaçlarını belirler.

1.2.1.2 Kavramları Geliştirme (conceptualization) Aşaması

Bu aşamada uzman ve bilgi mühendisi, söz konusu alanda problem çözme sürecini tanımlayabilmek için ihtiyaç duyulan, anahtar kavramlara, ilişkilere, ve bilgi akış diyagramlarına açıklık getirirler. Alt görevler, stratejiler, ve problem çözme aşamasıyla ilgili sınırlamalar yine bu süreçte ortaya konur.

1.2.1.3 Biçimlendirme (formalization) Aşaması

Bu aşama, anahtar kavramların ve ilişkilerin, seçilen US geliştirme aracı ya da dili tarafından sağlanan sunum şekillerine dönüştürülmesini içerir. Bilgi mühendisi kullanılacak dili seçer ve, uzmanın da yardımıyla, temel kavram ve ilişkileri dilin yapısı içinde sunar.

1.2.1.4 Uygulama (implementation) Aşaması

Bu aşamada bilgi mühendisi, bilgiyi, problemin bilgi akış diyagramlarıyla uyumlu hale getirmek amacıyla, birleştirir ve tekrar organize eder. Bu aşama sonucunda, çalıştırılmaya ve test edilmeye hazır bir programı tanımlayan kurallar ve ilişkili kontrol yapısı ortaya çıkar.

1.2.1.5 Sınama (testing) Aşaması

Bu aşama, prototip programın performansının değerlendirilmesi ve problem alanında uzmanın tanımlamış olduğu doğruluk standartlarıyla uygunluğunun sağlanması için düzeltilmesi çalışmalarını içerir. Alan uzmanı programın performansını değerlendirir ve düzeltmelerin yapılması için bilgi mühendisine yardımcı olur.

1.2.2 US Tipleri

US ler, üzerinde çalışılan problemin niteliğine göre farklı çözümlere yönelirler ve yapılanmaları da ona göre değişir. US lerin çözüme giderken izledikleri yollar şöyle sınıflandırılabilir (Hayes-Roth, vd., 1983).

1.2.2.1 Yorum (interpretation) Sistemleri

Bu sistemler gözlemlerden durum tanımı çıkarırlar. Bu sınıflamaya keşif, konuşma, anlama, görüntü çözümlemesi, kimyasal yapıları açıklama, işaret anlama gibi entellektüel çözümlemelerin herhangi biri girebilir. Bu tür sistemlerde gözlenen verilere sembolik anlamlar verilerek, elde edilenler üzerinde değerlendirme yapıp durum açıklanır.

1.2.2.2 Tahmin (prediction) Sistemleri

Bu sistemler, kendisine verilen bilgiler ışığında, bazı durumlar sonucunda ortaya çıkabilecek olanları tahmin ederler. Bu sınıflamaya hava tahmin sistemleri, nüfus tahminleri, trafik tahminleri, tahıl üretim tahminleri gibi konular girer. Tahmin sistemleri parametrik ve dinamik bir model kullanırlar. Bu modelde parametre değerleri, verilen duruma göre değişir. Modelden elde edilen sonuçlar tahmin için zemin oluştururlar. Bu tahmin sistemleri sadece bir sonuç vermek yerine, bir çok senaryo da ortaya koyabilirler.

1.2.2.3 Tanı (diagnosis) Sistemleri

Bu sistemler, gözlemleri sonucunda arızaların kaynağını bulan sistemlerdir. Tıp, elektronik, mekanik, ve yazılım hataları bu sınıfa girer. Tanı sistemleri gözlenen davranış bozukluklarının altında yatan nedenleri bulmakla görevlendirilirler. Bu amaçla iki teknik kullanılabilir:

- a. Sistemde esas olarak davranış ve tanı arasındaki bağlantıları içeren tablolar

kullanılır.

- b. Sistem gözlemlerle, bozukluğa neden olma olasılığı bulunan tasarım, yapım ya da bileşenlerdeki olası hata bilgileriyle sistem tasarımı bilgilerini birleştirir.

1.2.2.4 Tasarlama (design) Sistemleri

Tasarlama sistemleri, tasarlama kanıtlamalarını sağlayan nesne konfigürasyonlarını geliştirirler. Bu problemler elektronik devre planları, bina tasarımları ve bütçe yapımıyla ilgilidir. Tasarlama sistemleri, tasarıma dahil olan nesnelerin aralarındaki ilişkilerini tanımlarını yapar. Bunu yanında bir çok tasarlama sistemi olasılık teorisinin maliyet ve istenmeyen özellikleri ölçen amaç fonksiyonunu minimize ederek sonuca ulaşmaya çalışır. Tasarlama problemine bu bakış açısı, sistemin sonuca yönelik davranışını da belirler.

1.2.2.5 Planlama (planning) Sistemleri

Bu sistemler, belirli bir işleve sahip nesnelerin, tasarım problemleri üzerinde özelleşmişlerdir. Bunlar otomatik programlama, robot bilim, projeler, yol yapımı, iletişim, deney, ve askeri planlama projeleriyle ilgilidir. Planlama sistemleri, planlanan eylemlerin etkilerini görmek için eylem modellerinden faydalanırlar.

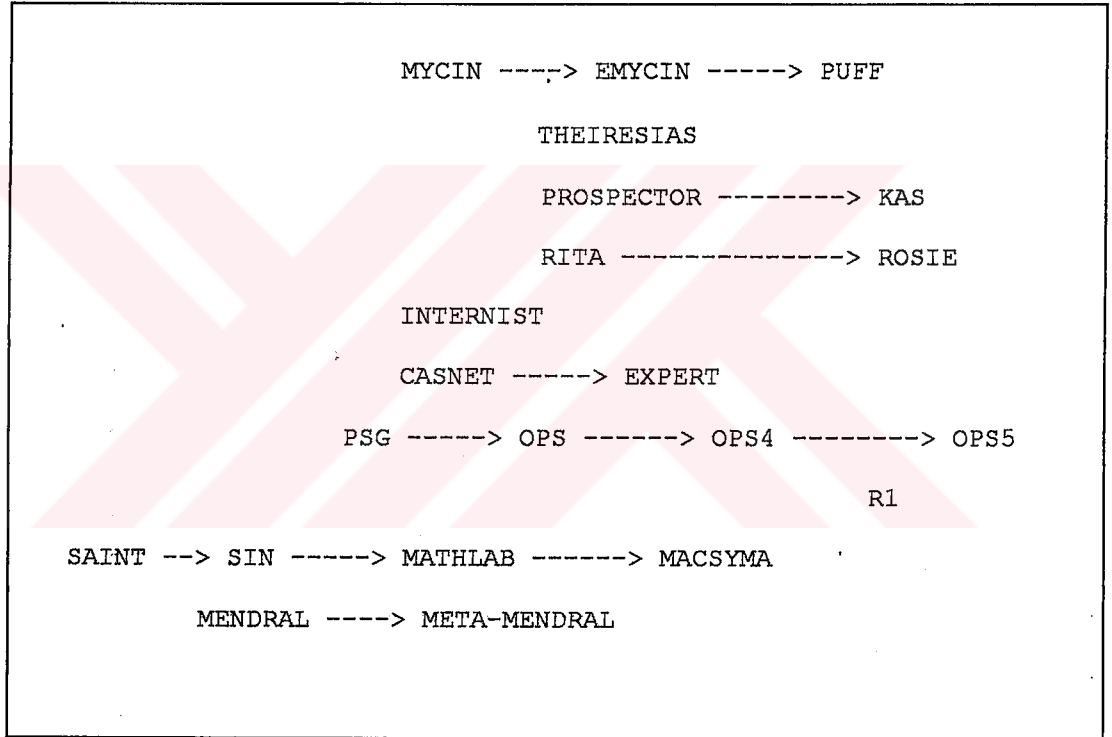
1.2.2.6 İzleme (monitoring) Sistemleri

Bu sistemler, planlama sonuçlarının başarısını değerlendirmek amacıyla, sistem davranışlarına ait gözlemlerle, zorunlu görülen özellikleri karşılaştırırlar. Bu önemli özellikler ya da zayıflıklar planın olası bozukluklarıyla ilgilenir ve sistemin sağlığı konusunda bilgi verir.

1.2.3 US Uygulamaları ve Geliştirme Araçları

Geliştirilmiş US bir kısmı Şekil 1.1 gösterilmiştir. Projelerin büyük kısmı yıllar boyunca geliştirilmeye ve yeni sürümler vermeye devam ettiği için, birbirinin devamı olan projeler aynı satırda gösterilmiştir. Bu US lerden bazıları, sahip oldukları bilgi tabanı boşaltılarak, ve bir takım yeni özellikler eklenerek genel amaçlı US geliştirme araçlarına dönüştürülmüşlerdir. Aşağıda en önemlileri açıklanmıştır (Hayes-Roth, 1983):

Tablo 1.1 US uygulamaları



1.2.3.1 DENDRAL

Stanford Üniversitesi tarafından yürütülen bu proje 30 yıldan beri sürdürülmektedir (Buchanan ve Mitchell, 1977,1978; Buchanan ve Feigenbaum, 1978; Feigenbaum, Buchanan, ve Lederberg, 1971; Lindsay, vd., 1980). Bu projeden iki sistem üretilmiştir; DENDRAL ve META-DENDRAL. DENDRAL, yeni kimyasal bileşimler keşfetmek için, spektrografik nükleer manyetik rezonans gözlemlerini, ve kimyasal deneylerin verilerini çözümler. DENDRAL problem çözerken oldukça yeterli bir

yarat-test et yöntemi kullanır. İkna edici olmayan olasılıkları derhal eleyerek çalışmasına hız kazandırır.

1.2.3.2 META-DENDRAL

Bu yeni sürüm, organik yapıların parçalanma kurallarını üreten ve seçen çözümleme bilgisini DENDRAL'e ekleyerek onu geliştirir. Deneylerden kaynaklanan verileri inceleyerek olası parçalanma durumlarını yaratır ve sınar, yeterince başarılı olanların hipotetik kurallarını geriye iletir. DENDRAL projeleri, insanların ve makinelerin kimya araştırmalarındaki rollerini tartışmaya açtıracak kadar başarılı olmuştur.

1.2.3.3 SAINT

Bir diğer gelişim çizgisi SAINT (Slagle, 1961) ile başlar ve MACSYMA (Martin ve Fateman, 1971) ile son noktasına varır. Bu seri MIT tarafından geliştirilmiş, sembolik matematikle ilgilenen US leri içerir. Sembolik olarak diferansiyel ve integral hesaplamaları yapar ve sembolik ifadelerle basitleştirerek sonuçları iyileştirir. Bütün dünyadan bir çok matematikçi ve fizikçi tarafında hergün yoğun olarak kullanılan MACSYMA, uygulamalı matematikçiler tarafından üretilen kurallarla desteklenmiştir.

1.2.3.4 EXPERT ve CASNET

Bir US geliştirme aracı olan EXPERT (Weiss, Kulikowski, ve Safir, 1978; Weiss ve Klukowski, 1979), "glaucoma" hastalığının tanı ve tedavisinde kullanılmak üzere geliştirilmiş bir US olan CASNET'in devamı olarak üretilmiş bir sistemdir. EXPERT temel olarak oftalmoloji, endokrinoloji, ve rümatoloji disiplinlerinde tedavi modelleri geliştirmek üzere kullanılmıştır.

1.2.3.5 MYCIN, EMYCIN, PUFF, ve CADUCEUS

Pittsburg Carnegie-Mellon Üniversitesin'de geliştirilen CADUCEUS (Pople, Myers, ve Miller, 1975; Pople, 1981) ve Stanford Üniversitesin'de geliştirilen MYCIN (Shortliffe, 1976) tıbbi tanı sorunlarını çözmek amacıyla kullanılan iki ayrı US dir. CADECEUS, dahili tıp disiplinine giren belirti ve hastalıkların çok büyük bir anlamsal ağına sahiptir. 1982 yılında sisteme, bu konuda derlenmiş olan bütün birikimin % 85 olan, 100.000 adet ilişkilendirme dahildi.

MYCIN bulaşıcı kan hastalıklarının tanı ve tedasında görevlendirilmiştir. Kullandığı bilgi,olası durumlarla ilişkili olan , 400 kuraldan oluşur. Uzmanların yaptığı değerlendirmeler sonucunda, MYCIN'in en az uzmanlar kadar tutarlı değerlendirmeler yaptığı anlaşılmıştır.

Stanford Üniversiteside, MYCIN'in alandan bağımsız bir sürümü olan EMYCIN (van Melle, 1979) geliştirilmiştir.EMYCIN, bulaşıcı kan hastalıklarına ait alan bilgisi dışında, MYCIN'in sahip olduğu herşeye sahiptir. EMYCIN, PUFF (Freiherr, 1980) gibi, bir çok tanı uygulamasının geliştirilmesine öncü olmuştur.

1.2.3.6 RITA ve ROSIE

Rand firmasına ait bir ticari uygulama olan ROSIE (Fain, vd., 1981) US geliştirmeye yarayan genel amaçlı bir programlama sistemidir. ROSIE, kendisinden önce üretilmiş bir programlama sistemi olan RITA (R. Anderson ve Gillogly, 1979) geliştirilerek oluşturulmuştur. Her ikisi de çıkış noktalarını MYCIN'in kural tabanlı bilgi temsil yapısından almışlardır.

1.2.5 Genel Amaçlı US Programlama Dilleri

US uygulamalarında yazılım geliştirmek için kullanılabilecek iki tür programlama dili vardır (Bayazıt ve Kavaklı, 1992) :

1. Probleme yönelik diller. FORTRAN, COBOL ve PASCAL bu sınıfa girer. Bu diller özel bir sınıf problem için tasarlanmış programlama dilleridir. FORTRAN, cebirsel problemlerin çözümü için tasarlanmış bir dildir, bu amacı gerçekleştirebilmesi için bir çok özel fonksiyon içerir. Benzer şekilde COBOL iş kayıtlarının tutulması için geliştirilmiştir.
2. Sembol işleme temelinde çalışan diller. LISP ve PROLOG bu sınıfa girer. Sembol işleme mantığı, YZ uygulamalarında sıkça karşılaşılan, karmaşık kavramları simgelemek ve işlemek için tasarlanmışlardır. LISP semboleri liste yapıları şeklinde işleyecek mekanizmalara sahiptir. Liste yapıları, her bir parçanın bir sembol, ya da bir başka liste olabileceği konumda parantezlerle kapatılan bir parçalar koleksiyonu, karmaşık kavramları simgelemek için kullanılan inşa bloklardır. LISP, US yazılımlarında en çok kullanılan programlama dilidir. Kullanım yaygınlığı açısından LISP'i PROLOG izler.

Tablo 1.2 de US yazılımlarında çok kullanılan bazı yazılım geliştirme araçları, etkin çalıştıkları donanım, ve üretici firmaların listesi verilmiştir.

Tablo 1.2 US programlama dilleri.

Program	Donanım	Üretici Firma
INTELISP-D	Xerox 1100	Exrox Corp.
LISP (COMMON)	Lamda mach.	Lisp Inc.
PROLOG	DEC 10 System	Quintus Computer
SMALLTALK-80	Tekronix 4404	Xerox Corp.
ZETALISP	Symbolics 3600	Symbolics Inc.

1.2.5.1 PROLOG

PROLOG ismi, PROgraming inLOGic kelimelerinden türetilmiştir. 1973 yılında, Fransa Marseilles Üniversitesinde, Alain Colmerauer ve yönetimindeki ekip tarafından geliştirilmiştir (Barut, 1991). Bu ekibin elde ettiği sonuçlardan yola

çıkarak, Eddinburg Üniversitesinden Robert Kovalsky tarafından mantıksal programlama teorisi geliştirilmiştir.

Marseilles PROLOG olarak da bilinen ilk sürüm PROLOG I dir. 1981 yılında, o tarihlerde mikro bilgisayarlar olarak adlandırılan, 8 bitlik PC lerde çalıştırılmak amacıyla, McCabe tarafından Micro PROLOG geliştirilmiştir. Bu sürümün diğerlerinden farkı, söz diziminde ve komutların uygulanış tarzında ortaya çıkmaktadır. Colmerauer ve ekibi tarafından 1982 yılında PROLOG II geliştirilmiştir. PROLOG I ve PROLOG II nin ticari hakları 1986 yılında Borland Corp.' satılmıştır. Borland'ın 1989 yılında PROLOG'a verdiği desteği çektiğini açıklamasıyla birlikte PROLOG'un geleceği belirsizleşmiştir. 1990 yılında Colmerauer ve ekibi PROLOG'u Borland'dan geri alıp ve yeni bir sürüm üzerinde çalıştıklarını açıklamışlardır. Bu açıklamanın ardından ekip, zamanının teknolojik gelişmelerine uyum sağlamış olan, PDC 3.2 numaralı PROLOG sürümünü duyurmuşlardır.

1.2.5.2 LISP

LISP 1958 yılında Massachusetts Institute of Technology (MIT) de John McCarthy tarafından geliştirilmiştir. LISP kısaltması LISt Processing kelimelerinden oluşturulmuştur. US uygulamalarında yaygın olarak kullanılmasının gerekçeleri şöyle sıralanabilir (Bayazıt ve Kavaklı, 1992) :

1. Kolay ve esnek sembol işleme,
2. Hafıza ayarlarının kendiliğinde yapılması,
3. Gelişmiş yazım ve hata düzeltme desteği,
4. Program kodu ve verinin aynı tarzda ele alınması. Bu, bir LISP programının, verisini değiştirebileceği kadar kolayca kendi kodunu da değiştirebileceği anlamına gelir. Bu son özellik ona, bilgi tabanında bulunan verileri, bilgileri öğrenen programlar yazma olanağı verir.

Değişik kullanıcı gruplarının, farklı ihtiyaçlarına cevap verebilmek için bir çok LISP sürümü geliştirilmiştir. Bunlar iki ana başlık altında toplanabilir:

1. Xerox PARC (Paolo Alto Research Center) tarafından geliştirilen INTERLISP, INTERLISP-D, ve VAX LISP isimli sürümler.
2. MIT'te yürütülen MAC projesi kapsamında geliştirilen sürümler. MACLISP, Franz LISP, ZETALISP, ve GC-LISP bu gruba girer. Bu sürümlerden, Franz LISP UNIX makinelerinde, ZETALISP 3600 Sembolics makinelerinde, GC-LISP PC lerde kullanılmak üzere geliştirilmiştir.

John McCarthy, LISP i geliştirirken gerçekleştirmek istediklerini şöyle sıralar (McCarthy,1979) :

1. Numaralar yerine sembolik ifadelerle çalışmak,
2. Sembolik ifadelerin ve diğer bilgilerin, bilgisayarın hafızasında liste yapıları olarak temsil edilmesi,
3. Bilginin dış ortamlarda, çoğunlukla çok düzeyli listeler ve bazen de S ifadeleriyle temsil edilmesi,
4. Fonksiyonlar olarak görev yapacak olan bir dizi seçici ve uygulayıcı,
5. Fonksiyonların, daha karmaşık fonsiyonları yaratmak üzere bir arada kullanılmalarını sağlayan bir dizi araç,
6. Koşul cümlelerinin, fonksiyon tanımlamalarına dallanmak amacıyla kullanılabilmesi,
7. Koşul cümlelerinin, işletilebilir fonksiyonları oluşturabilmek için, rekürsif olarak kullanılabilmesi
8. Fonsiyonları isimlendirmede komut cümlelerinin kullanılabilmesi,
9. LISP program kodunun, LISP verisi şeklinde temsil edilebilmesi,
10. Koşul cümlelerinin Boolean bağlantıları şeklinde yorumu,
11. Bir LISP fonksiyonu olan eval in hem dilin normal bir üyesi gibi, hem de bir komut yorumcusu olarak çalışabilmesi,
12. Silme probleminin giderilmesi için "garbage collection" ın kullanılması.

LISP dilinin geniş hacimli US ve YZ uygulamalarında kullanımının yaygınlaşmasıyla birlikte, LISP kodunu en verimli şekilde işlemek amacıyla LISP temelli bilgisayarlar tasarlanıp üretilmiştir. Son 10 yılda işlemcilerin hızında meydana gelen artış sayesinde masaüstü bilgisayarlarda da çalışabilecek LISP türevleri geliştirilmiştir.

1.3 Banyo

Banyoyla ilgili bir uzman sistem yazılımının oluşturulabilmesi için öncelikle banyonun ne olduğunun, hangi işlevlere karşılık verdiğinin, hangi elemanlardan oluştuğunun, ve elemanlar arasındaki ilişkilerin hangi kurallara bağlı olduğunun bilinmesi gerekmektedir.

Bu bölüm altında, oluşturulacak uzman sisteme bilgi üretmek ve problemi açmak amacıyla, banyo tasarımında rol alacak elemanlar, ve banyo tasarım ilkeleri ayrıntılı olarak incelenmiştir.

1.3.1 Tarihçe

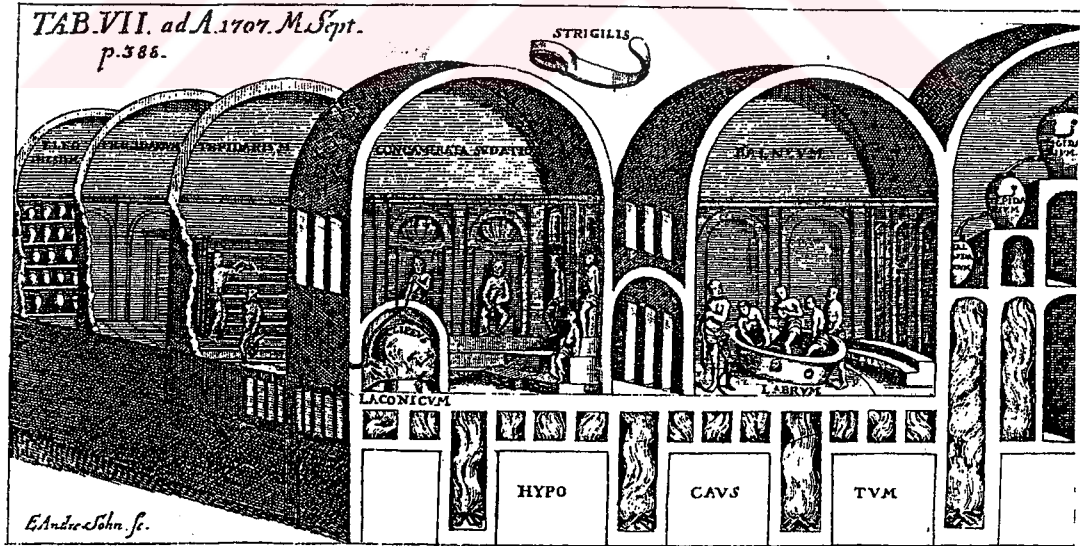
Batıda, özellikle Batı Avrupa'da 19. yy kadar fıçı veya benzeri kaplar içinde durgun suyun kullanılması şeklinde yapılan vücut yıkama, doğuda evlerin içindeki gusülhanelerde ya da dışarıdaki hamamlarda taşınan suyun dökülerek uygulanması esasına göre düzenlenmiştir.

19. yy kadar ataerkil aile tipinde geleneksel aile topluluğu içerisindeki küçük ailelerin her biri için ayrı ayrı yatma mekanlarında gusülhane veya hamamlıklar kullanılmıştır. Eski Türk evlerinde avlunun giriş kapısının sağında veya solunda evin tuvaleti bulunur, tuvaletin kibleye gelmemesine dikkat edilirdi. Dış sofalı evlerde genellikle tuvaletler ikametgah kısmından ayrılıp bahçeye yerleştirilmiştir. Odalarda yüklükler vardır. Bazı evlerde bazı odaların yüklüklerinin bir bölümü gusülhanedir.



Şekil 1.1 M.Ö. 5. yy a ait bir Girit vazo resmi.

Bu vazo resmi bir Helen kadını yardımcılarıyla birlikte banyo sırasında göstermektedir.



Şekil 1.2 Roma Titus dönemine ait bir hamam kesiti.

Oldukça karmaşık bir pratiğe sahip olan Roma yıkanma kültürü aynı zamanda da sosyal bir eylemdi.



Şekil 1.3 Ortaçağ Avrupa'sında yıkanma.

Sol: Ortaçağ aileleri yaklaşık her üç haftada bir yıkanmak ve temizlenmek için bir araya gelirler. Fiçi içinde yıkanan insanlara küçük ahşap kovalar ile hem su getirilir hemde kirli suları götürülür.

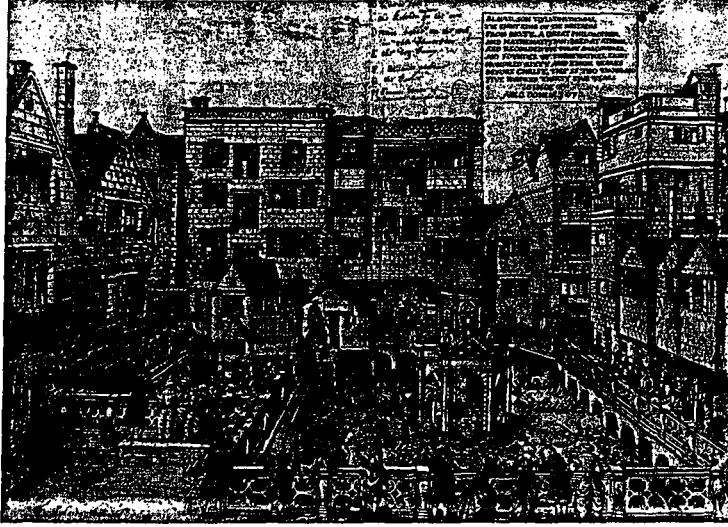
Sağ: 14 yy insanları için yıkanma yanı zamanda bir keyif bahanesiydi. Fiçi üzerine yerleştirilen ahşap üzerinde yemek yemek oldukça yaygındı.



Şekil 1.4 Zenginler ve sıradan insanlara ait banyolar.

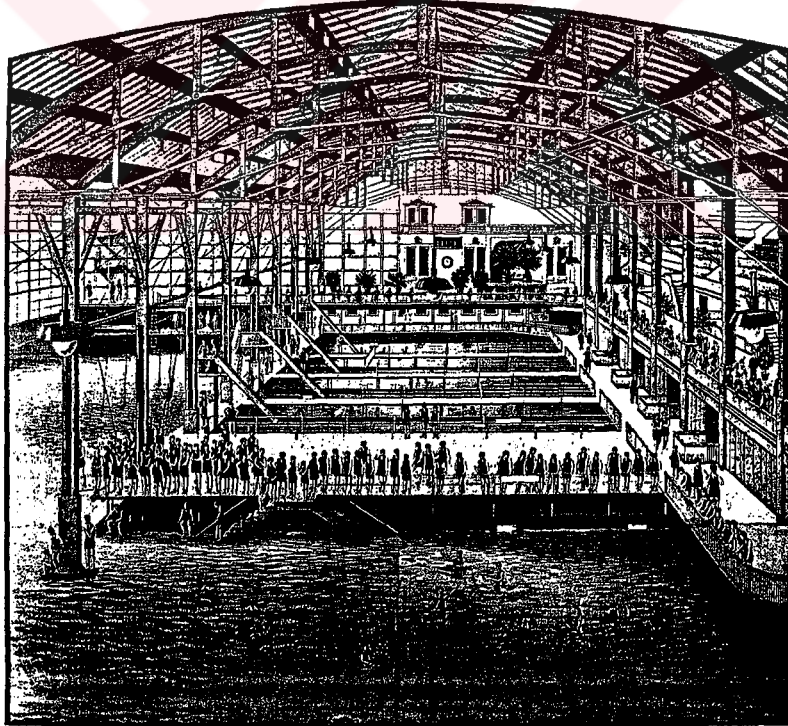
Sol: 1470 li yıllarda sıradan insanların evlerinde yemek yemek, yatmak ve yıkanmak için kullanılabilecek sadece bir sıcak oda bulunurdu.

Sağ: 18 yy in varlıklı insanları için banyolar, evin en iyi mobilyalarından biri olarak tasarlanırdı.



Şekil 1.5 Toplu yıkanma.

18. yy ait bu resimde, sıcak aylarda halkın müzik ve seyircilerin eşliğinde topluca yıkanması görülüyor.



Şekil 1.6 Modern havuzlar.

1900 lü yılların başına ait olan bu resimde çok büyük bir havuz görülmektedir.

1.3.2 Başlıca Eylemler, Elemanlar, ve Boyutları

Bu bölümde banyolarda yoğun olarak kullanılan elemanlar malzeme, boyut, kullanım alanı, vb. konularda incelenmiştir.

1.3.2.1 Vücut Yıkama

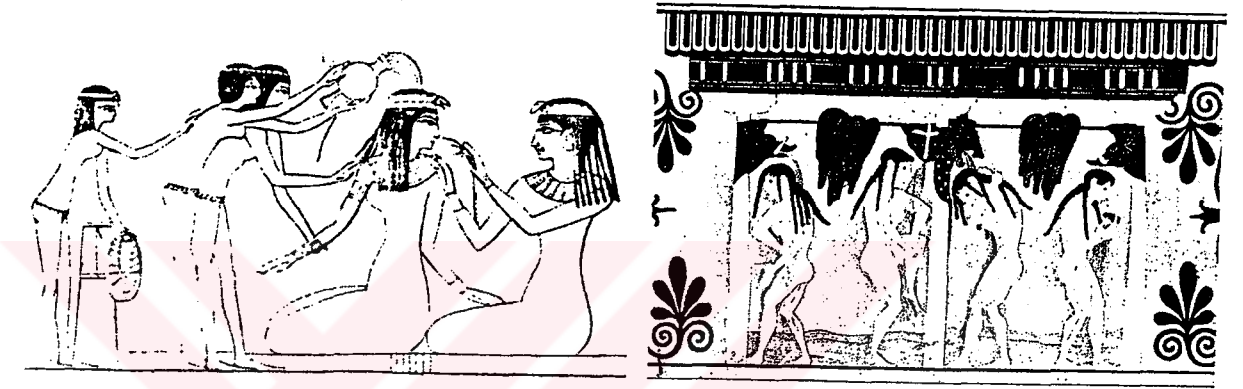
Bu bölüm altında vücut yıkama ihtiyacını karşılamak amacıyla geçmişte kullanılmış olan kurna ve günümüzde onun yerini almış olan küvet ve duş teknesi hakkında bilgi verilecektir.

- a. Kurna : Kurna genellikle mermerden yapılır. Boyutları 45X55 cm., yüksekliği ise 30 cm. civarındadır. Günümüzde terk edilmiş, yerini duş, küvet, vb. gibi çağdaş araçlar almıştır.
- b. Banyo Küveti : Dökme demir, emaye çelik, fayans, sert plastik, fiberglas ve suni mermerden üretilmiş modelleri vardır.
 - b.1. Ayaklı Küvet : Bunlar serbest duran küvetlerdir. Genellikle armatürleri alın tarafındadır. Çok yaygın olarak kullanılmazlar.
 - b.2. Çevresi Kapalı Küvetler : Serbest duran çevre duvarları küvetle beraber örülür.
 - b.3. Gömme Küvetler : Oturulan ve yatılan tipleri vardır. Dışarı taşan kenarları çevresindeki duvarı ve kaplamayı kavrar. Ön duvar tuğlayla örülür ve fayans, seramik, ya da mermerle kaplanır.
 - b.4. Köşe Küvetler : Mekandan tasarruf sağlar. Avrupa ve Amerika'da kullanımı daha yaygındır.

Bunların dışında çift kişilik küvetler, yuvarlak küvetler gibi değişik tipte küvetler de vardır. Bu tipler genellikle sert plastik veya fiberglastan yapılırlar.

- c. Duş Teknesi : Dökme demir, emaye, çelik, seramik, fayans, sert plastik, fibergals, veya mermerden duş teknesi yapılabilir. 70x70x16, 60x60x16, 90x90x18 cm ölçülerinde bulunabilirler. Ayrıca 75x90x28 cm

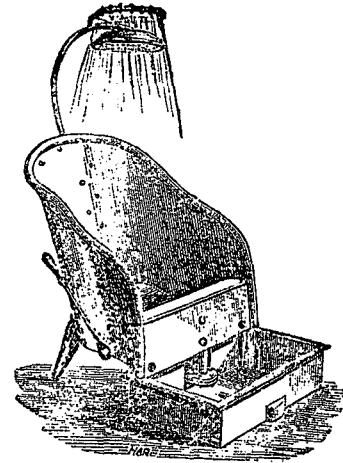
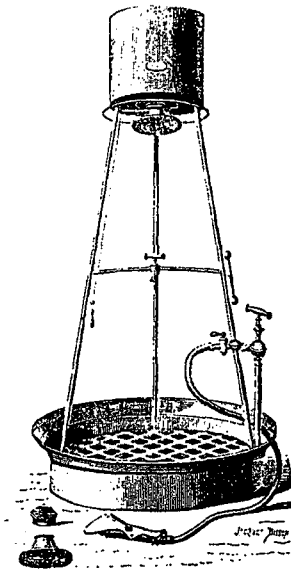
boyutlarında oturtmalı duş teknesi de vardır. Duş teknelerinde yıkanma zamandan ve sudan tasarruf sağlar. Ayrıca küçük boyutlarıyla yer kaybına neden olmadığından idealdir. Küvet bulunan banyolar da bile, mümkünse bir de duş teknesi koymak faydalıdır. Duş tekneleri suya dayanıklı perde, sekürit kapı veya paraduş denilen alüminyum çerçeve ve camdan oluşan bölücülerle çevrilerek bir duş kabini oluşturulur.



Şekil 1.7 Antik duş figürleri

Sol: Antik vazo üzerindeki bu desende kadınlar hayvan başlarından akan suyla duş almaktadırlar.

Sağ: Antik kadın vazodan akan su ile duş yapmaktadırlar.



Şekil 1.8 İlk duş tasarımları.

Sol: Griffith isimli bir tasarımcıya ait olan koltuk şeklindeki duş.

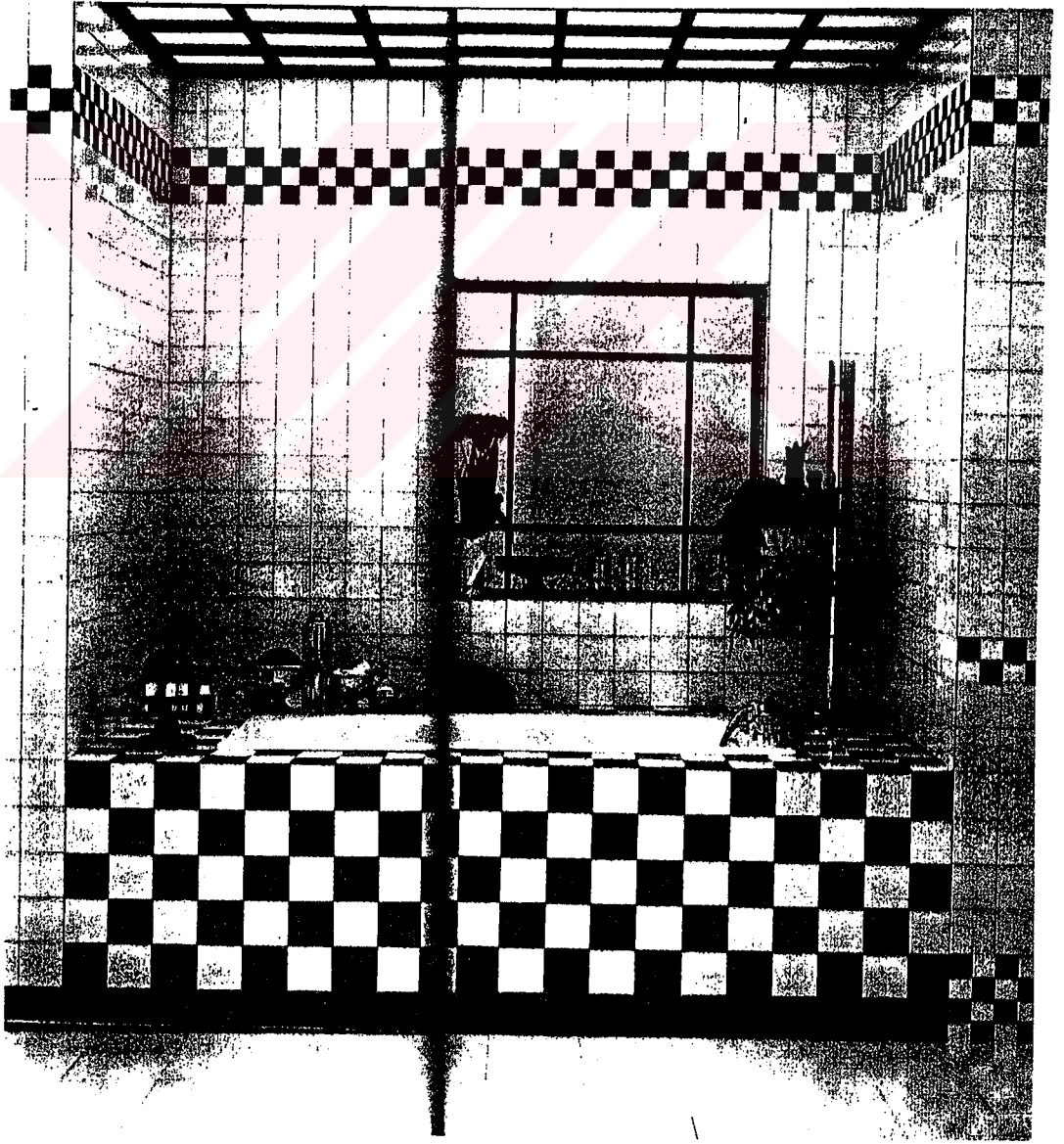
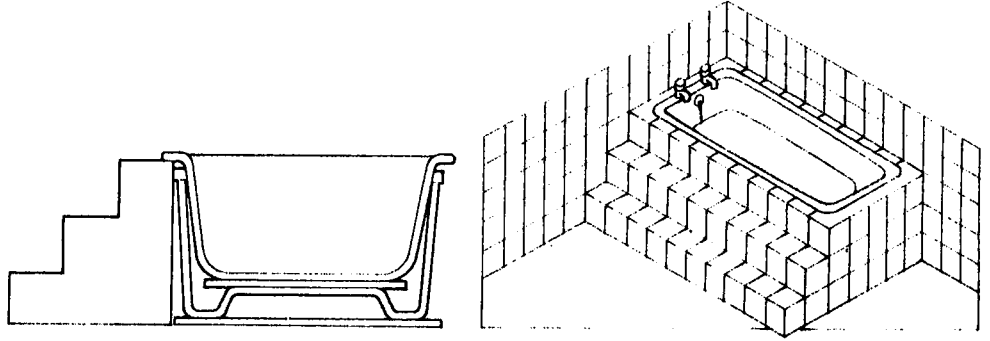
Sağ: 1889 yılına ait bu tasarım modern duşların ilk örneklerinden biridir.



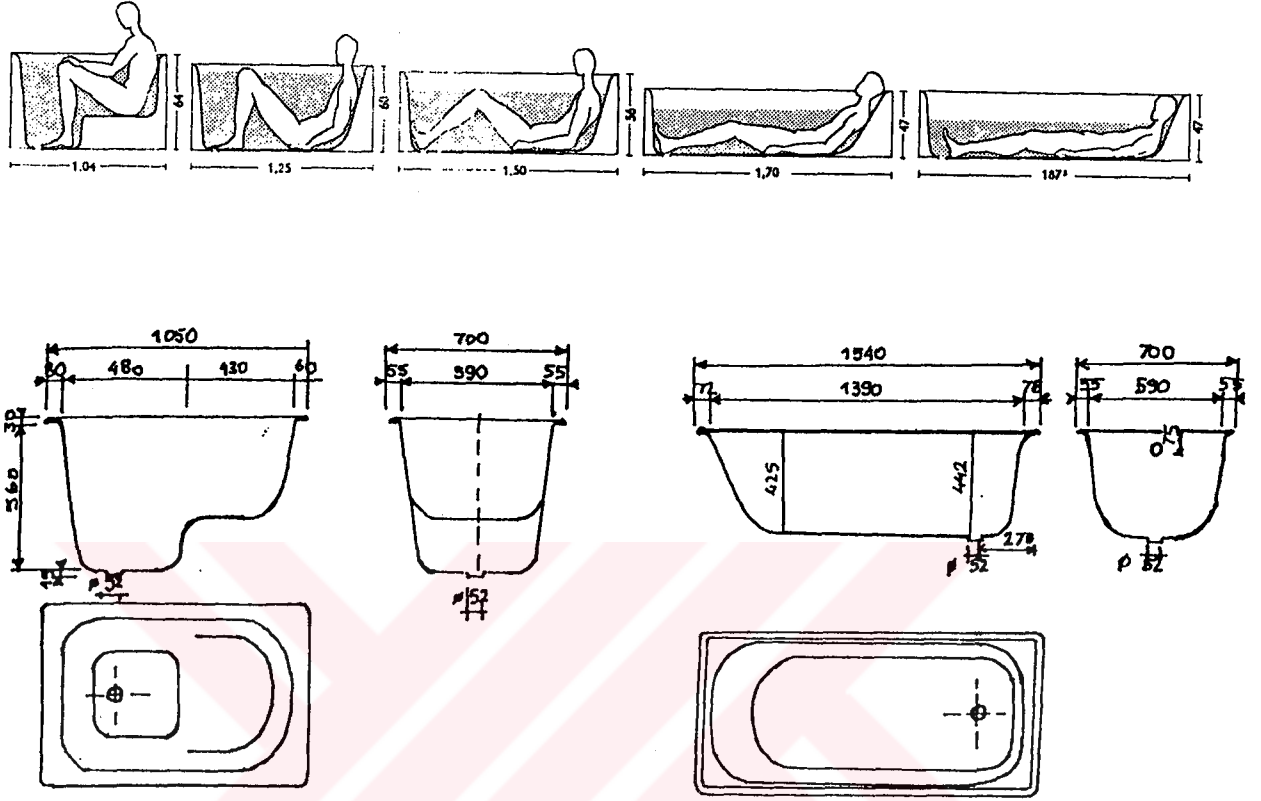
Şekil 1.9 Çevresi kapalı küvet uygulaması.



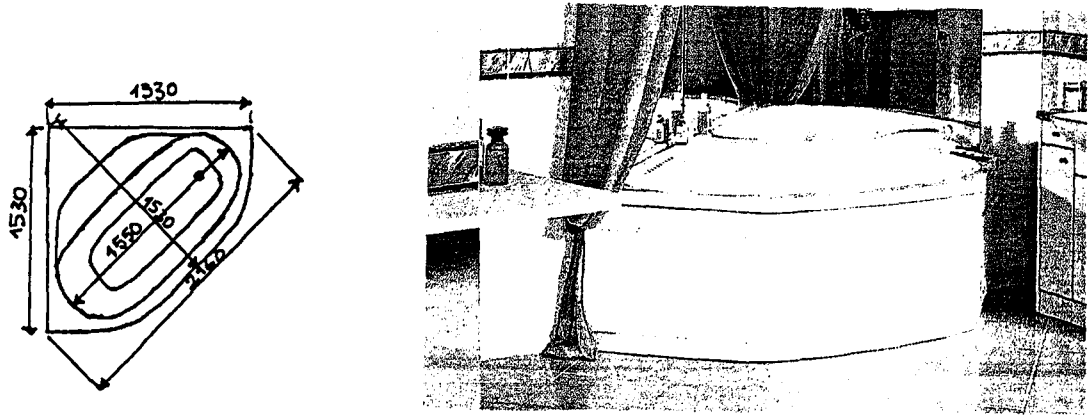
Şekil 1.10 Ayaklı küvet uygulaması.



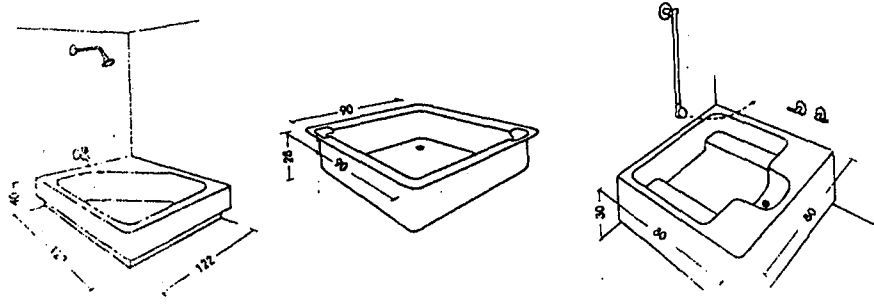
Şekil 1.11 Gömme küvet uygulaması ve tipik bir kesit.



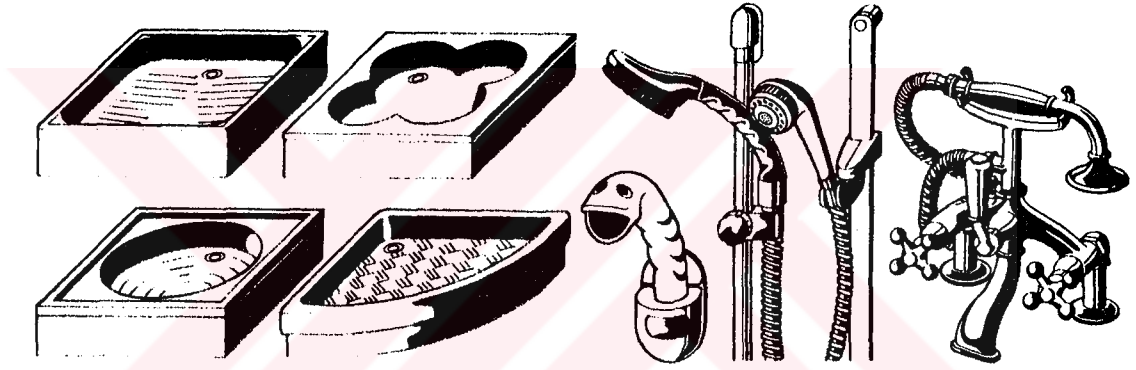
Şekil 1.12 Gömme küvetler için kullanım şekilleri ve ölçüler.



Şekil 1.13 Köşe küvetlere ait bir uygulama ve çok kullanılan ölçüleri.



Şekil 1.14 Standart duş boyutları.



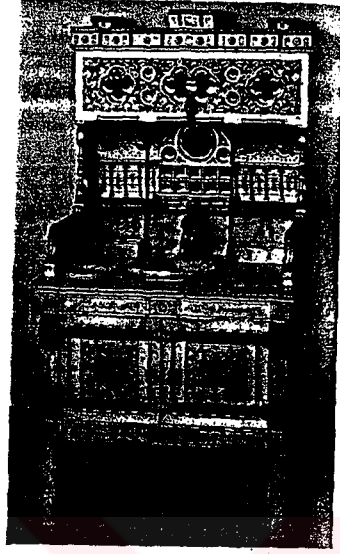
Şekil 1.15 Kullanımda bulunan bazı duş çeşitleri ve çok kullanılan duş aksesuarları.

1.3.2.2 Lavabolar

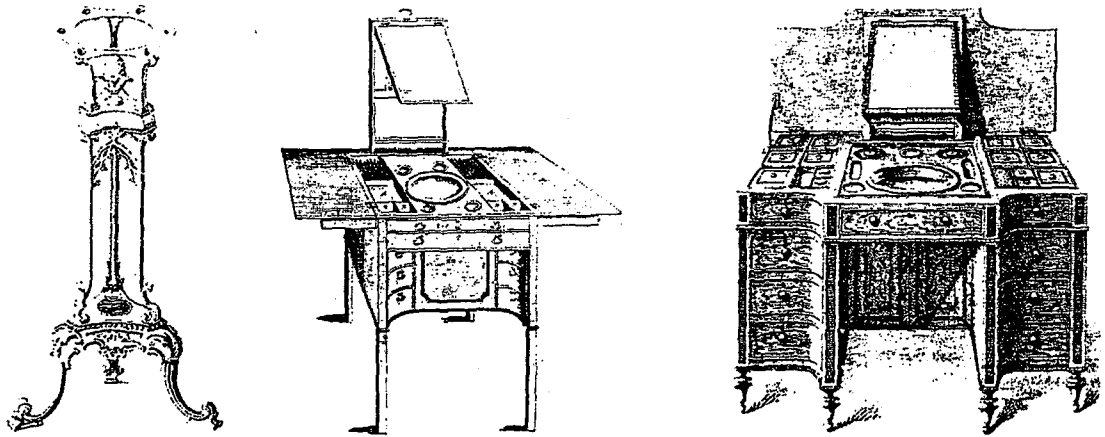
Paslanmaz çelik, seramik, sert plastik ve mermerden yapılmaktadır. Paslanmaz çelik daha çok mutfaklarda kullanılır, temizlenmesi ve kullanılması zordur. Çeşitli renk, ebat ve modellerde lavabolar yapılmaktadır.

- a. Duvara Asılan Lavabolar: Bunların en büyük dezavatajı boruları saklayamamalarından doğan estetik olmayan durumdur. Arkalıklı tipleri sıvalı duvarlarda kullanışlıdır. Arkalıksız tipleri yıkanabilir duvar önünde kullanmak şarttır, aksi takdirde su sıçratarak sıvayı bozar. Köşe lavabolar yerden tasarruf sağlarlar ve küçük mekanlar için idealdirler.
- b. Ayaklı Lavabolar : Lavaboyu duvara tutturup altına bir ayak koyarak boru tertibatını bunun içine saklanabilir. Lavabo ayakları yere kadar olabildiği gibi, yine duvara monte edilebilen yarım ayaklar da mevcuttur.
- c. Tezgaha Gömülen Lavabolar . Estetik kaygılardan dolayı boru tertibatını gizlemek için en iyi çözüm lavaboyu bir tezgah üzerine gömmektir. Böylece hem borular gizlenmiş olur, hem de temizliği daha rahat yapılabilir. Çeşitli banyo malzemeleri tezgah üzerinde durur ve eğer tezgah altına dolap konursa, havlu ve temizlik malzemeleri gibi banyo gereçleri burada depolanabilir. Lavabo tezgahları mermer, ahşap, seramik ve plastik esaslı malzemeden yapılabilir. Tezgah genişliği 63 - 65 cm arasında değişir. Fakat özel yapım küçük lavabolarda bu ölçü lavabonun boyutlarıyla orantılı olarak değişebilir. Tezgah yüksekliği 80 - 83 cm arasındadır.

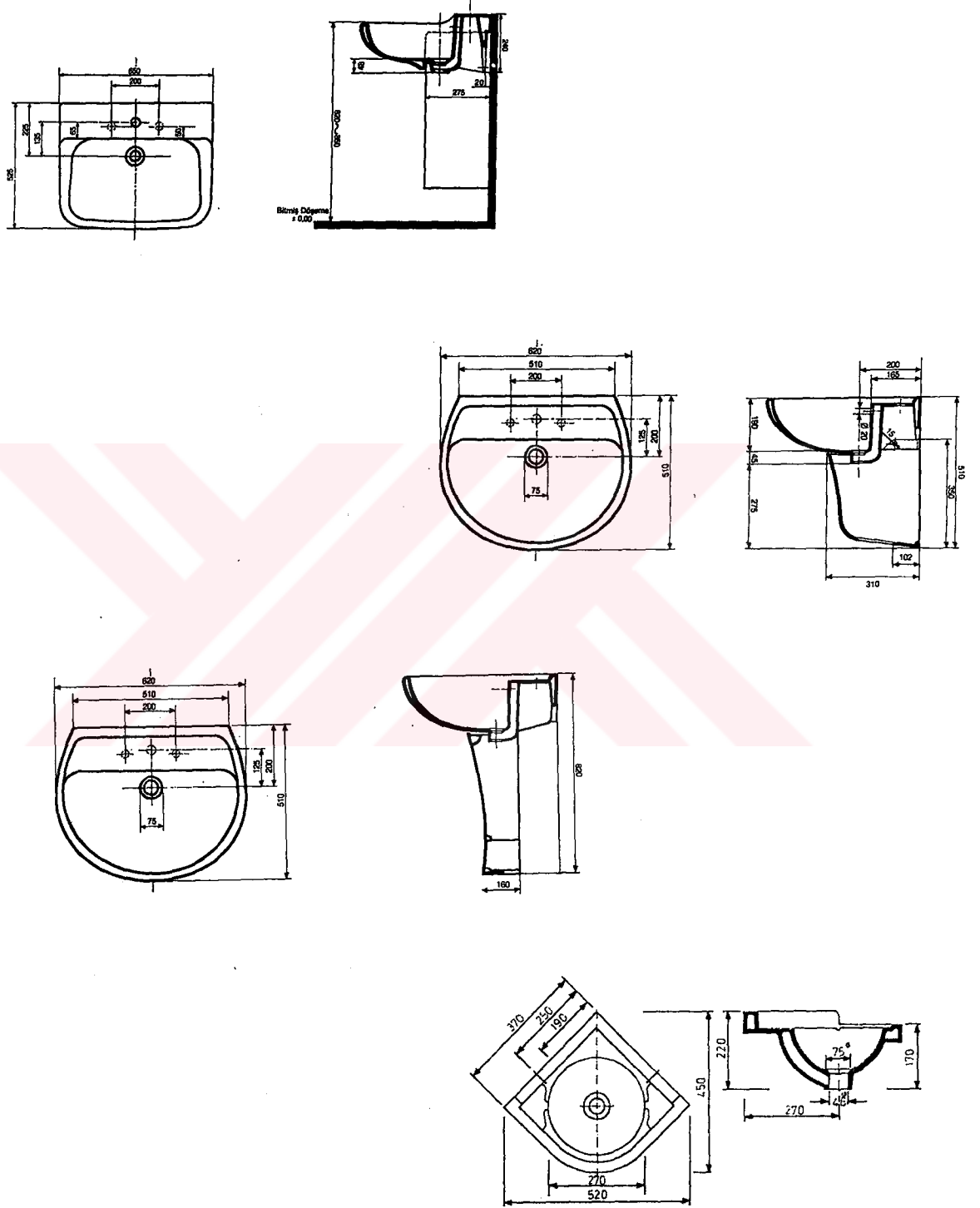
Özel Durumlar : Tekerlekli iskemlede oturmak zorunda olanlar veya çocuklar için normal standartlardaki lavaboyu kullanmak oldukça zordur. Bu durumlar için yüksekliği ayarlanabilir lavabolar tasarlanmıştır. Duvara monte edilebilen borular üzerinde lavabonun kaydırılmasıyla istenilen yükseklikte kullanım sağlanır:



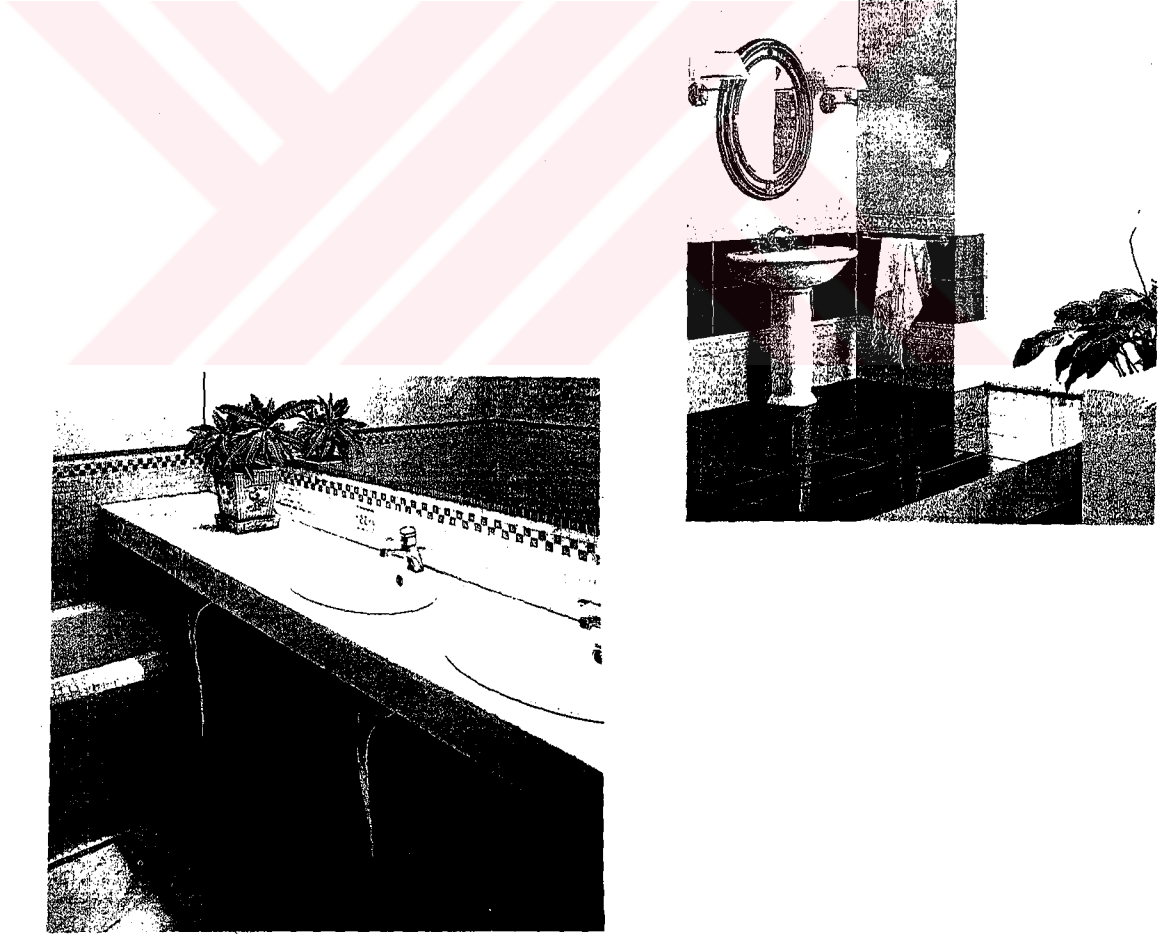
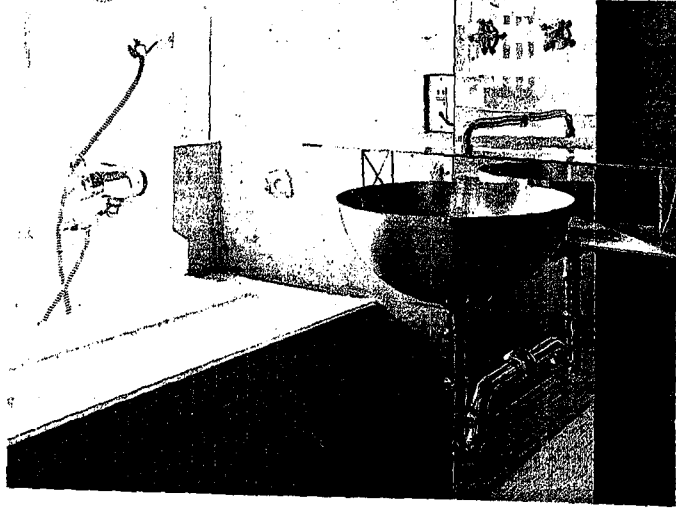
Şekil 1.16 Mimar William Burges (1827-81) tarafından gerçekleştirilmiş olan Viktorya tarzındaki el yıkama masası.



Şekil 1.17 Soldaki iki resim Chippendale tarafından 1754 tasarlanan el yıkama masalarına aittir; sağdaki resim ise Sheraton tarafından 1803 de tasarlanan el yıkamalı giyinme dolabına aittir.



Şekil 1.18 Sırasıyla ayaksız (duvara asılan), yarım ayaklı, tam ayaklı, ve köşe lavabolara ait örnek çizimler.



Şekil 1.19 Sırasıyla ayaksız, ayaklı, ve tezgaha gömülen lavabolara örnek uygulamalar

1.3.2.3 Tuvalet

- a. Hela Taşı : Fayans, mermer, font, emaye, ve mozaikten üretilebilirler. Genellikle 60x70, 60x50 cm ve 19, 23 cm yükseklikte olanları mevcuttur.
- b. Klozet : Klozetler genellikle porselen ve seramikten yapılırlar. Paslanmaz çelikten ve mermerden yapılan özel klozetlere de rastlanır. Klozet yüksekliği 38-40 cm civarındadır. Değişik model ve boyutlarda yapılmaktadır.

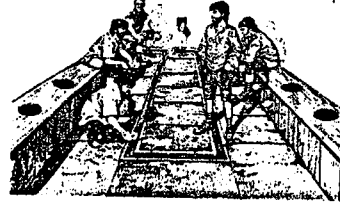
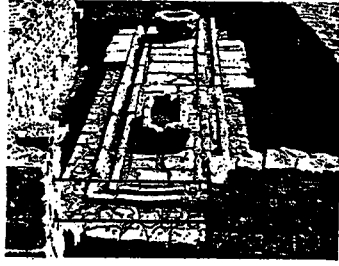
Klozetler pis su çıkışlarına göre sınıflandırılabilir:

Altan çıkışlı klozetler.

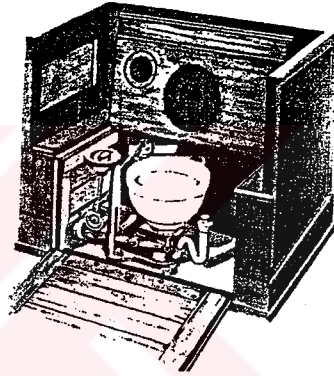
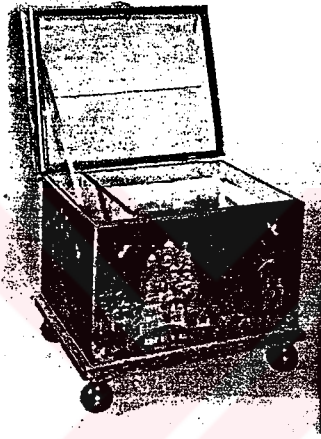
Arkadan çıkışlı klozetler : Bu tipin ayakları yere oturmayan modelleri de mevcuttur. Bunlar banyo zemininin kolay temizlenmesini sağlar.

Klozetler ayrıca ; rezervuarı klozetin arkasına takılan tipler, rezervuarı yükseğe asılan tipler, ya da rezervuarı duvarın içinde olan tipler olarak sınıflandırılabilir.

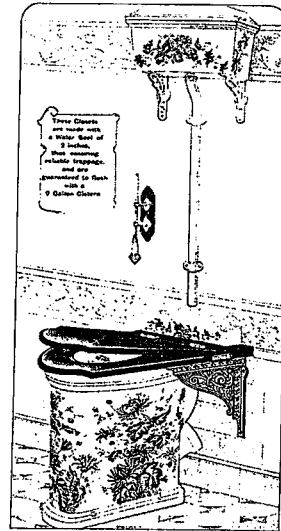
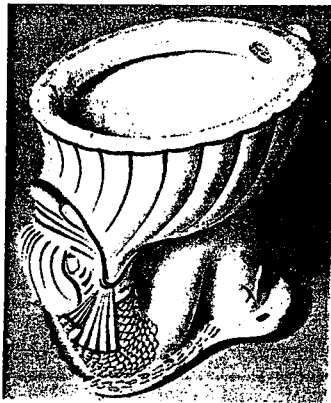
- c. Bideler: Bideler genellikle porselen veya seramikten yapılırlar. Mermerden yapılan özel imalatlara da rastlanır. Yerden yüksekliği 38-39 cm arasındadır. Duvara asılan ve döşemeye oturan tipleri vardır. Suyu altan duş şeklinde verebilir veya lavabo musluğu gibi musluğu olabilir.



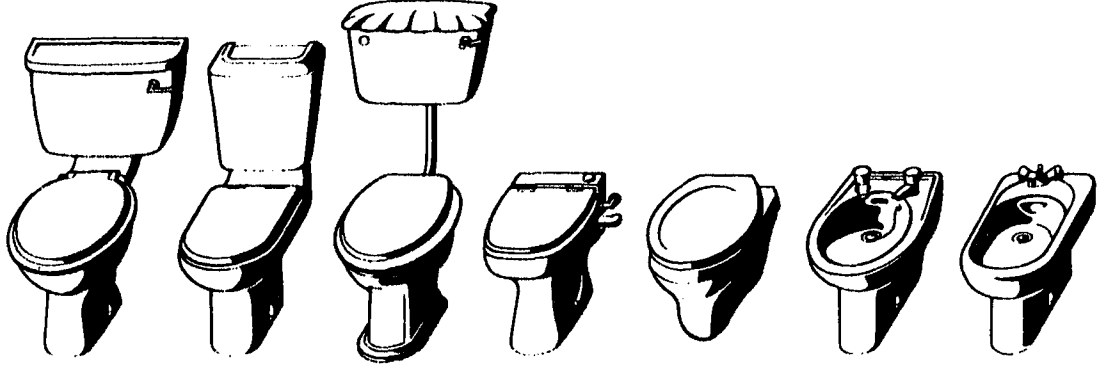
Şekil 1.20 Roma dönemine ait tuvalet kalıntıları ve yine Roma dönemi tuvalet kullanımını anlatan temsili resim görülmektedir.



Şekil 1.21 Solda 17 yy a ait bir klozet, sağda ise 1778 de tasarlanan ve 1890 yıllarına başarıyla kullanılan bir klozet tasarımı görülmektedir.



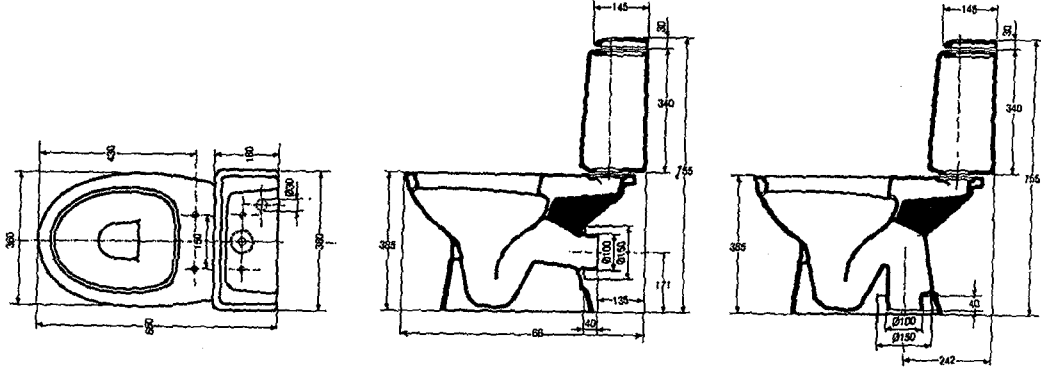
Şekil 1.22 Solda 1880 yılına ait yunus şekilli bir klozet, sağda ise 1900 yılına ait özel bir tasarım görülmektedir.



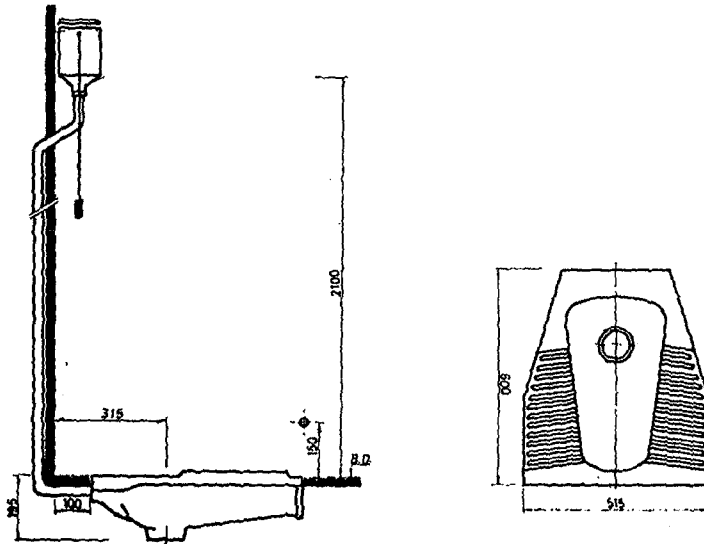
Şekil 1.23 Çeşitli özelliklere sahip klozet ve bideler.



Şekil 1.24 İtalya'dan bir uygulama.



Şekil 1.25 Gideri alttan ve arkadan olan klozetler.



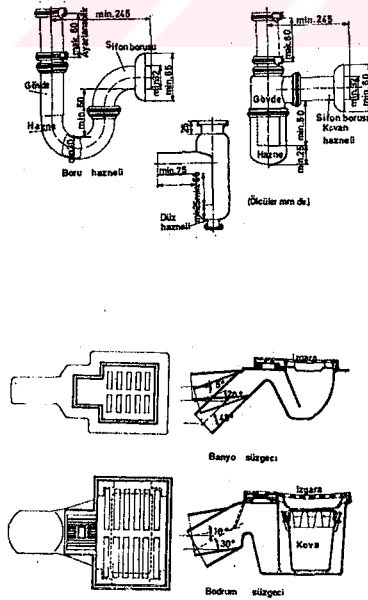
Şekil 1.26 Standart alturka boyutları.

1.3.2.4 Sifon ve Süzgeçler

Pis su boruları duvar veya döşeme içine yerleştirilirler. Bu borular kanalizasyona bağlandıkları için sıhhi tesisat borularından pis kokular bina içine gelebilir, ya da sıhhi tesisat donatılarından bir çok yabancı madde borulara kaçarak tıkanmaya sebep olabilir.

Bunları önlemek için ; her sıhhi tesisat donatısı kirli suyla bir "S" veya sifon aracılığıyla ilişkide bulunur. Döşeme kirli suları ise, yer süzgeçleriyle pis su kolon ve yatay ana borularına bağlanır. S'ler ve sifonlar içlerinde daima belli bir yükseklikte su bulundurarak kokunun geri gelmesini engellerler. S'ler plastik, pik döküm, sarı maden, ve kurşundan yapılır. Klozet ve hela taşlarında Ø 100 lik S, diğer sıhhi tesisat donatılarında Ø 70 lik S kullanılır.

Sifonlar sarı madendendir, ve üzerleri nikelaj veya kromaj yapılır. 1 ve ¼ inçlikleri vardır. Yer süzgeçleri kare veya daire kesitli olabilir. 10x10 cm , 17x17 cm arasında değişen ebatları bulunabilirler. Yer süzgecinin borusu Ø 70 dir.



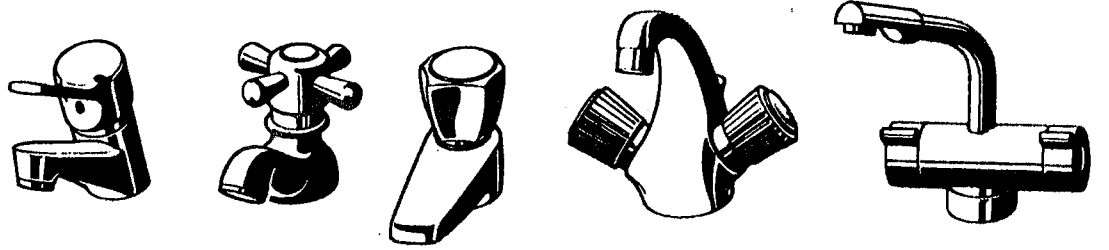
Şekil 1.27 Sifonlar ve süzgeçler.

1.3.2.5 Bataryalar

Sarıdan yapılp üzeri kromaj veya nikelajlanan bataryalar olduđu gibi, altın kaplama veya plastik bataryalar da mevcuttur.

- a. Lavabo Bataryası : Duvara veya lavabonun üzerine monte edilen tipleri vardır. Kendi eksenini etrafında dönen modeller de bulunmaktadır.
- b. Duş Bataryası : Gagalı duş bataryası ve elde kullanılan sipiralli duş bataryaları da kullanıma sunulmuştur.
- c. Banyo Bataryası : Gömme küvete ve duvara monte edilen türleri vardır.

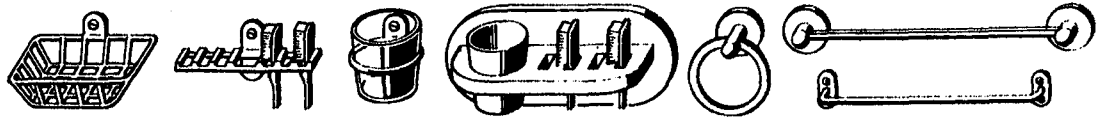
Tüm bataryalarda sağa soğuk su, sola sıcak su bağlanır.



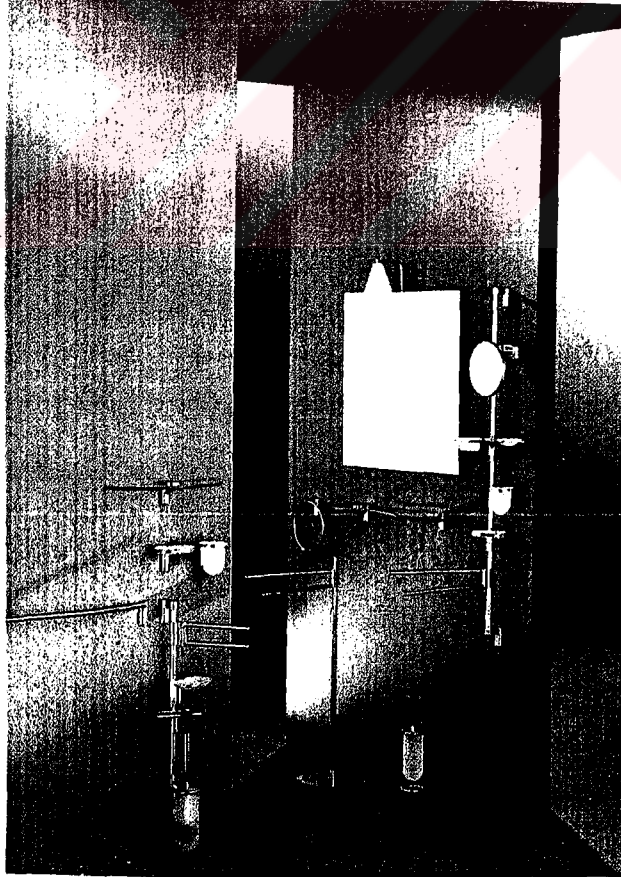
Şekil 1.28 Bataryalar.

1.3.2.6 Aksesuarlar

Lavabo etajeri, el tutamağı, havluluk, tuvalet kağıdı silindiri, sabunluk, diş fırçalığı, vb. gibi bir çok aksesuar piyasada ahşap, plastik, krom kaplı metal, pirinç, ya da altın kaplamadan imal edilmiş olarak bulunmaktadır.



Şekil 1.29 Çeşitli aksesuar çizimleri.



Şekil 1.30 Carlo Bartoli den bir aksesuar serisi.

1.3.2.7 Dolaplar

Dolaplar son yıllara kadar banyolarda çok sık rastlanmayan elemanlardı. Ancak ecza dolabı niteliğinde, lavabonun üzerine asılan, kapağı aynalı olup aynı zamanda ayna vazifesi de gören dolaplara rastlanırdı. Fakat son zamanlarda mutfaklarda kullanılan tezgahlar ve dolaplar banyolarda da kullanılmaya başlandı. Bu dolaplar hem banyo eşyalarının düzenli olarak saklanması, boruların kaplanması, hem de konutun diğer odalarıyla banyo arasındaki ses yalıtımını sağlamaları bakımından faydalıdır.

Tezgaha lavabonun gömülmesi aşamasında izolasyona dikkat edilmelidir. Ayrıca dolapların altında 6-10 cm arasında değişen ölçülerde baza yapılır. Bu, zemindeki suyun dolabın içine girmesini ve kapaktaki herhangi bir sarkma dolayısıyla kapağın yere çarpmasını önlemek için önemli bir ayrıntıdır. Baza, dolabın malzemesinden olabileceği gibi yer kaplamasının malzemesinden de olabilir.

Aynanın ve lavabonun aydınlatma sorununu çözmek için lavabonun üzerine gelen bölümde çeşitli şekillerde aydınlatma düşünülmelidir.

Banyo dolabında kullanılan malzemenin suya dayanıklı olması dayanım süresi açısından önemlidir. Polyester esaslı malzemeler tercih edilmelidir. .

1.3.2.8 Sıcak Su

Binaların sıcak suyu ya ortak bir merkezden sağlanır, ya da katlarda ayrı ayrı su ısıtıcı araçlar kullanılarak elde edilir. Banyo, duş, vs. için 40 derece, çamaşır yıkama için 60 derece suya ihtiyaç vardır. Konutlar için sıcak su sağlayan araçlar şunlardır:

- a. Şofbenler : Havagazı şofbeni, likit gazla çalışan şofben ve elektrikli şofben olmak üzere üç yaygın tip vardır. Şofbenler banyoyu ısıtamazlar, sadece suyu ısıtırlar. Şofbenler genellikle duvara asılarak monte edilirler. Havagazı ve likitgazla çalışan şofbenlerden çıkan yanmış gazlar ve CO₂

gazı için mutlaka şofbenin bacaya bağlanması gerekir. Elektrikli şofbenler bacaya ihtiyaç duymazlar.

b. Termosifonlar :

Bakır Termosifonlar : Odun, kömür veya sıvı yakıt yakan tipleri vardır. Termosifonlar piyasada 31, 33, 35 cm çapta bulunurlar, yükseklikleri yaklaşık olarak 150 cm dir.

Otomatik Termosifonlar : Gazla çalışırlar. Çapları Ø48. cm ve yükseklikleri 150 cm dir. Termosifonlar banyoyu da ısıtırlar. Bacaya bağlanmaları şarttır.

c. Merkezi Sıcak Su Tesisatı : Bu tesisat yakıtın yandığı bir kazan, kazanda ısınan suyun veya buharın ısınıp sıcak suya veren boyler, ve borulardan oluşur. Dikkat edilmesi gereken noktalar şunlardır:

1. Yaygın ve büyük tesisatlarda dolaşımı hızlandırmak için bir dolaşım pompası kullanılır.
2. Boyler ve sıcak su borularının yalıtılmaları gerekir.
3. Sıcak su boruları suyun kolay dolaşmasına izin verecek şekilde döşenmelidir.

1.3.2.9 Isıtma

Konut merkezi sistemle ısıtılıyorsa yıkanma-yıkama mekanı da aynı sistemle ısınacaktır. Tercih edilmesi gereken sistem de budur. Radyatör seçimi için piyasada oldukça geniş bir yelpaze bulunmaktadır. Eğer fazla yer kaybetmek istenmiyorsa, panel radyatör seçilebilir. Bir de döşemeden ısıtma sistemi vardır. Bu sistemde yer kaplama malzemesi olarak genellikle mermer tercih edilir. Bu sistemin en büyük sorunu bir arıza durumunda bazen bütün döşemenin sökülmesini gerektirmesidir. Fakat bunun yanında homojen bir şekilde ısınmış bir döşeme büyük bir rahatlaktır. Bunu da en iyi döşemeden ısıtma sağlar. Eğer merkezi ısıtma yoksa su ısıtıcısı olarak termosifon çeşitlerinden birini seçerek bununla suyu ısıtırken aynı zamanda banyonun da ısınması sağlanabilir.

1.3.3 Boyutlar ve Alanlar

Çeşitli araçlar ve bunlara takılacak bataryaların yüksekliği yapılış şekline göre biraz değişse de, genellikle aşağıdaki gibidir :

Lavabolar	80 cm
Lavabo bataryaları (duvara takılı)	105 cm
Lavabo üzerindeki etajer	125-130 cm
Lavabo üzerindeki aynanın alt kenarı	135 cm
Küvet üst kenarı	49-52 cm
Banyo bataryaları	65-75 cm
Duş süzgeç borusu	220 cm
Klozet üst kenarı	39-41 cm
Alaturka rezarvuar alt kenarı	210-220 cm
Alafranga rezarvuar alt kenarı	190 cm
Rezervuar çekme kolu	135 cm
WC. kağıtlık	65-70 cm
Alaturka taharet musluğu	20-25 cm
Alafranga taharet musluğu	40 cm
Otomatik rezervuar (bastip)	110 cm
Duşlu bide üstü	39 cm
Duşsuz bide üstü	37,5 cm

Yıkama mekanı 3.00 m² den, en dar kenarı da 1.20 m den az olmamalıdır. Aynı mekanda çamaşır yıkama işlemi de yapılacaksa 4.50 m² den ve en dar kenarı da 1.80 m den kısa olmamalıdır. Dikdörtten çözülmüş yıkama mekanları daha optimum çözümler olarak kabul edilirler.

1.3.4 Kaplama Malzemeleri

Banyo kaplama malzemeleri mutfak için kullanılan malzemelerden çok farklı değildir. Her iki durumda da ıslaklık, sıcaklık, ve devamlı kullanım vardır. Banyonun evin iç kısımlarında, çamur, toz, ve yağdan uzak olması daha temiz kullanılmasını sağlar.

1.3.4.1 Döşeme Kaplamaları

Banyolar genellikle çıplak ayak veya terlikle kullanıldığından ıslanmanın dışında döşemenin yıpranmasına yol açacak fazla bir etken yoktur.

- a. Ahşap Kaplama : Banyo için çok alışılmış ve kullanışlı bir malzeme değildir. Fakat gerekli bakım düzenli olarak yapılırsa fazla sorun çıkarmaz. Ayağa sıcak gelen, kaygan olmayan ve belli bir ıslaklığa dayanabilen bir malzemedir.
- b. Muşamba : Kaliteli bir muşamba banyo için kullanılabilir bir malzemedir. Estetik olarak çok gösterişli olmadığından pek tutulmaz.
- c. Mantar Tuğlalar : Bu malzeme plastik kaplı olarak veya olduğu gibi kullanılabilir. Plastik kaplı olanlar suya karşı dayanıklıdır. Kaplı olmayanlar ise verniklenerek kullanılabilirler.
- d. Seramik : Seramik estetik etkisi dolayısıyla banyolarda çok kullanılan bir malzemedir. Suya dayanıklılığı ve temizleme kolaylığı açısından tercih edilir. En büyük sakıncası ıslakken kaygan olmasıdır.
- e. Mermer : Mermer döşeme banyolarda çok kullanılır. İstenilen cins ve boyutta kesilebilir. Estetik değerinin yanında suya dayanıklılığı ve temizleme kolaylığı da vardır. Cilası zamanla bozulsa da tekrar cilalanma imkanı olduğundan uzun süre kendini koruyabilen bir malzemedir. Mermer yer döşemesinde alttan ısıtma imkanı da vardır.
- f. Mozaik Döşeme : Eskiden çok kullanılan bu döşeme kaplaması artık, eskisi kadar estetik bulunmadığından kullanım potansiyelini yitirmiştir. Mozaik döşeme uygulaması, mermer kırıklarının çimentoyla karıştırılarak ya, önceden hazırlanmış plakalar halinde, ya da yerinde dökülerek yapılmaktadır. Maliyet açısından hesaplıdır, fakat mat bir görünümü vardır ve kir tutar.

g. Halı Kaplama : Halı kaygan olmayan, sıcak, sessiz ve yumuşak bir yüzey sağlaması açısından tercih edilen bir malzemedir. Banyolar için sentetik liflerden imal edilmiş olanları tercih edilmelidir. Halı kaplamanın en büyük problemi su ve temizlik sorunudur. Klozetin çevresindeki halı döşemeye özellikle dikkat edilmelidir, gerekirse klozetin önüne ikinci bir paspas konulmalıdır.

1.3.4.2 Duvar Kaplamaları

- a. Boya : Boya kullanılacaksa duvarların yüzeylerinin düzgün olması gerekmektedir. Ucuz olması, çok sayıda renk olanağının bulunması, ve kolay uygulanması nedeniyle tercih edilir. Yağlıboya güzel görünümüne rağmen sudan etkilenmesi nedeniyle banyolar için kullanışlı değildir. Normal boya ise, buharı hemen emdiği için sudan çok etkilenmez ama üzerinde oluşan sabun ve yağ lekeleri kolay çıkmaz. Duş, küvet, ve lavabonun etrafına özel bir kaplamaya ihtiyaç duyulur.
- b. Duvar Kağıdı : Duvar kağıdı da boya gibi düz yüzeylere uygulanmalıdır. Pahalı olmayan ve buharı olduğu gibi emen bir kaplama malzemesidir. Fakat fazla nemli ortamlarda duvar kağıdı kısa sürede kabarır ve düşer. Plastik esaslı duvar kağıdı nem sorununu ortadan kaldıracığından daha kullanışlı olabilir. Burada da boyada olduğu gibi duş, küvet, ve lavabonun çevresinde özel bir kaplamaya ihtiyaç duyulur.
- c. Ahşap Kaplama : Boyanmaya ve kağıt kaplanmaya hazır olmayan bozuk yüzeyli duvarların gizlenmesinde geçmeli veya oluklu ahşap kullanılabilir. Ahşabın buhara veya neme dayanıklı hale gelebilmesi için bir kaç kat vernik çekilmesi gerekir. Verniğin dışında lake veya polyester cila da yapılabilir. Bu uygulamalar temizlik açısından daha kullanışlıdır. Ahşap kaplama estetik açıdan diğerlerine göre daha tercih edilir bir kaplama türüdür.
- d. İnce Plastik Tabakalar : Sudan ve buhardan etkilenmedikleri için rahatlıkla kullanılabilecek malzemelerdir.

- e. Mantar Tuğlalar : Yumuşak, çeşitli doğal renklerde, buharı emen, akustik problemi olmayan, ucuz bir duvar kaplamasıdır. Çok su sıçrayan yerlerde plastikle kaplanmış çeşitleri kullanılmalıdır. Çok yaygın bir duvar kaplama malzemesi değildir.
- f. Sentetik Kauçuk : Akustik açıdan kullanışlıdır. Çeşitli renkleri bulunur. Tozu göstereceğinden arada bir vakumlu bir süpürgeyle temizlenmesi gerekir. Çok yaygın bir duvar kaplama malzemesi değildir.
- g. Mermer : Duvar için lüks ve pahalı bir kaplama malzemesidir. Fayans gibi ince kesilmiş ve arkası harcı tutması için oluklu işlenmiş mermer duvar kaplamaları da üretilmektedir. Kullanışlı bir malzemedir.
- h. Seramik : En pratik ve en çok kullanılan duvar kaplamasıdır. Çok çeşitli renk ve desenleri bulunur. Montajı sırasında harcı ve uygulaması iyi yapılırsa, kaplama iyi olursa yıllarca su geçirmezliğini ve parlaklığını korur.
- i. Ayna ve Cam Tuğla : Çeşitli boyutlarda uygulanabilir. Banyonun su damlalarından korunması gereken köşelerinde pratik ve dekoratif bir satih olarak olarak kullanılabilirler.

1.3.4.3 Tezgahlar

- a. Verzalit : Yeni olmasına rağmen çok kullanılan bir tezgah kaplama malzemesidir. Kolay şekil verilebildiğinden, ve rahatça yuvarlak köşe dönüşleri yapabildiğinden tercih edilir. Duvarla çok iyi temas ettiğinden su geçirmez bir yüzey oluşturur.
- b. Sert Ahşap : Bir kaç kat vernik ve polyester esaslı cila atılarak sudan korunabilir. Lavabonun tezgaha gömülme detayı iyi çözülebilirse kir ve nemin birikmesi önlenir. Dikkat edilir ve belirli sürelerde gereken bakım yapılırsa ahşap tezgah kullanışlı ve dekoratif olabilir.

- c. **Seramik :** Gömme lavabo ve küvetlerin etrafında çok kullanışlı ve dekoratif bir yüzey oluştururlar. Sabun lekeleri ve diğer pisliklerin kolayca temizlenmesi kullanım kolaylığı getirir. Uzun ömürlü ve kullanışlı bir malzemedir. Bu özelliklerinden dolayı banyolarda tezgah ve küvet alınlarnın kaplanmasında en çok kullanılan malzemelerden biridir.
- d. **Mermer :** Leke tuttuğu için mutfak tezgahlarından çok banyolarda kullanılan bir malzemedir. Uygulanışı 2-3 cm lik plakaların kullanımıyla olur. Tezgahın boyutlarına bağlı olarak tek parça ya da çok parçalı olarak kullanılabilir. Genelde tek parçadan yapıldığından düzgün bir yüzeye sahip olur. Banyolarda en yaygın kullanıma sahip tezgah malzemelerinden biridir.



BÖLÜM 2

METOD

2.1 Programlama Dili ve Ortamı

Yazılım, günümüz koşullarında standart sayılabilecek bir PC kullanılarak geliştirilmiştir. Geliştirilecek US, çıktılarını tefrişi yapılmış banyo planları olarak vereceği için, seçilecek programlama dili ya da aracının bu tarz bir sunuma olanak veriyor olması gerekmektedir. Bu gereklilik göz önünde bulundurularak programlama dili olarak AutoLISP seçilmiştir.

Bu dilin seçilmesinde iki önemli etken rol oynamıştır: AutoCAD programı altında çalışması, ve LISP kökenli olması. AutoCAD programı, bilgisayar destekli tasarım ya da çizim konusunda milyonlarla ifade edilen kullanıcı kitlesi tarafından yararlanılan bir programdır. Ülkemizde de yaygın bir kullanıma sahiptir. Piyasada daha güçlü ve bu alana daha uygun programlar bulunmasına rağmen, mimarlık alanında AutoCAD kabul görmüş bir programdır. AutoCAD'in bu başarıyı sağlamasında, konusunda (PC CAD pazarında) ilkler arasında bulunması kadar, açık bir program olması da etkili olmuştur. Öyle ki, AutoCAD kullanıcısı kadar çok olmasa da, AutoCAD programcısı adı altında kalabalık bir grup vardır. Bu programcılar AutoLISP dilini kullanarak AutoCAD programının tüm fonksiyonlarını kullanabildikleri gibi, LISP dilinin üstün özelliklerini kullanarak çok güçlü algoritmalar geliştirme şansına da sahiptir.

2.1.1 AutoLISP

AutoLISP, David Betz tarafından geliştirilen XLISP dilinden uyarlanarak ve AutoCAD programına ait bir çok özellik eklenerek geliştirilmiştir (AutoDesk, 1988). AutoLISP liste işlemek temelinde tasarlanmış bir programlama dilidir. Listeler şeklinde saklanan bilgiyi ve veriyi yönetmek için programcıya birçok fonksiyon sunar. Bir diğer önemli özelliği ise, programın kendi koduna karşı aynı veriye olduğu gibi davranabilmesidir. Programın kodu program işletilirken değiştirilebilir, yeni fonksiyonlar eklenebilir.

Örneğin

```
(if (and (= A 1) (= C 3)) (setq B 2))
(if (and (= A 3) (= C 2)) (setq B 4))
(if (= A 0) (setq Z 2))...
```

şeklinde girilen koşul satırlarına program çalışırken yeni bir koşul eklemek ya da bir koşulu ortadan kaldırmak olanaklı değildir (teknik olarak mümkündür, fakat kullanışlı değildir). Oysa AutoLISP dilinin imkanları kullanıldığında;

```
(setq X "(if (and (= A 3) (= C 2)) (setq B 4))" )
```

şeklindeki bir komut satırını kullanarak, tıpkı bir ismi bir değişkene atar gibi, kural olmaya aday bir cümle bir değişkene atanabilir. Bu cümle üzerinde herhangi bir "string" ile oynuyor gibi oynanabilir:

```
(setq Y1 (substr X 1 5))
(setq Y3 "or (= K")
(setq Y2 (substr X 13 25))
(setq Y (strcat Y1 Y2 Y3))
(eval Y)
```

şeklinde bir komut dizisiyle yeni bir koşul yaratılabilir. Burada (eval Y) tam olarak;

```
(if (or (= K 3) (= C 2)) (setq B 4))
```

komut satırıyla aynı işleve sahiptir.

Bu özellik bu çalışmanın en önemli görevini yerine getirmektedir. Kural tabanının kullanıcıya açık, ve özelleştirilebilir olması bu özellik sayesinde mümkün olabilmektedir.

Yeni bir fonksiyon tanımlamanın bilinen sözdizimi aşağıdaki gibidir:

```
[1] (defun self () (print "satir 1"))
```

Fakat LISP in kendi kodunu bir liste gibi algılamasından yararlanılarak bu fonksiyon aşağıdaki gibi tanımlanabilir:

```
[2] (setq self '(()(print " satir 1")))
```

Burada ([2]) self fonksiyonunun yukarıda ([1]) tanımlanan self fonksiyonundan farksızdır. Her iki şekilde de tanımlanan fonksiyonlara aşağıdaki ([3]) işlemi uygulayabiliriz:

```
[3] (setq self (append self '((print "satir 2"))))
```

Bu fonksiyonun (append list1 exp : list1 listesinin sonuna exp1 elemanını ekler) uygulanmasıyla birlikte artık self fonksiyonu aşağıdaki fonksiyona ([4]) hale gelmiştir (Spiler ve Marincek ,1995):

```
[4] (defun self ()
      (print "satir 1")
      (print "satir 2")
    )
```

LISP programlama dilini US çalışmalarında aranılır hale getiren bu iki önemli özelliktir.

2.2 Programlama Metodu

Problem dikdörtgen bir banyo mekanı ve banyolarda en çok kullanılan elemanlar seçilerek sınırlanmıştır. Tasarımda kullanmak üzere 5 eleman grubu seçilmiştir. Lavabo, klozet, çamaşır makinası, yıkanma, ve sıcak su sağlama. Yıkanmaya ait duş teknesi ve küvet şeklinde iki seçenek bulunur. Sıcak su için ise şofben ya da termosifon arasında seçim yapılabilir. Kullanıcı bu elemanlardan kullanmak istediklerini seçer. Seçilen her eleman grubu için birden fazla uygulanabilir tasarım seçeneği bulunur. Aşağıda olası tüm tasarım durumlarının sayısal bir dökümü verilmiştir. Tasarım problemini çözülebilir hale getirmek için, üretilen olasılıklar, tek tesisat duvarı kullananlar, ve iki tesisat duvarı kullananlar olarak ikiye ayrılmışlardır. Bu sınıflamaya dayanarak türetilen olasılıklardan, tek duvarlı olasılıklar, ve iki duvarlı olasılıklar şeklinde bahsedilecektir. Tüm kombinezonların açıklamalı dökümü Ek. A da verilmiştir.

- 5 elemanlı, tek duvarlı 4 kombinezon
her kombinezon için $5 \times 4 \times 3 \times 2 \times 1 = 120$ olasılık
 $4 \times 120 = 480$ adet 5 elemanlı olasılık bulunur.
- 4 elemanlı, tek duvarlı 16 kombinezon
her kombinezon için $4 \times 3 \times 2 \times 1 = 24$ olasılık
 $16 \times 24 = 384$ adet 4 elemanlı olasılık bulunur.
- 3 elemanlı, tek duvarlı 25 kombinezon
her kombinezon için $3 \times 2 \times 1 = 6$ olasılık
 $25 \times 6 = 150$ adet 3 elemanlı olasılık bulunur.
- 2 elemanlı, tek duvarlı 19 kombinezon
her kombinezon için $2 \times 1 = 2$ olasılık
 $19 \times 2 = 36$ adet 2 elemanlı olasılık bulunur.

Tek duvar kullanımı durumunda toplam 1032 olasılık sözkonusudur. Kullanılan duvar sayısı 2 ye çıktığında,

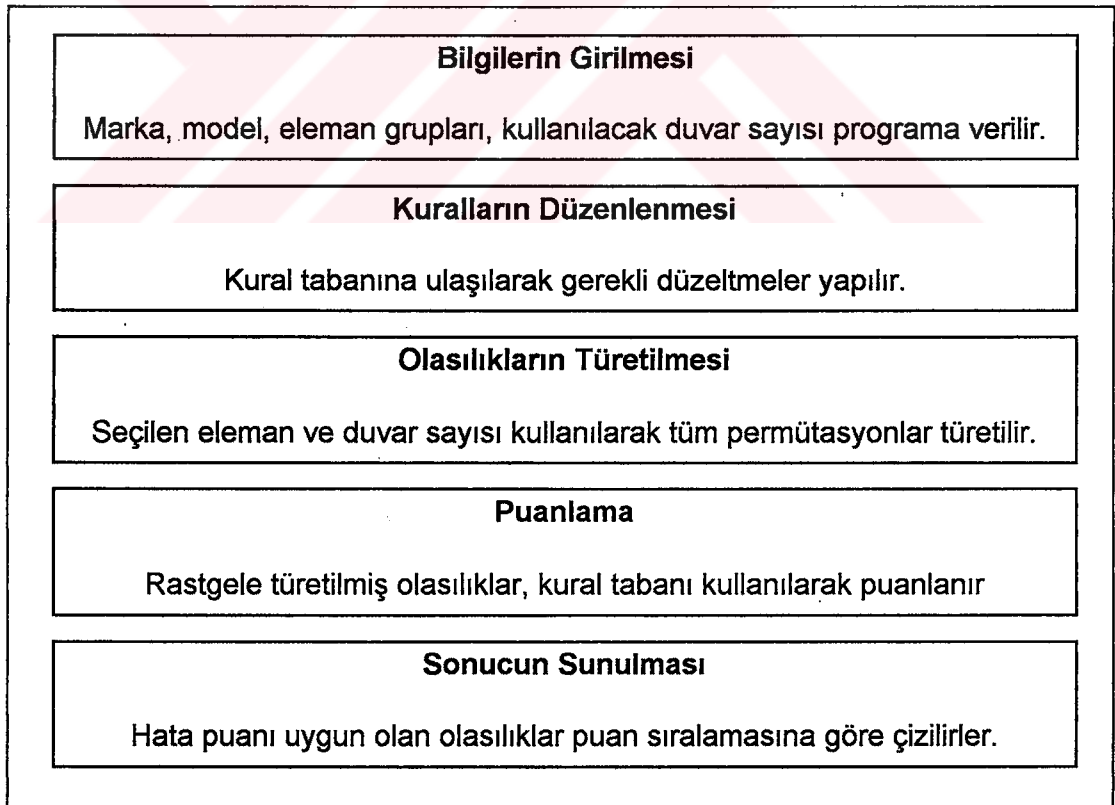
$$5 \text{ elemanlı, iki tesisat duvarlı } 480 \times 5 = 2.400$$

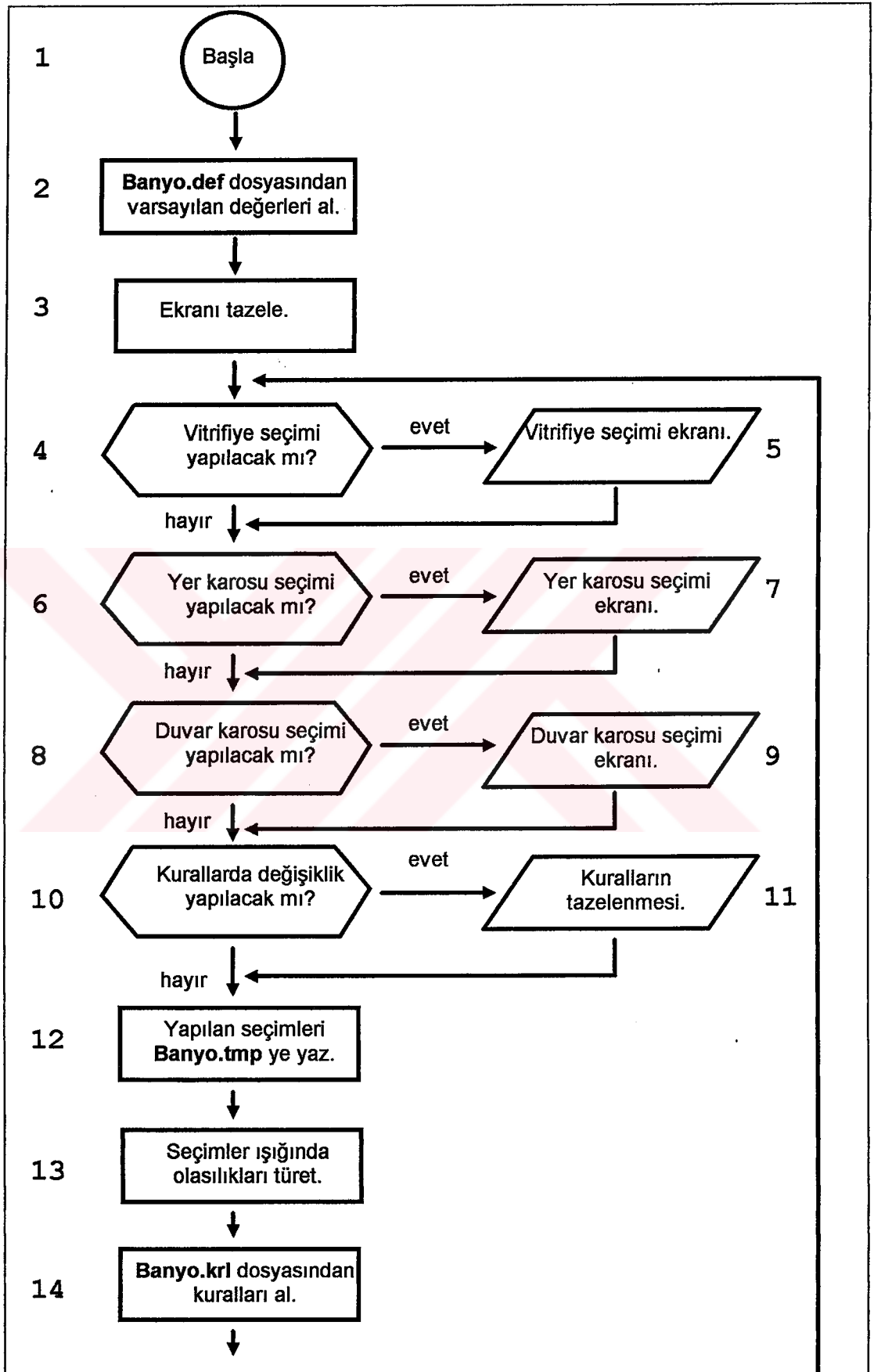
4 elemanlı, iki tesisat duvarlı 384x4 =	1.536
3 elemanlı, iki tesisat duvarlı 150x3 =	450
2 elemanlı, iki tesisat duvarlı 36x2 =	72
yanlızca tek tesisat duvarı kullanan	<u>1.032</u>
Toplam:	5.508

toplam olarak 5.508 olasılığa ulaşılır. Tüm bu olasılıklar program tarafından üretilir. Bunların yalnızca çok küçük bir kısmı uygulanabilir duruma sahiptir. Programın kullanıcı yardımına ihtiyaç duymadan bu olasılık kalabalığı arasından sağlıklı alternatifleri seçebilmesi kural tabanı sayesinde olur. Kural tabanı yardımıyla bu değerlendirme işlemi aşağıda anlatıldığı şekilde gerçekleşir.

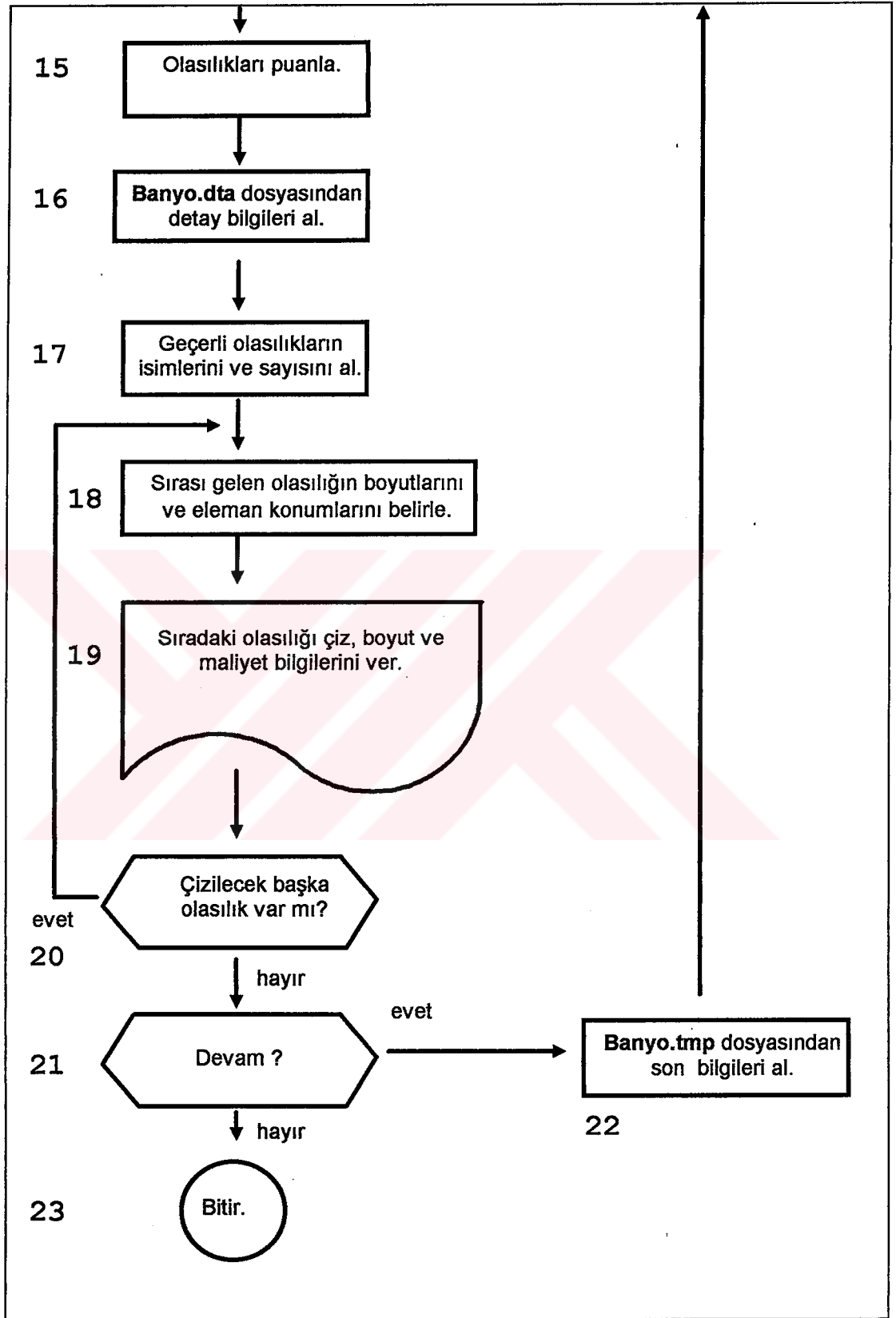
Aşağıda anlatılacak süreç Tablo 2.1 Programın Akış Şemasında ve Şekil 2.1 Programın Akış Diyagramında özetlenmiştir.

Tablo 2.1 Programın akış şeması.





Şekil 2.1a Programın akış diyagramı.



Şekil 2.1b Programın akış diyagramı.

1 :

Program çalıştırıldığında ilk olarak banyo.dwg dosyasının yüklü olup olmadığına bakar. Bu dosya çizim için gerekli olan blokları içerdiğinden, sistemin çalışması için gereklidir. Dosya yüklü değil ise, program kullanıcıyı uyarıp çalışmayı durdurur. Eğer dosya yüklü ise program dosyanın boş olup olmadığına bakar. Dosyada yapılmış çizimler bulunuyorsa kullanıcıyı uyarır. Kullanıcı, çizimleri kaydetmek için programı terk eder, ya da silinmesini isteyip programa devam eder. Program devam etmeden önce son olarak banyo.dta, banyo.def dosyalarının varlığını kontrol eder. Bu dosyaları bulamazsa kullanıcıyı uyarıp çalışmayı durdurur. Bu dosyaları bulursa ikinci adım çalışmaya başlar.

2,3 :

Banyo.def dosyasından varsayılan (default) değerler, banyo.dta dosyasında da marka ve model bilgileri alınarak ilk ekran yüklenir, ve kullanıcıdan komut beklenir.

4,5,6,7,8,9 :

Kullanıcı varsayılan vitrifiye, yer karosu, ya da duvar karosu seçimlerini değiştirmek isterse, ilgili ikonlara basarak, vitrifiye için 4 ve 5, yer karosu için 6 ve 7, duvar karosu için 8 ve 9 numaralı adımlardan geçerek tercihlerini programa bildirir.

10,11 :

Kural tabanı ile ilgili bir düzeltme yapılmak istenirse Kural tuşuna basılarak ilgili ekran yüklenir. Bu ekran yüklenmeden önce program banyo.krl dosyasının varlığını kontrol eder. Bulamazsa çalışmayı durdurur, bulursa derhal bir kopyasını yaratır. Kullanıcı farkında olmadan tüm değişiklikleri bu kopya üzerinde yapar. Bu ekran iptal tuşuyla terk edilirse, kural veritabanı üzerinde hiçbir değişiklik yapılmaz. Tamam tuşuna basılmışsa yapılan değişiklikler kural tabanına aktarılır.

12 :

Kullanıcı ana ekranı Ciz tuşuna basarak terk ederse, put_last fonksiyonu çağrılarak yapılmış olan seçimler banyo.tmp dosyasına kayıt edilir.

13:

Kullanıcıdan, tasarımda kullanılacak elemanların listesini alan program `do_perm` fonksiyonunu çalıştırır. Burada önce `test` fonksiyonuyla, kullanıcıdan gelen eleman listesi programda kullanılabilecek hale getirilir. Sonra tasarıma dahil edilecek eleman sayısına bağlı olarak `per_2`, `per_3`, `per_4`, ya da `per_5` fonksiyonları çağırılır. Bu fonksiyonlar seçilen eleman kombinasyonuna bağlı olarak permütasyonlar üretir, ve sonucu `PERM` isimli global değişkene atarlar. Program kuralları uygulamak için `exam` fonksiyonunu çağırır.

14,15:

Kullanıcının kaç tesisat duvarı kullanan tasarımlar geliştirmek istediğini tespit eden program, buna bağlı olarak `exam_1d` ya da `exam_2d` fonksiyonlarını çağırır (bu noktada düzeltilememiş bir tasarım hatası bulunmaktadır. Programın, sonuçları etkilemese de, böyle bir ayırıma gitmeden çözüme ulaşması gerekmektedir).

Exam_1d: Burada, önce `solve` fonksiyonu çağırılarak tasarımın kaç elemandan oluştuğu belirlenir (kural tabanında kurallar, kullanılan eleman sayısına göre sınıflandırılmışlardır). Eleman sayısı ile ilgili olan kurallar `get_rule` fonksiyonu çağırılarak kural tabanından alınır, ve `RULEp` değişkenine atanırlar. Bu fonksiyon, kural tabanını tarar ve kombinasyonun nitelikleriyle ilişkili olan satırları bulur. Bu satırlarda yer alan ifadeleri, `if ... then ...` komut cümleciklerine dönüştürülür.

Kural tabanında,

```
+ 034 1D 4E A2=KU AD=2 => 8
```

şeklinde yer alan bir ifade,

```
(if (and (= A2 "KU") (= AD "2")) (setq A 8))
```

tarzındaki bir komut satırına dönüştürülür. Bu cümlecikler arka arkaya dizilerek, bir tek "string" şeklinde `RULEp` değişkeni içinde saklanır.

Bu noktada **PERM** listesi içinde bulunan her permütasyon için tekrarlanan bir döngü başlar. Bu döngü içinde, eleman sayısı ile ilişkili olan kurallar, sıradaki permütasyon, ve permütasyonun eleman sayısı ile ilgili bilgiler **exam1** fonksiyonuna gönderilir. Bu fonksiyonda, **RULEp** değişkeni içinde bulunan sözcükler komut satırı haline getirilir ve işletilirler. Doğrulan her komut satırı ilişkili bulunduğu hata puanını **A** isimli değişkene atar. Listedeki bütün komutlar işlenip, ilişkili puanlarını **A** ya atadıktan sonra elde edilen değer, o olasılığın toplam hata puanını verir. **exam1** fonksiyonu hata puanlarını **exam_1d** fonksiyonuna gönderir. Burada olasılıklar ve hata puanları birleştirilerek tek bir liste oluşturulur, ve bu liste puan sırasına dizilerek sonuçlar **FINAL** değişkenine atandıktan sonra fonksiyon terk edilir.

Exam_2d: Bu fonksiyona gönderilen olasılıklar önce parçalanırlar. Her olasılık parçalanmak için, olasılığı temsil eden ifadenin ("**LA KL*DT SF CM**" = 15) boyutuna bağlı olarak; **ex5**, **ex6**, **ex8**, **ex9**, **ex11**, **ex12**, **ex14**, **ex15** şeklinde isimlendirilen fonksiyonlara gönderilir.

LA KL*DT SF CM şeklinde gelen bir olasılık.

LA KL

KL*DT

DT SF CM şeklindeki üç parçaya bölünür.

Birinci ve üçüncü parçalar, tek duvarlı bir olasılıkmış gibi, **exam_1d** de anlatıldığı şekilde sınanırlar. İkinci parça için kural tabanının ikinci bölümünde yer alan kurallar uygulanır. Bu kurallar, iki tesisat duvarının birleştiği köşede yer alan iki elemanın birbirleriyle olan ilişkilerini düzenlerler. Burada her olasılığın hata puanı, her parçanın almış olduğu hata puanları toplanarak elde edilir. Bu fonksiyonun sonunda da, yukarıda olduğu gibi, olasılıklar ve hata puanları birleştirilerek tek bir liste oluşturulur, ve bu liste, puan sırasına dizilerek sonuçlar **FINAL** değişkenine atandıktan sonra fonksiyon terk edilir.

Eğer en az bir olasılık izin verilen hata puanının altında puan alabilmişse, program 16. Adım ile devam eder.

16:

Banyo.dta dosyasının varlığı tekrar kontrol edilir, varsa kullanıcının seçmiş olduğu vitrifiye, yer karosu, ve duvar karosu için fiat, ölçü, kullanım alanı, ve blok ismi gibi bilgiler alınır. Bu bilgiler **DETAIL** isimli listede saklanırlar.

17:

Bu aşamada **cut_final** isimli fonksiyon **FINAL** listesinden hata puanları toleranstan yüksek olan olasılıkları çıkarır.

18:

draw_it adlı fonksiyon **FINAL** listesinde her eleman için uygulanacak bir döngüyü kurar. Her olasılık çizilmek üzere tek tek **main** fonksiyonuna gönderilir.

19,20:

Olasılıklar, kullandıkları duvar sayısına göre **boyutla_1d** ya da **boyutla_2d** fonksiyonlarına gönderilirler. Burada elemanlar banyo içine yerleştirilirler. Yerleşim sonunda banyo boyutları belirlenir. Daha sonra sırasıyla;

boyut_uyarla:	seçilen döşeme kaplaması boyutlarına göre banyo boyutları düzeltilir,
duvarlar:	duvarlar çizilir,
kapi:	kapi çizilir,
pencere:	pencere çizilir,
doseme:	döşeme kaplaması çizilir,
gercekler:	eleman blokları yerleştirilir,
golgeler:	elemanların altında kalan döşeme maskelenir,
olculer_yazilar:	maliyet hesaplandıktan sonra, yazılar ve ölçüler yazılır.

21 :

Olasılıkların çizimi tamamlandıktan sonra program kullanıcıya, ana ekrana dönmek isteyip istemediğini sorar. Olumlu cevap alırsa yapılan çizimleri silerek 22. Adıma geçer.

22 :

Bu adımda, daha önce ana ekranda yapılmış olan seçimler banyo.tmp dosyasından alınarak ekran tazelenir.

23 :

Kullanıcının yapılan çizimleri daha ayrıntılı inceleyebilmesi için AutoCAD komut satırına dönülür.



2.3 Kural Tabanı ve Değerlendirme

Program düzenlenebilir bir kural tabanıyla çalışmaktadır. Kullanıcı almak istediği sonuç doğrultusunda gerekli gördüğü yeni kuralları ekleyebilir, ya da gereksiz bulduklarını çıkarabilir.

Değerlendirmeyi yapan program modülüne bir önceki modülden, kullanıcının belirlediği kombinezon temel alınarak üretilmiş bir permütasyon listesi gelir. Değerlendirme modülü, listede bulunan bütün olasılıkları, kuralları kullanarak puanlar. Hata derecesini gösteren bu puanlar 0 ile 8 arasında değişir. Hatasız bir olasılık 0 puana sahiptir. Uygulanamaz durumda olanlar ise 8 ya da daha yüksek bir puana sahiptirler. Her olasılık bir hata puanıyla eşlendikten sonra, olasılıklar küçükten büyüğe doğru puan sırasına dizilirler. Son durumu içeren bir matris programın bir sonraki modülüne iletilir.

Program değerlendirme yapmak için ihtiyaç duyduğu kuralları Banyo.krl isimli dosyadan alır. Bu dosyada, programın rastgele üretilen olasılıkları sınaması için gerekli olan kural cümlecikleri bulunur. Yapısal olarak bir veri tabanı yapısı gösterir, fakat gerçek bir veri tabanı dosyası kullanılmamıştır.

Kural dosyası üç bölümden oluşur. İlk iki bölümde şart cümleleri, kurgulanmış bir yapı içinde temsil edilir, ve sınırlı hareket imkanı verirler. Kuralların satır uzunlukları ve terimlerin yeri mutlaka aynı olmalıdır. Aksi takdirde program yanlış bir sonuç çıkarır, ya da bir hata mesajı verip çalışmaya son verir. Son bölüm ise ileri kullanıcılar için hazırlanmıştır. Bu bölümde koşul cümleleri LISP dilinin yapısına uygun bir komut satırı olarak girilir.

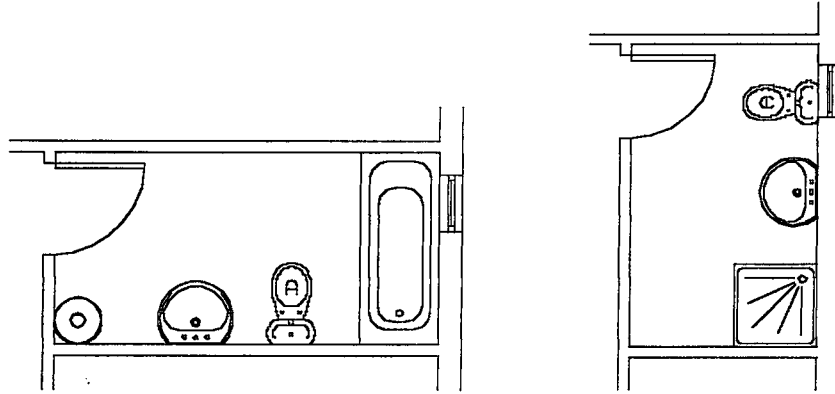
Tüm kurallar puanlama esasına göre düzenlenmiştir. Kuallar genel olarak

<mevcut durum> => <hata puanı>

şeklinde bir söz dizimine sahiptir.

2.3.1 Birinci Kural Bölümü

Birinci bölüm yalnız tek duvarlı olasılıklarla ilgilenir. Bu duvar, birinci ya da ikinci ıslak duvar olabilir (Şekil 2.1).



Şekil 2.1 Tek duvarlı olasılıklar: a). TS LA KL KU*, ve b). *DT LA KL.

Söz dizimi

<A konumunda B elemanı varsa> ve
<Aktif duvar C ise> => <hata puanı>

şeklindedir. Bu bölümün kural yapısı şu şekildedir:

+ 002 1D 2E A1=KU AD=1 => 8

Bu komut şu anlama gelir:

Olasılıkta, tüm elemanlar tek duvarda yer alıyorsa (1D), ve olasılıklar 2 elemanlıysa (2E), ve birinci sırada küvet varsa (A1=KU), ve söz konusu olan duvar 1. duvar ise (AD=1), bu olasılığa 8 puan ver, yani kullanım dışı bırak.

Bu kuralın komut satırındaki alanlar şunlardır:

- +** : Kuralın aktif olduğunu bildirir. Eğer işaret **-** ise, koşul veri tabanında olduğu halde değerlendirmeyi etkilemez.
- 001** : Kural numarasını verir. Sonucu etkilemez.
- 1D** : Kuralın yalnız bir duvarla ilgili olduğunu gösterir. Anlatılan yapı içinde kural verilecekse bu terimin kullanılması şarttır.
- 2E** : İki elemanlı olasılıklarla ilgili olduğunu gösterir.
2E, 3E, 4E, ve 5E şeklinde kullanılabilir.
- A1** : Kuralın hangi sıradaki elemanla ilgili olduğunu gösterir.
A1, A2, A3, A4, A5 şeklinde kullanılabilir.
- =** : Bu işaretin seçeneği yoktur. Tam o noktada kullanılmak zorundadır.
- KU** : Kuralın ilgili olduğu elemanı gösterir.
LA, KL, CM, DT, KU, SF, TS şeklinde kullanılabilir.
- AD** : Aktif duvar anlamına gelir. Değiştirilmemesi gerekir.
- 1** : Söz konusu hangi duvarın aktif olduğunu belirler.
1. ya da 2. duvar aktif olabilir.
- 8** : Kural sağlanmışsa olasılığa atanacak hata puanıdır.
1 den 8 e kadar kullanılabilir. 8 verilirse söz konusu durumu içeren olasılıklar kullanım dışı kalırlar. Pratik olarak bu değer 1 ile 5 arasında olması gerekir.

Bu gruba giren ve kullanıcıya Banyo.krl dosyasında hazır olarak verilen kuralların bir kısmı aşağıda sıralanmıştır.

```
+ 001  1D 2E A1=DT AD=1 => 1
+ 002  1D 2E A1=KU AD=1 => 8
+ 003  1D 2E A2=SF AD=1 => 1
+ 004  1D 2E A2=TS AD=1 => 1

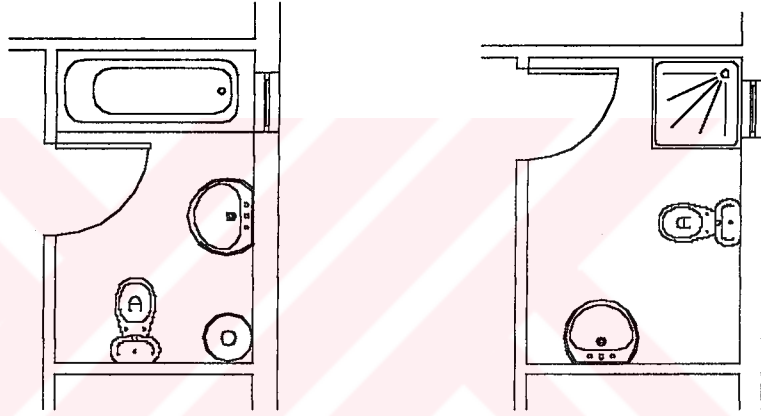
+ 005  1D 2E A2=SF AD=2 => 1
```

2.3.2 İkinci Kural Bölümü

İki tesisat duvarının birleştiği köşede, köşenin iki yanında yer alan elemanların birbirleriyle olan durumlarını değerlendiren kurallar bu bölümde bulunur (Şekil 2.2). Söz dizimi

<köşenin ilk duvar tarafında A elemanı varsa> ve
<köşenin ikinci duvar tarafında B elemanı varsa> => <hata puanı>

şeklindedir.



Şekil 2.2 İki duvarlı olasılıklar: a). KL*TS LA KU, ve b): LA*KL DT.

Bu bölümün kural yapısı şu şekildedir:

+ 079 2D Ax=DT ve Ay=CM => 4

Bu komut şu anlama gelir:

Olasılıkta elemanlar iki duvara yayılmışlarsa (2D), ve birinci duvarın son elemanı duş teknesi ise (Ax=DT), ve ikinci duvarın ilk elemanı çamaşır makinası ise (Ay=CM), bu olasılığa 4 hata puanı ver.

Bu komut satırındaki alanlar şunlardır:

- +** : Kuralın aktif olduğunu bildirir. Eğer işaret **-** ise, koşul veri tabanında olduğu halde değerlendirmeyi etkilemez.
- 079** : Kural numarasını gösterir. Sonucu etkilemez.
- 2D** : İki duvarı birden kullanan olasılıklarla ilgili bir kural olduğunu ifade eder.
- Ax** : Birinci duvarın son elemanını tarif eder. Değiştirilmemesi gereken bir terimdir.
- DT** : Birinci duvarın son sırasında bulunan elemanın adını içeren bir alandır. LA, KL, CM, DT, KU, TS, SF kısaltmaları kullanılabilir.
- Ay** : İkinci duvarın ilk elemanını tarif eder. Değiştirilmemesi gereken bir terimdir.
- CM** : İkinci duvarın ilk sırasında bulunan elemanın adını içeren bir alandır. LA, KL, CM, DT, KU, TS, SF kısaltmaları kullanılabilir.
- 1** : Kural sağlanmışsa olasılığa atanacak hata puanıdır. 1 den 8 e kadar kullanılabilir. 8 verilirse söz konusu durumu içeren olasılıklar kullanım dışı kalırlar. Pratik olarak bu değerin 1 ile 5 arasında olması gerekir.

Üçüncü bölümde ise yapısı ve uzunluğu tamamen açık bırakılmış kurallar yer alır. Kullanılan söz dizimi tümüyle LISP e aittir. Program bu satırı ya da satırları alır ve doğrudan işleme sokar. Bu satırların programın kendi kodundan hiçbir farkı yoktur. İlk anda zor gibi görünse de, kullanıcı bu dile alıştığı zaman çok etkili kurallar geliştirme şansına sahip olur. Söz dizimi aşağıdaki şekildedir:

```
(if (and (= DV 2) (> D2 1) (= Ax "KU")) (setq A (+ A 8)) )
```

bu komut şu anlama gelir:

Olasılık iki duvar kullanıyorsa ($(= DV \ 2)$), ve ikinci duvarda birden fazla eleman bulunuyorsa ($(> D2 \ 1)$), ve ilk duvarın son elemanı küvet ise ($(= Ax \ "KU")$) bu olasılığı geçersiz ilan et ($(setq \ A \ (+ \ A \ 8))$).

Burada kullanılan terimler aynı zamanda programın içinde tanımlanmış olan değişkenlerdir. Bu bölümde kullanılabilen terimler (ilk iki bölümde anlatılanlara ek olarak) şunlardır:

Ax	:	İlk duvarın son elemanı.
Ay	:	İkinci duvarın ilk elemanı.
DV	:	Kullanılan duvar sayısı.
D1	:	Birinci duvarda kullanılan eleman sayısı.
D2	:	İkinci duvarda kullanılan eleman sayısı.
A1	:	Olasılığın ilk elemanı. Burada olasılıklar iki duvarlı da olsalar elemanlar tek duvara dizilmiş gibi ele alınırlar. Örneğin 1. duvarda 1 eleman, 2. duvarda 3 eleman varsa, ve 2. duvarın 2. elemanı aranıyorsa A3 kullanılarak bir sorgu oluşturulması gerekir.
A2	:	2. eleman.
A3	:	3. eleman.
A4	:	4. eleman.
A5	:	5. eleman.
LA	:	Lavabo için kullanılan kısaltma.
KL	:	Klozet için kullanılan kısaltma.
CM	:	Çamaşır makinası için kullanılan kısaltma.
DT	:	Duş teknesi için kullanılan kısaltma.
KU	:	Küvet için kullanılan kısaltma.
SF	:	Şofben için kullanılan kısaltma.
TS	:	Termosifon için kullanılan kısaltma.

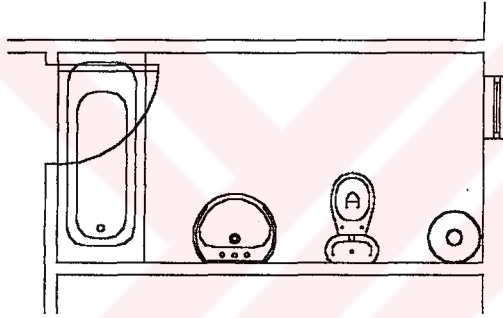
Bu bölümde kullanıcı, eğer gerek görürse, kendi değişkenlerini yaratabilir, AutoLIPS e ait bütün fonksiyonları kullanabilir, ya da program içinde tanımlanmış tüm global değişkenlere ulaşabilir.

2.3.3 Kuralların Açıklanması

Bu bölümde kuralların açıklaması yapılacaktır. Açıklamalar yapılırken kurallar, kural tabanından farklı olarak, elemanlara göre sınıflandırılacaktır

Aşağıda dökümü verilip açıklaması yapılan kuralların tümü, geliştirilmeye çalışılan açık kural tabanlı bir sistemin karallılığının test edilebilmesi için uyarlanmış kural önerileridir. Bu nedenle, ne kural tabanı eksiksizdir, ne de kurallar tartışılmazdır.

2.3.3.1 Küvetle İlgili Kurallar

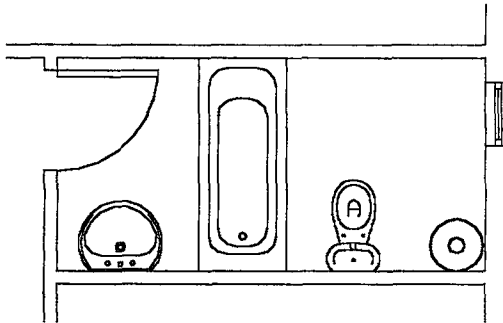


Tablo 2.2a Küvetle İlgili Kurallar.

+ 002	1D	2E	A1=KU	AD=1	=>	8
+ 008	1D	3E	A1=KU	AD=1	=>	8
+ 021	1D	4E	A1=KU	AD=1	=>	8
+ 043	1D	5E	A1=KU	AD=1	=>	8

Şekil 2.2a Küvetle İlgili Kurallar.

Tablo 2.2a da verilen kuralların tümü, eleman sayısına bağlı kalmaksızın, şekil 2.2a da görüldüğü biçimde, küvetin kapının önüne gelmesini engellemeye yöneliktir.

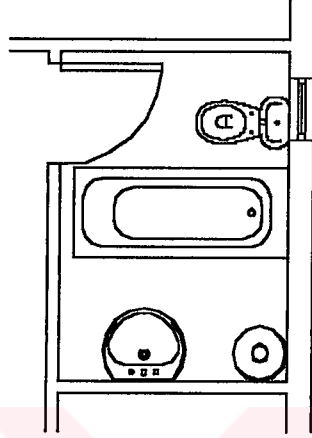


Tablo 2.2b Küvetle İlgili Kurallar.

+ 011	1D	3E	A2=KU	AD=1	=>	8
+ 024	1D	4E	A2=KU	AD=1	=>	8
+ 046	1D	5E	A2=KU	AD=1	=>	8
+ 029	1D	4E	A3=KU	AD=1	=>	8
+ 051	1D	5E	A3=KU	AD=1	=>	8
+ 056	1D	5E	A4=KU	AD=1	=>	8

Şekil 2.2b Küvetle İlgili Kurallar.

Tablo 2.2b de verilen kuralların tümü, eleman sayısına bağlı kalmaksızın, şekil 2.2b de görüldüğü biçimde, küvetin birinci tesisat duvarının ortasında kalmasını engellemeye yöneliktir.



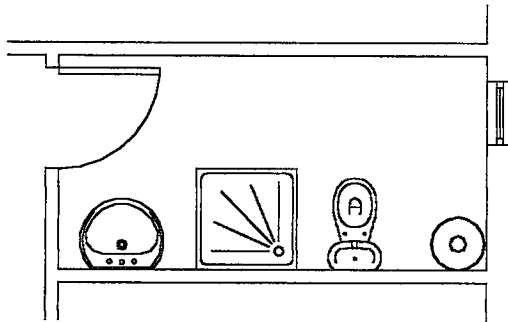
Şekil 2.2c Küvetle İlgili Kurallar.

Tablo 2.2c Küvetle İlgili Kurallar.

+ 017	1D	3E	A2=KU	AD=2	=>	8
+ 034	1D	4E	A2=KU	AD=2	=>	8
+ 061	1D	5E	A2=KU	AD=2	=>	8
+ 039	1D	4E	A3=KU	AD=2	=>	8
+ 066	1D	5E	A3=KU	AD=2	=>	8
+ 071	1D	5E	A4=KU	AD=2	=>	8

Tablo 2.2c de verilen kuralların tümü, eleman sayısına bağlı kalmaksızın, şekil 2.2c de görüldüğü biçimde, küvetin ikinci tesisat duvarının ortasında kalmasını engellemeye yöneliktir.

2.3.3.2 Duş Teknesiyle İlgili Kurallar

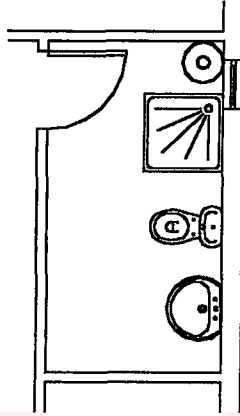


Tablo 2.3a Duş Teknesiyle İlgili Kurallar

+ 010	1D	3E	A2=DT	AD=1	=>	8
+ 023	1D	4E	A2=DT	AD=1	=>	8
+ 028	1D	4E	A3=DT	AD=1	=>	8
+ 045	1D	5E	A2=DT	AD=1	=>	8
+ 050	1D	5E	A3=DT	AD=1	=>	8
+ 055	1D	5E	A4=DT	AD=1	=>	8

Şekil 2.3a Duş Teknesiyle İlgili Kurallar.

Tablo 2.3a da verilen kuralların tümü, eleman sayısına bağlı kalmaksızın, duş teknesinin, şekil 2.3a da görüldüğü biçimde, birinci tesisat duvarının ortasında kalması durumunu engellemeye yöneliktir.

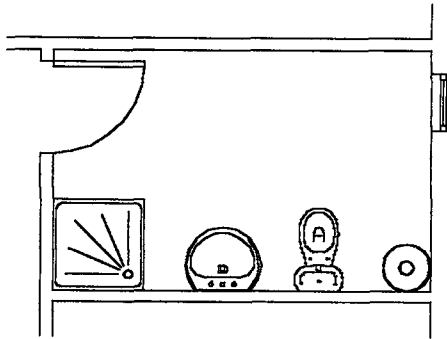


Tablo 2.3b Duş Teknesiyle İlgili Kurallar

+ 016	1D	3E	A2=DT	AD=2	=> 8
+ 038	1D	4E	A3=DT	AD=2	=> 8
+ 033	1D	4E	A2=DT	AD=2	=> 8
+ 065	1D	5E	A3=DT	AD=2	=> 8
+ 060	1D	5E	A2=DT	AD=2	=> 8
+ 070	1D	5E	A4=DT	AD=2	=> 8

Şekil 2.3b Duş Teknesiyle İlgili Kurallar.

Tablo 2.3b de verilen kuralların tümü, eleman sayısına bağlı kalmaksızın, duş teknesinin, şekil 2.3b de görüldüğü biçimde, ikinci tesisat duvarının ortasında kalması durumunu engellemeye yöneliktir.



Tablo 2.3c Duş Teknesiyle İlgili Kurallar

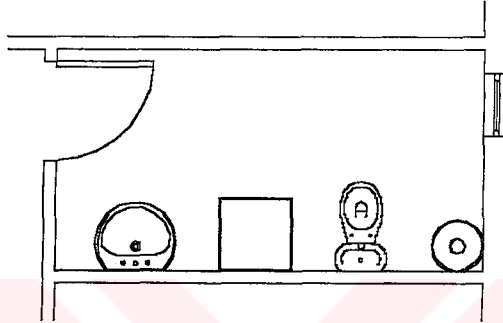
+ 001	1D	2E	A1=DT	AD=1	=> 1
+ 007	1D	3E	A1=DT	AD=1	=> 1
+ 020	1D	4E	A1=DT	AD=1	=> 1
+ 042	1D	5E	A1=DT	AD=1	=> 1

Şekil 2.3c Duş Teknesiyle İlgili Kurallar.

Tablo 2.3c de verilen kuralların tümü, eleman sayısına bağlı kalmaksızın, duş teknesinin, şekil 2.3c de görüldüğü biçimde, kapının önüne durumunu olumsuzlamaya yöneliktir ($\Rightarrow 1$).

2.3.3.3 Çamaşır Makinesiyle İlgili Kurallar

Tablo 2.4a Çamaşır Makinesiyle İlgili Kurallar.

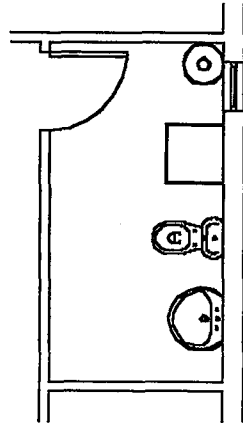


+ 009	1D	3E	A2=CM	AD=1	$\Rightarrow 3$
+ 022	1D	4E	A2=CM	AD=1	$\Rightarrow 3$
+ 027	1D	4E	A3=CM	AD=1	$\Rightarrow 3$
+ 044	1D	5E	A2=CM	AD=1	$\Rightarrow 3$
+ 049	1D	5E	A3=CM	AD=1	$\Rightarrow 3$
+ 054	1D	5E	A4=CM	AD=1	$\Rightarrow 3$

Şekil 2.4a Çamaşır Makinesiyle İlgili Kurallar.

Tablo 2.4a da verilen kuralların tümü, eleman sayısına bağlı kalmaksızın, çamaşır makinesinin, şekil 2.4a da görüldüğü biçimde, birinci tesisat duvarının ortasında kalması durumunu olumsuzlamaya ($\Rightarrow 3$) yöneliktir.

Tablo 2.4b Çamaşır Makinesiyle İlgili Kurallar.

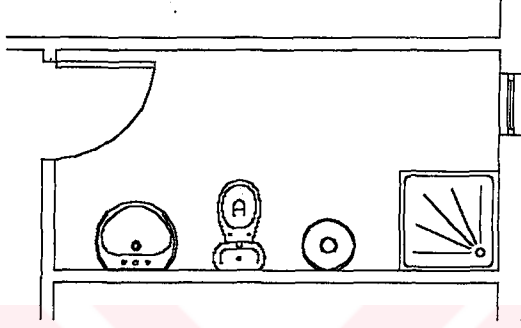


+ 015	1D	3E	A2=CM	AD=2	$\Rightarrow 3$
+ 032	1D	4E	A2=CM	AD=2	$\Rightarrow 3$
+ 037	1D	4E	A3=CM	AD=2	$\Rightarrow 3$
+ 059	1D	5E	A2=CM	AD=2	$\Rightarrow 3$
+ 064	1D	5E	A3=CM	AD=2	$\Rightarrow 3$
+ 069	1D	5E	A4=CM	AD=2	$\Rightarrow 3$

Şekil 2.4b Çamaşır Makinesiyle İlgili Kurallar.

Tablo 2.4b de verilen kuralların tümü, eleman sayısına bağlı kalmaksızın, çamaşır makinesinin, şekil 2.4b de görüldüğü biçimde , ikinci tesisat duvarının ortasında kalması durumunu olumsuzlamaya ($\Rightarrow 3$) yöneliktir.

2.3.3.4 Termosifonla İlgili Kurallar



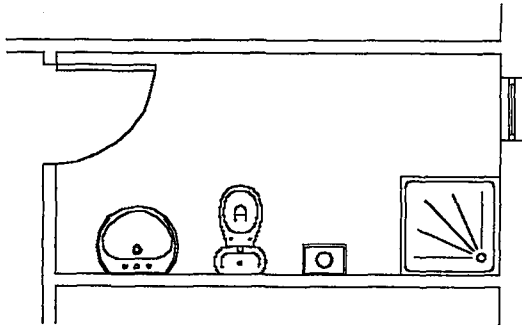
Şekil 2.5 Termosifonla İlgili Kurallar.

Tablo 2.5 Termosifonla İlgili Kurallar.

+ 013	1D	3E	A2=TS	AD=1	$\Rightarrow 1$
+ 026	1D	4E	A2=TS	AD=1	$\Rightarrow 1$
+ 031	1D	4E	A3=TS	AD=1	$\Rightarrow 1$
+ 048	1D	5E	A2=TS	AD=1	$\Rightarrow 1$
+ 053	1D	5E	A3=TS	AD=1	$\Rightarrow 1$
+ 058	1D	5E	A4=TS	AD=1	$\Rightarrow 1$
+ 018	1D	3E	A2=TS	AD=2	$\Rightarrow 1$
+ 036	1D	4E	A2=TS	AD=2	$\Rightarrow 1$
+ 041	1D	4E	A3=TS	AD=2	$\Rightarrow 1$
+ 063	1D	5E	A2=TS	AD=2	$\Rightarrow 1$
+ 068	1D	5E	A3=TS	AD=2	$\Rightarrow 1$
+ 073	1D	5E	A4=TS	AD=2	$\Rightarrow 1$

Tablo 2.5 de verilen kuralların tümü termosifonun, eleman sayısına ve aktif duvara bağlı kalmaksızın, şekil 2.5 de görüldüğü biçimde, arada kalması durumunu olumsuzlamaya ($\Rightarrow 1$) yöneliktir.

2.3.3.5 Şofbenle İlgili Kurallar



Şekil 2.6 Şofbenle İlgili Kurallar.

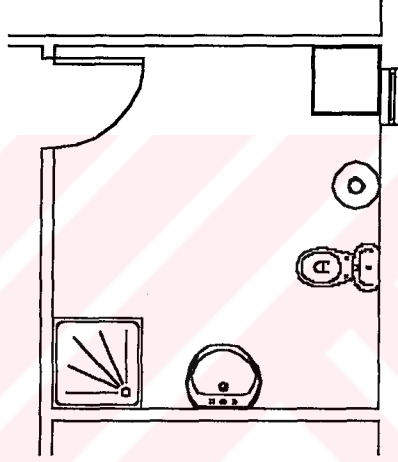
Tablo 2.6 Şofbenle İlgili Kurallar.

+ 012	1D	3E	A2=SF	AD=1	$\Rightarrow 2$
+ 025	1D	4E	A2=SF	AD=1	$\Rightarrow 2$
+ 030	1D	4E	A3=SF	AD=1	$\Rightarrow 2$
+ 047	1D	5E	A2=SF	AD=1	$\Rightarrow 2$
+ 052	1D	5E	A3=SF	AD=1	$\Rightarrow 2$
+ 057	1D	5E	A4=SF	AD=1	$\Rightarrow 2$
+ 035	1D	4E	A2=SF	AD=2	$\Rightarrow 2$
+ 040	1D	4E	A3=SF	AD=2	$\Rightarrow 2$
+ 062	1D	5E	A2=SF	AD=2	$\Rightarrow 2$
+ 067	1D	5E	A3=SF	AD=2	$\Rightarrow 2$
+ 072	1D	5E	A4=SF	AD=2	$\Rightarrow 2$

Tablo 2.6 da verilen kuralların tümü şofbenin, eleman sayısına ve aktif duvara bağlı kalmaksızın, şekil 2.6 da görüldüğü biçimde, arada kalması durumunu olumsuzlamaya ($\Rightarrow 2$) yöneliktir.

2.3.4 Kural Uygulanması için Bir Örnek

Sistemin değerlendirme yöntemini açıklamak amacıyla, son olarak, örnek bir alternatifin değerlendirilmesi açıklanacaktır.



Şekil 2.7 Kural Uygulanması için Bir Örnek : DT LA*KL TS CM.

Program, Şekil 2.7 de görülen, DT LA*KL TS CM şeklindeki bir alternatifi değerlendirmek için ilk olarak kural tabanının 3. Bölümünü yükler. Bu bölümde yer alan kurallar, hiçbir değişkene bağlı kalmadan, değerlendirmeye katılacak kurallar arasına alınırlar. Seçilen olasılıklardaki eleman sayısı, eleman türleri, kullanılan duvar sayısı ne olursa olsun bu kurallar yüklenirler. Bu durumda kural listesinin ilk hali aşağıdaki gibi olur;

```
(if (and (= DV 2) (> D2 0) (= Ax "KU"))) (setq A (+ A 8)))
(if (and (= DV 2) (> D1 0) (= Ay "KU"))) (setq A (+ A 8)))
```

Her olasılık en azından 1 duvar kullanmak zorunda olduğu için, program bir sonraki adımda tek duvarla ilgili kurallara yönelir. Seçilen olasılık 5 eleman içerdiği için

başında 5E yazılı olan tüm kurallar yüklenir ve kural listesinin sonuna eklenir. Kural listesinin yeni hali aşağıdaki gibidir;

```
(if (and (= DV 2) (> D2 0) (= Ax "KU")) (setq A (+ A 8)))
(if (and (= DV 2) (> D1 0) (= Ay "KU")) (setq A (+ A 8)))
```

```
+ 042 1D 5E A1=DT AD=1 => 1
+ 043 1D 5E A1=KU AD=1 => 8
+ 044 1D 5E A2=CM AD=1 => 3
+ 045 1D 5E A2=DT AD=1 => 8
+ 046 1D 5E A2=KU AD=1 => 8
+ 047 1D 5E A2=SF AD=1 => 2
+ 048 1D 5E A2=TS AD=1 => 1
+ 049 1D 5E A3=CM AD=1 => 3
+ 050 1D 5E A3=DT AD=1 => 8
+ 051 1D 5E A3=KU AD=1 => 8
+ 052 1D 5E A3=SF AD=1 => 2
+ 053 1D 5E A3=TS AD=1 => 1
+ 054 1D 5E A4=CM AD=1 => 3
+ 055 1D 5E A4=DT AD=1 => 8
+ 056 1D 5E A4=KU AD=1 => 8
+ 057 1D 5E A4=SF AD=1 => 2
+ 058 1D 5E A4=TS AD=1 => 1

+ 059 1D 5E A2=CM AD=2 => 3
+ 060 1D 5E A2=DT AD=2 => 8
+ 061 1D 5E A2=KU AD=2 => 8
+ 062 1D 5E A2=SF AD=2 => 2
+ 063 1D 5E A2=TS AD=2 => 1
+ 064 1D 5E A3=CM AD=2 => 3
+ 065 1D 5E A3=DT AD=2 => 8
+ 066 1D 5E A3=KU AD=2 => 8
+ 067 1D 5E A3=SF AD=2 => 2
+ 068 1D 5E A3=TS AD=2 => 1
+ 069 1D 5E A4=CM AD=2 => 3
+ 070 1D 5E A4=DT AD=2 => 8
+ 071 1D 5E A4=KU AD=2 => 8
+ 072 1D 5E A4=SF AD=2 => 2
+ 073 1D 5E A4=TS AD=2 => 1
```

Son olarak program, olasılığın kaç duvar kullandığına bakar. Eğer seçilen olasılık 2 duvar kullanıyorsa, ilk duvarki son elemanın adı alınır ve bu elemanla ilgili olup da 2. Kural bölümünde bulunan bütün kurallar kural listesine eklenir. Kural listesinin son hali aşağıdaki gibidir;

```
(if (and (= DV 2) (> D2 0) (= Ax "KU")) (setq A (+ A 8)))
(if (and (= DV 2) (> D1 0) (= Ay "KU")) (setq A (+ A 8)))
```

```
+ 042 1D 5E A1=DT AD=1 => 1
+ 043 1D 5E A1=KU AD=1 => 8
+ 044 1D 5E A2=CM AD=1 => 3
+ 045 1D 5E A2=DT AD=1 => 8
+ 046 1D 5E A2=KU AD=1 => 8
+ 047 1D 5E A2=SF AD=1 => 2
+ 048 1D 5E A2=TS AD=1 => 1
+ 049 1D 5E A3=CM AD=1 => 3
+ 050 1D 5E A3=DT AD=1 => 8
+ 051 1D 5E A3=KU AD=1 => 8
+ 052 1D 5E A3=SF AD=1 => 2
+ 053 1D 5E A3=TS AD=1 => 1
+ 054 1D 5E A4=CM AD=1 => 3
+ 055 1D 5E A4=DT AD=1 => 8
+ 056 1D 5E A4=KU AD=1 => 8
+ 057 1D 5E A4=SF AD=1 => 2
+ 058 1D 5E A4=TS AD=1 => 1
```

```
+ 059 1D 5E A2=CM AD=2 => 3
+ 060 1D 5E A2=DT AD=2 => 8
+ 061 1D 5E A2=KU AD=2 => 8
+ 062 1D 5E A2=SF AD=2 => 2
+ 063 1D 5E A2=TS AD=2 => 1
+ 064 1D 5E A3=CM AD=2 => 3
+ 065 1D 5E A3=DT AD=2 => 8
+ 066 1D 5E A3=KU AD=2 => 8
+ 067 1D 5E A3=SF AD=2 => 2
+ 068 1D 5E A3=TS AD=2 => 1
+ 069 1D 5E A4=CM AD=2 => 3
+ 070 1D 5E A4=DT AD=2 => 8
+ 071 1D 5E A4=KU AD=2 => 8
+ 072 1D 5E A4=SF AD=2 => 2
+ 073 1D 5E A4=TS AD=2 => 1
```

```
+ 089 2D Ax=LA ve Ay=CM => 3
+ 090 2D Ax=LA ve Ay=DT => 3
+ 091 2D Ax=LA ve Ay=KL => 2
+ 092 2D Ax=LA ve Ay=SF => 3
+ 093 2D Ax=LA ve Ay=TS => 2
```

Kural yükleme işleminden sonra programın elinde 49 adet kuraldan oluşan bir kural listesi bulunur.

Kuralları uygulamadan önce, program kural uygulama bölümü için bir dizi değişkeni belirler. Bunlar aşağıda sıralanmıştır;

$A_x = LA$	:	İlk duvarın son elemanı Lavabo.
$A_y = KL$	:	İkinci duvarın ilk elemanı Klozet.
$A_1 = DT$	:	İlk eleman Duş teknesi.
$A_2 = LA$	:	İkinci eleman Lavabo.
$A_3 = KL$	:	Üçüncü eleman Klozet.
$A_4 = TS$	:	Dördüncü eleman Termosifon.
$A_5 = CM$	:	Beşinci eleman Çamaşır makinesi.
$D_1 = 2$	:	İlk duvarda 2 eleman var.
$D_2 = 3$	:	İkinci duvarda 3 eleman var.
$DV = 2$	:	Bu olasılık iki duvar kullanıyor.

Kural listesinin ilk iki satırı Küvet içeren olasılıklarla ilgili olduğu için bu kurallar değerlendirmeye katılmazlar. Yapılan değerlendirme sonucunda olasılığın özellikleri 3 ayrı kuralın koşullarına uyar;

+ 042 1D 5E $A_1=DT$ $AD=1 \Rightarrow 1$

Duş teknesi 1.sırada, yani kapının önünde olduğu için olasılığa 1 hata puanı verilir,

+ 073 1D 5E $A_4=TS$ $AD=2 \Rightarrow 1$

Termosifon diğer elemanların arasında kaldığı için olasılığın hata puanı 1 artırılır,

+ 091 2D $A_x=LA$ ve $A_y=KL \Rightarrow 2$:

Son olarak olasılık aynı anda iki tesisat duvarı kullandığı, ve lavabo ile klozet köşede karşılaşip alan kaybına yol açtıkları için olasılığın hata puanı iki artırılır.

Bu üç kuralın sağlanması sonucunda olasılığın hata puanı 4 e ulaşır, ve bu olasılığın 0, 1, 2, ve 3 hata puanlı olasılıkların arkasında yer alması sağlanır.

BÖLÜM 3

YAZILIM

3.1 Programın Çalıştırılması

Yazılım çalışmak için AutoCAD sürüm 12 ya da daha ileri bir sürüme ihtiyaç duyar. Yazılım geliştirilirken AutoCAD 13 programının Windows sürümü kullanılmıştır. Programlanabilir diyalog pencerelerinin AutoCAD programının çalıştığı ortama bağımlı olmamalarından dolayı bu yazılım Unix ya da DOS gibi diğer ortamlarda da sorunsuz çalışabilir.

Yazılım geliştirilirken, Intel Pentium 90 işlemci, 16 MB RAM, 1200 MB sabit disk, SVGA monitor ve 2 MB Vram ekran kartına sahip IBM PC uyumlu bir bilgisayar kullanıldı. Fakat program, bir çok AutoLISP programı gibi, çalışmak için sadece AutoCAD sürüm 12 programının sorunsuz çalıştığı bir makineye ihtiyaç duymaktadır. Bunun dışında ekran kartının en az 256 renkle çalışıyor olması gerekmektedir. Elemanların altına çizilen döşemeyi gizlemek için bir takım saklı nesneler kullanılmıştır. Bu nesnelerin görünmemesi için AutoCAD in ekran fon rengine yakın bir renkle çizilmeleri gerekmektedir. Oysa kullanıma yalnızca 256 dan daha az renk açık iken uygun rengi bulmak mümkün olmamaktadır.

Programı kullanmak için özel bir programcılık bilgisine ihtiyaç duyulmamaktadır. AutoCAD programıyla tanışık olmak programdan alınacak verimi artıracaktır. Kural tabanının 3. Seviyesini kullanmak için ise bir miktar programcılık bilgisi veya LISP sözdizimiyle aşinalık gerekmektedir. Programlanabilir diyalog pencerelerinin kullanımı sayesinde son kullanıcıyla programın iletişimi güçlendirilmiştir.

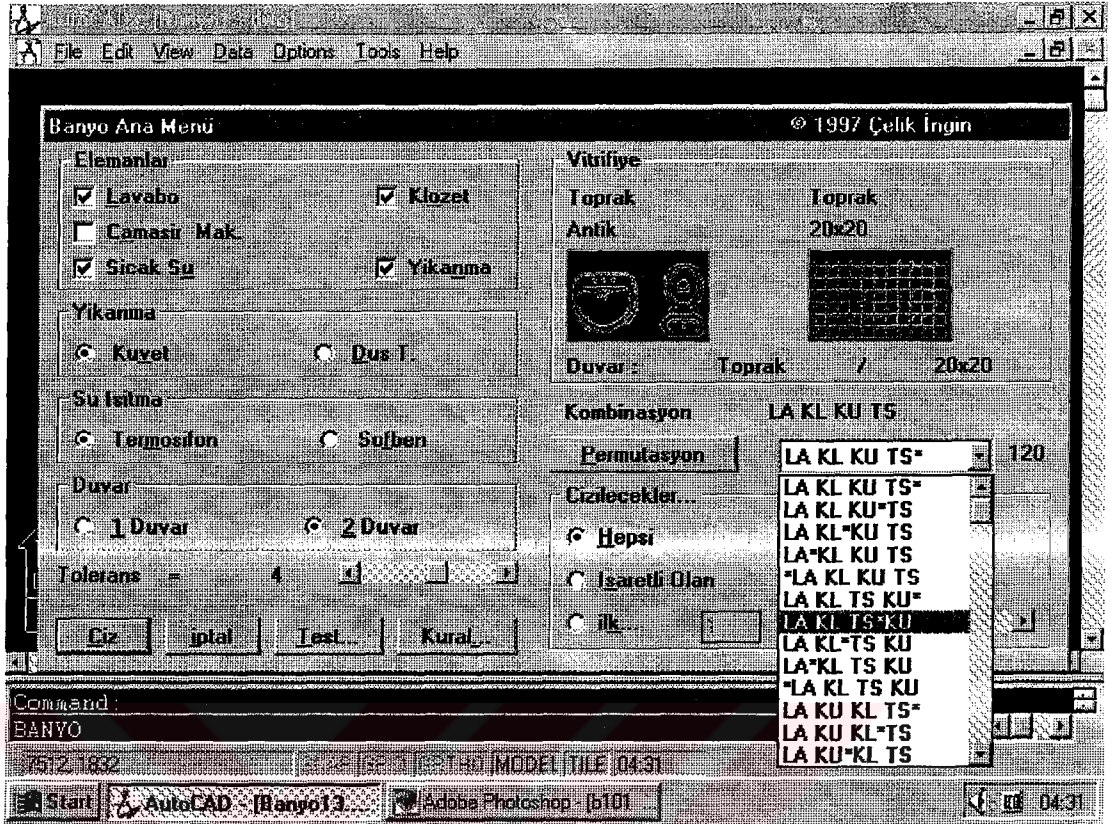
Program 9 adet dosyadan oluşmaktadır:

Banyo12.dwg :	Tasarımda kullanılacak elemanların bloklarını içeren AutoCAD sürüm 12 formatındaki dosya.
Banyo13.dwg :	Tasarımda kullanılacak elemanların bloklarını içeren AutoCAD sürüm 13 formatındaki dosya.
Banyo.exe :	AutoLISP programının Visual C++ 4.2 ile derlenmiş hali.
Banyo.dcl :	AutoLISP DCL dosyası.
Banyo.dta :	Firmalar, modeller, ve diğer elemanların bilgilerini içerir.
Banyo.krl :	Kural tabanı bilgilerini içerir.
Banyo.krx :	Kural tabanı üzerinde çalışılırken geçici olarak yaratılan ve veri tabanının güvenliğini sağlayan bir dosyadır.
Banyo.def :	Programın başlangıç değerlerini saklar.
Banyo.tmp :	Programın çalışması sırasında üretilen bilgileri saklar.

Bu dosyalardan Banyo.tmp ve Banyo.krx kullanıcıya hiç görünmez. Banyo.exe ve Banyo.dcl programın çalışması için herhangi bir müdahaleye ihtiyaç duymazlar. Banyo12.dwg ya da Banyo13.dwg dosyalarına yeni bloklar eklemek gerekebilir. Fakat ilk aşamada, program içinde hazır olarak gelen bloklar programın çalışması için yeterlidir. Banyo.def dosyasını değiştirmek için en azından programı bir kaç saat kullanmış olmak gerekir. Banyo.dta ve Banyo.krl dosyaları ise her zaman kullanıcının müdahalesine açık olarak bulunurlar.

DOS işletim sisteminde çalışılıyorsa "acad" komutu verilerek AutoCAD programı yüklenir. Windows ortamında çalışılıyorsa, AutoCAD program grubundan AutoCAD programı çalıştırılır. Banyo12.dwg ya da Banyo13.dwg dosyası açılır. Daha sonra Banyo.exe dosyası yüklenir. Bu dosya yüklendikten sonra "banyo" yazılır ve "enter" tuşuna basılır.

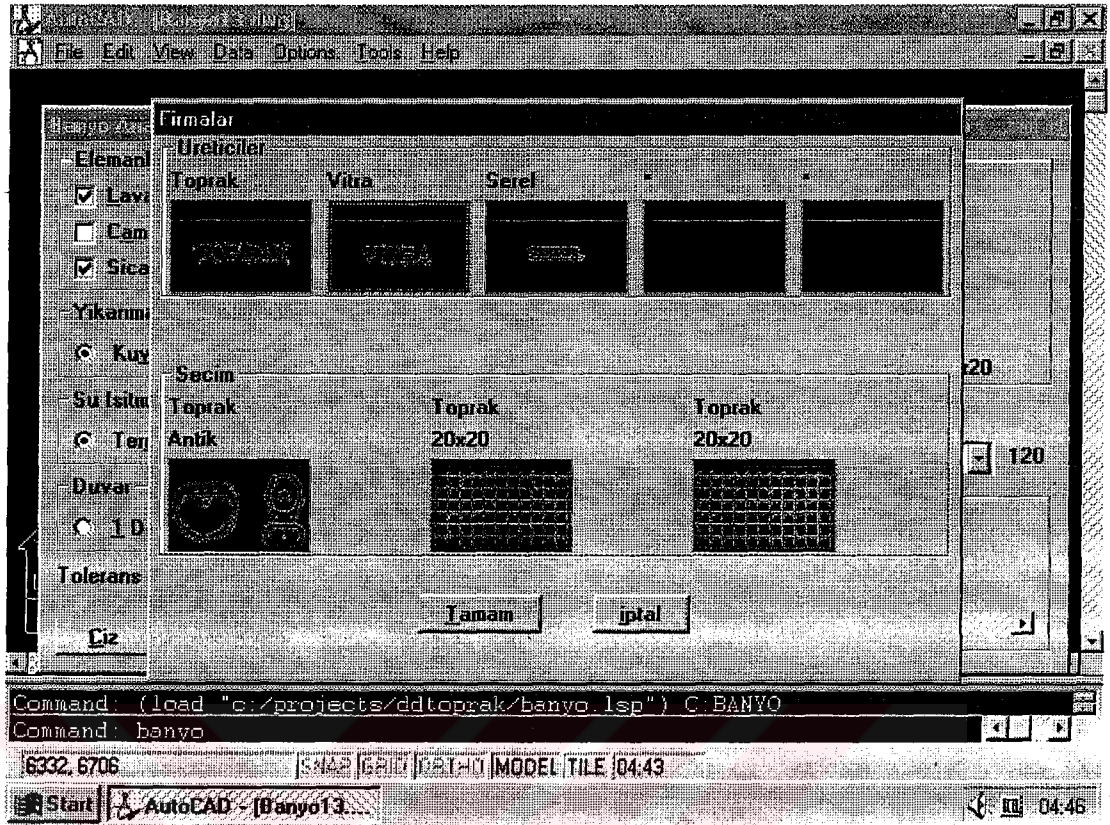
Bu komutun verilmesiyle beraber ilk ekran kullanıcının karşısına çıkar (Şekil 3.1). Bu ekranda sol sütunda kullanılabilecek elemanların isimleri, yıkanma ve sıcak su için seçenek elemanlar, ve en altta da kullanılacak duvar sayısı bulunur. Sağ sütunda ise seçilmiş vitrikiye takımının ve yer karosunun "slide" ları ve isimleri, duvar karosunun adı ve markası, o an için işaretlenmiş olan kombinezon,



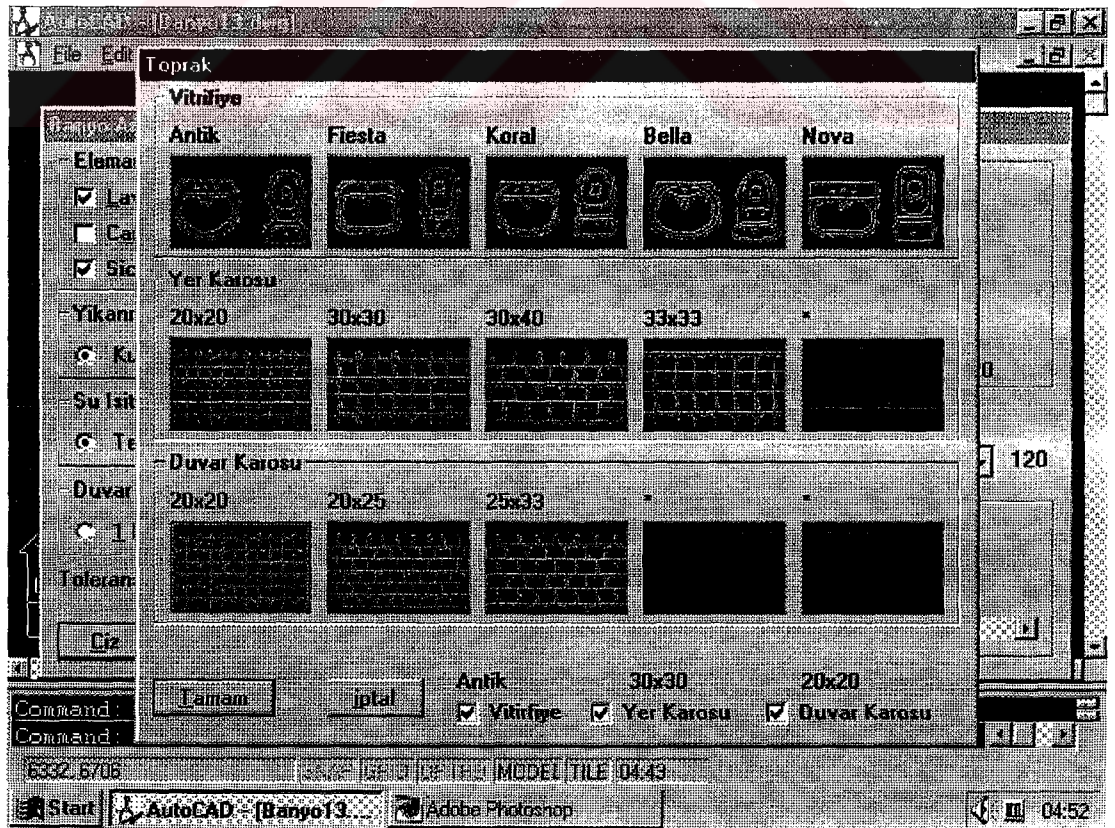
Şekil 3.1 Birinci Ekran : Ana Menü.

belirlenmiş kombinezon için permütasyonları üreten komut tuşu, türetilmiş olasılıkları gösteren aşağı açılabilir bir liste, ve son olarak da çizilecek olasılığı ya da olasılıkların sayısını belirlemeye yarayan seçenekler bulunur.

Kullanıcı seçili olan vitriye takımında, yer ve duvar karosunda bir değişiklik yapmak isterse vitriye ve yer karosunun "slide" larının üzerine fareyle tıklar. Bu tıklamayla birlikte ikinci ekran açılır (Şekil 3.2). Bu ekranın ilk satırında, o anda Banyo.dta dosyasında kayıtlı olan firmaların isimleri bulunur. Alt sıradá ise vitriye, yer karosu ve duvar karosu için yapılmış seçimler "slide" halinde görülür. Kullanıcı üst sıradaki firma isimlerinden birine tıkladığında, tıklanan firmaya ait yeni bir ekran açılır (Şekil 3.3). Bu üçüncü ekran yatayda 4 e bölünmüştür. İlk satıra Banyo.dta dosyasında bulunan, seçilen firmaya ait vitriye takımları yerleştirilir. İkinci sıraya yine Banyo.dta dosyasından alınan, firmaya ait yer karoları yerleştirilir. Aynı şekilde üçüncü sıraya duvar karoları doldurulur. Son sırada ise yapılan seçimlerden hangilerinin bir önceki ekrana aktarılacağını belirleyen işaret kutucukları (check box) bulunur.



Şekil 3.2 İkinci ekran : Firmalar Ana Menü.



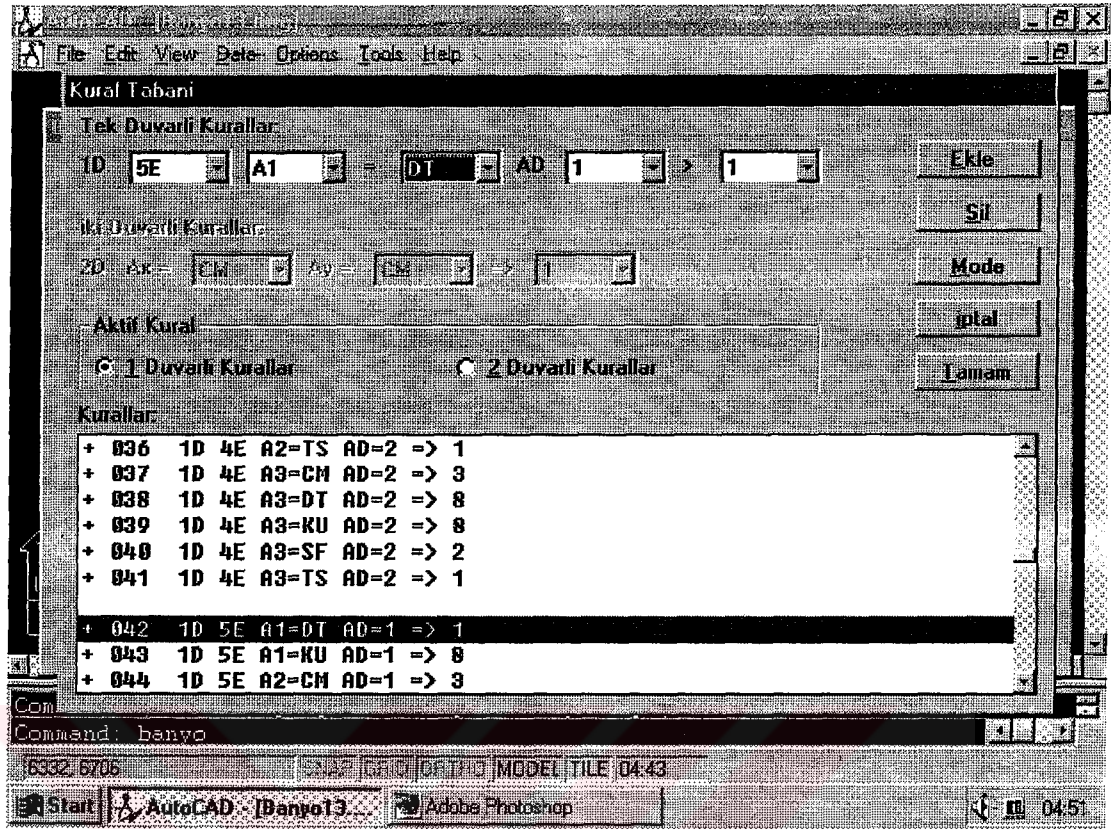
Şekil 3.3 Üçüncü ekran : Firmalar Ara Ekranı.

Kullanıcı 3. ekrandan uygun gördüklerini seçip Tamam tuşuna bastığında bir önceki ekrana (2. ekrana) geri döner. Yapılan seçimler resimleriyle birlikte alt sırada yerlerini alırlar. Kullanıcı, eğer tercih ederse, vitrifiye takımını, yer karosunu, ve duvar karosunu farklı firmalardan seçebilir. Burada yapılacak seçimler tamamlandığında kullanıcı tamam tuşuna basarak bir önceki ekrana (1. ekrana) geri döner. Geri döndüğünde, vitrifiye ve yer karosu resimleriyle birlikte diğer yazılar, yapılan seçimlere göre güncellenir. Vitrifiye ve karolarının seçimi yapıldıktan sonra sıra kullanılacak elemanların seçimine gelir.

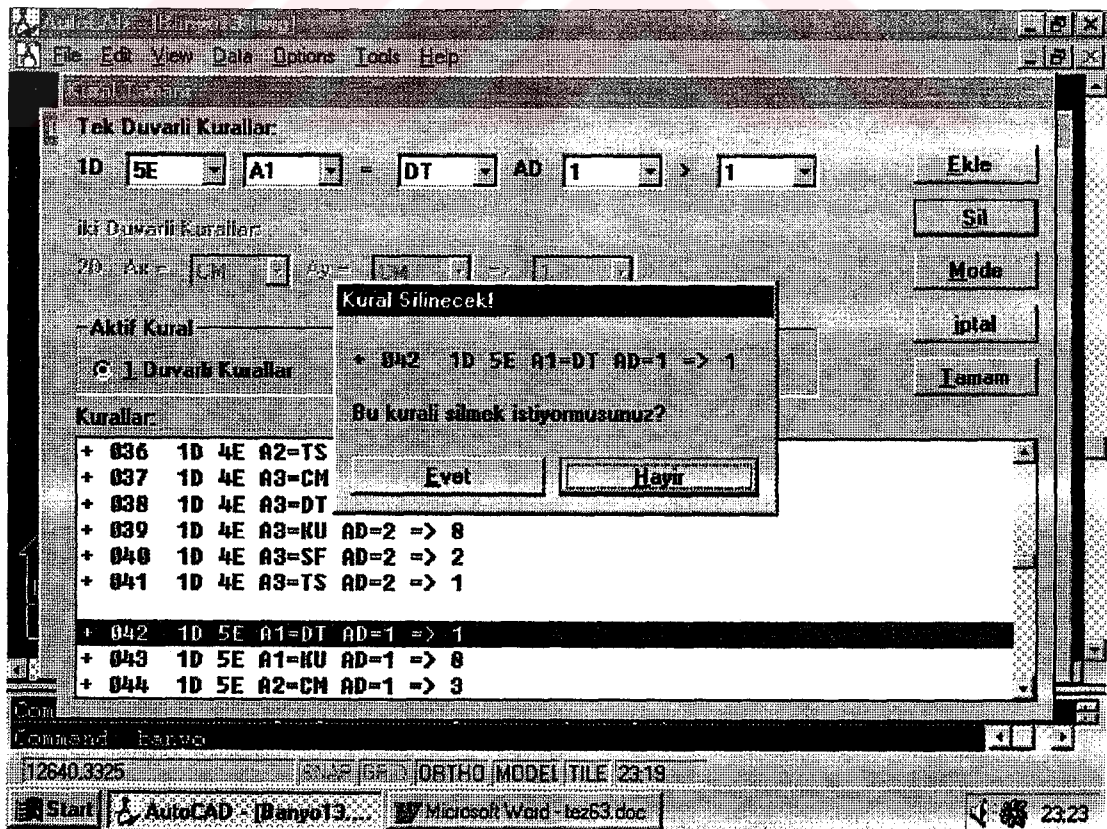
Kullanıcı lavabo, klozet ve diğer elemanların işaret kutularına basıp bu elemanların seçim durumlarını değiştirdikçe kombinezonun yazılı olduğu kutucuk (text box) sürekli güncellenir. Aynı durum sıcak su ve yıkanma seçeneklerini içeren alan (boxed_radio_row) için de geçerlidir. Burada yapılan değişiklikler kombinezonu değiştirir. Benzer durum Duvar Sayısı alanı için de geçerlidir. Kombinezonda bir değişiklik meydana gelirse, bu kombinezon için üretilmiş olan olasılıkların yazılı olduğu kutucuğun (pop-up list) içeriği boşaltılır ve Çiz komutu etlisizleştirilir. Kullanılacak duvar sayısını değiştirmek kombinezona bir etkide bulunmaz fakat permütasyon listesini sıfırlar.

Kullanıcı kombinezona etkide bulunacak seçimlerin hepsini tamamladıktan sonra permütasyon tuşuna basar. Bu tuşa basılınca program o kombinezon için geçerli olan, 2 duvar ya da tek duvar kullanan, tüm olasılıkları bir listeye doldurur. Bu hesaplama bittikten sonra Çiz tuşu kullanıma açılır.

Kullanıcı kural tabanına Kural tuşuna basarak ulaşır. Bu tuşa basılınca kural tabanı kullanıcı arabirimi açılır (Şekil 3.4). Bu ekranda, imleç bir kural üzerinde iken Mode tuşuna basılırsa üzerinde bulunulan kural silinmez, fakat etkisizleşir. Bunu ifade etmek için kuralın başında bulunan "+" işareti "-" ye dönüşür. Yine imleç bir kural üzerinde iken Sil tuşuna basılırsa, program bu kural silineceğine dair kullanıcıyı uyarır (Şekil 3.5), evet cevabı alınırsa üzerinde bulunulan kural veri tabanından silinir. Ekle tuşuna basıldığında ise o an aktif bulunan kural grubuna ait kural veri tabanına kaydedilir. Ekranı Tamam tuşuyla terketmek, yapılan bütün değişikliklerin veri tabanına kaydedilmesini sağlar. İptal tuşu ise, veri tabanını ilk haline getirerek bu ekranı kapatır.



Şekil 3.4 Kural Tabanı Arabirimi



Şekil 3.5 Kural Tabanı Arabirimi : Silinecek Kural için Uyarı

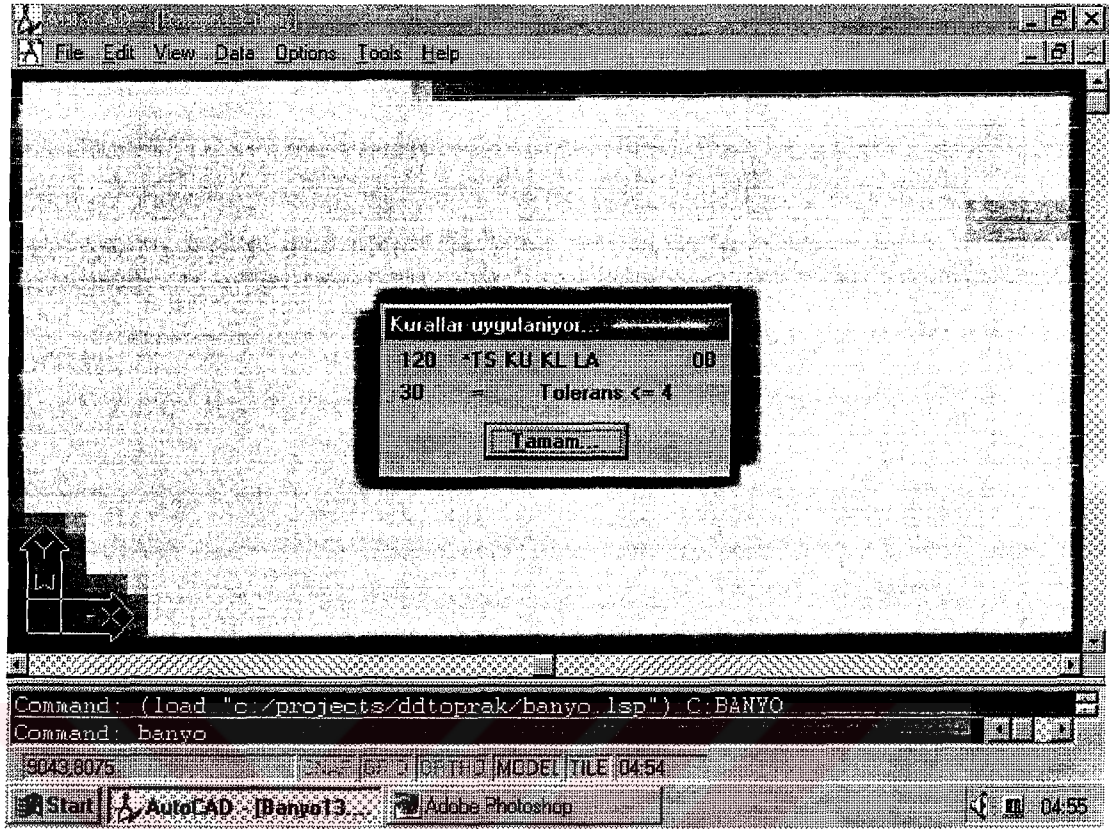
Kullanıcı son yaptığı seçimlerden memnun ise Çiz tuşuna basarak değerlendirme ve çizim sürecini başlatır. Bu tuşa basılmasıyla birlikte seçimlerin yapıldığı 1. ekran kaybolur. Birinci ekran kaybolurken kullanıcının bu ekrana girmiş olduğu bilgiler Banyo.tmp dosyasına yazılır.

Birinci ekrandan sonra geçici ve kullanıcı müdahalesine açık olmayan bir ekran gelir (Şekil 3.6). Programın kural tabanından kuralları aldığı ve bu kurallarla olasılıkları değerlendirdiği aşama, değerlendirilen olasılık sayısına göre değişmekle beraber, görece uzun sürdüğünden, kullanıcının programın çalışmakta olduğundan haberdar edilebilmesi için bu ekran hazırlanmıştır. Ekranın ilk satırında değerlendirilmiş olan olasılık sayısı, o anda değerlendirilmekte olan olasılık ve hata puanı, ikinci satırında ise hata puanı toleransın altında olan olasılıkların sayısı ve tolerans değeri bulunur. Kullanıcı işlem bitene kadar bu sürece müdahale edemez. Eğer hiçbir olasılık tolerans değerinden daha küçük bir hata puanına sahip değilse, program kullanıcıyı uyarır ve ana ekrana (Şekil 3.1) geri döner. Değerlendirme işlemi tamamlandığında Devam tuşuna basılarak devam edilir.

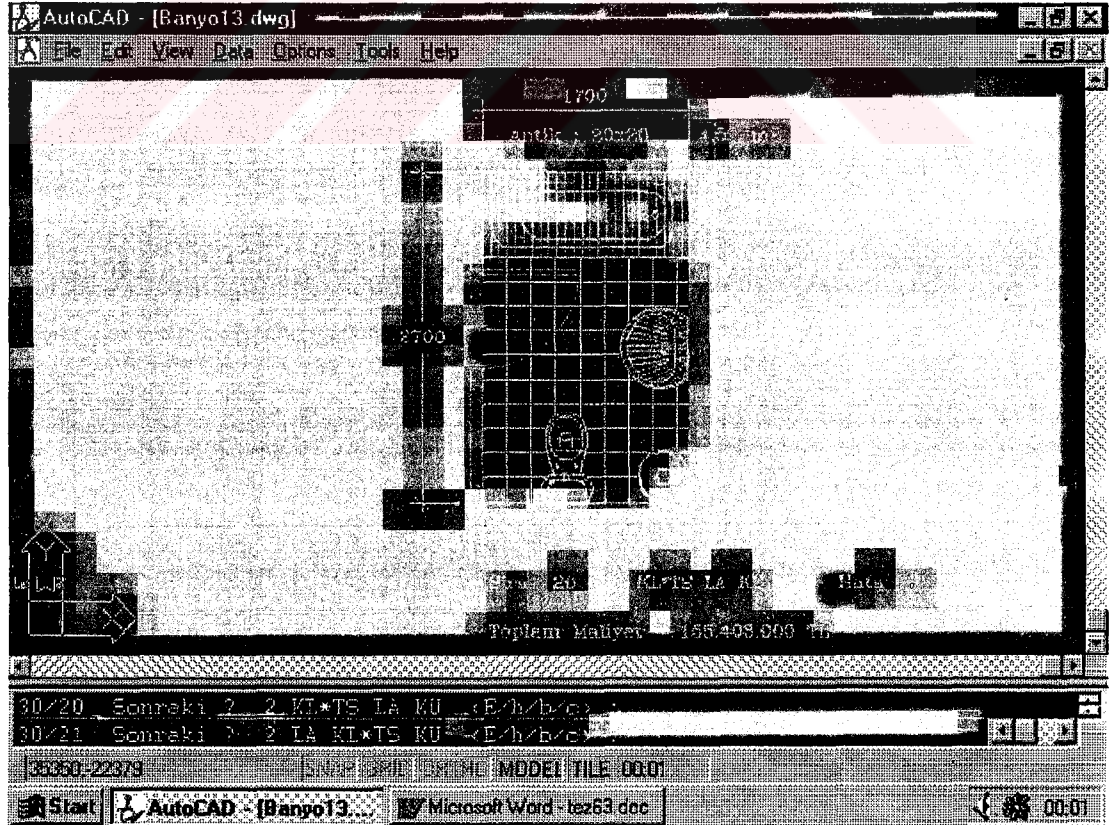
Bu noktadan sonra puan sırasına göre (küçükten büyüğe doğru) olasılıklar çizilir. Program her aşamada çizilecek olan olasılık hakkında kullanıcıya bilgi verir:

30/20 Sonraki? 2 KL*TS LA KU <E/h/b/c> :

Bu satırda sırasıyla puanı uygun olasılıkların toplam sayısı, çizilecek olan olasılığın hata puanı, sıra numarası ve adı bulunur. Kullanıcı "E" tuşuna basarsa sıradaki olasılık çizilir. "H" tuşuna basılırsa bir sonraki olasılığa geçilir. "C" tuşuna basılırsa programdan çıkar. "B" tuşuna basılırsa çizilen tüm olasılıklar topluca gösterilir, ve kullanıcıya devam etmek isteyip istemediği sorulur. Cevap "H" ise program çalışmayı durdurur. Cevap "E" program dosyada yapılmış çizimler bulunduğunu, bunların silinip silinmeyeceğini sorar. Çizimler incelenmek isteniyorsa ve silinmeyecekse programdan çıkılıp AutoCAD komut satırına dönülür. Çizimler silinip devam edilmek isteniyorsa, birinci ekran yüklenir (Şekil 3.1), Banyo.tmp dosyası açılır ve kullanıcının birinci ekrana girmiş olduğu bilgiler alınarak ekran tazelenir.



Şekil 3.6 Altıncı Ekran: Olasılıkların Değerlendirilmesi.



Şekil 3.7 Yedinci Ekran : Puanı Uygun Olan Olasılıkların Çizilmesi.

AutoLISP derleyici (compiler) değil de komut yorumcusu (command interpreter) bir dil olduğu için, AutoLISP programları derlenmiş (C, Pascal, vb.) programlara göre daha yavaş çalışır. ASCII formatında metin olarak saklanan AutoLISP dosyaları her seferinde tekrar “yorumlanırlar”. Bu çok yavaş bir işlemdir. Bununla beraber program çok sayıda tekrar içeren ve çalışmayı yavaşlatan bir çok algoritmaya sahiptir. Bu algoritmalar “bir noktaya kadar” optimize edilmişlerdir. Bu nedenle program zaman zaman kullanıcıyı bekletmektedir. Programın geliştirildiği donanımda;

- a. Dört elemanlı ve tek duvarlı 24 olasılığı üretmek 0.3 sn.,
- b. Dört elemanlı ve iki duvarlı 120 olasılığı üretmek 0.6 sn.,
- c. Beş elemanlı ve iki duvarlı 720 olasılığı üretmek 1 sn.,
- d. Beş elemanlı ve iki duvarlı 120 olasılığın kural tabanındaki 105 kural cümlesi tarafından değerlendirilmesi, ve puan sırasına dizilmesi 2 sn.,
- e. Beş elemanlı ve iki duvarlı 720 olasılığın, kural tabanındaki 105 kural cümlesi tarafından değerlendirilmesi, ve puan sırasına dizilmesi 25 sn.,

sürmektedir. Çok tekrarlanan döngüler içindeki algoritmalar “optimize” edilir ve bazı süreçler birleştirilirse bu süreler %20 ile % 50 arasında kısalabilir. Program tek bir sonucu çizmek için 3 sn. ye ihtiyaç duymaktadır. Tüm sonuçların ekrana yansıtılması ekrana kaç olasılığın basıldığına bağlıdır.

Programın ekranlarında Türkçe harf sorunu vardır. Bu sorun DOS ortamında Config.sys dosyasına

```
device = C:\dos\ansi.sys
```

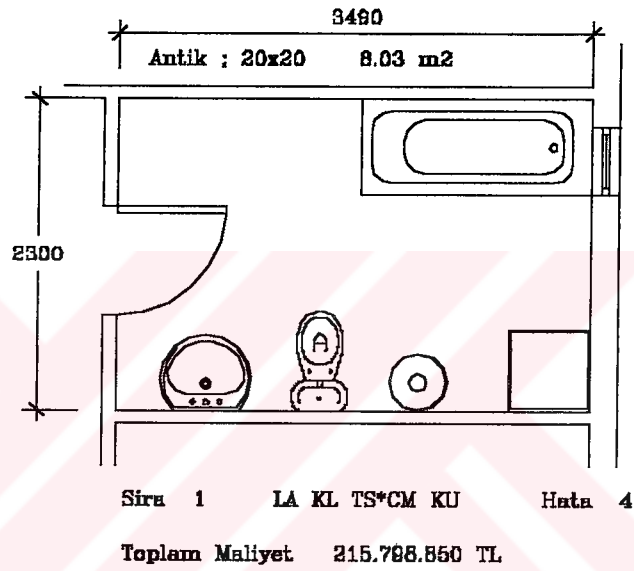
ya da

```
devicehigh = C:\dos\ansi.sys
```

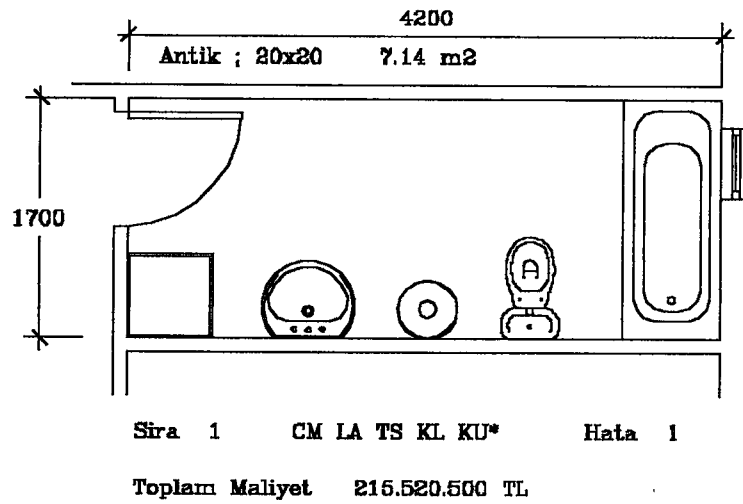
satırı eklenerek çözülebilirdi. Fakat programın, ekrana okunabilir birşeyler basabilmek için, kullanıcısına bu tür bir ayarlamayı şart koşmasının gereksiz bir zorlama olacağı düşünülerek, İ, ı, Ş, ş, Ğ, ğ harfleri yerine I, i, S, s, G, g harfleri kullanılmıştır.

3.2 Programın Çıktıları

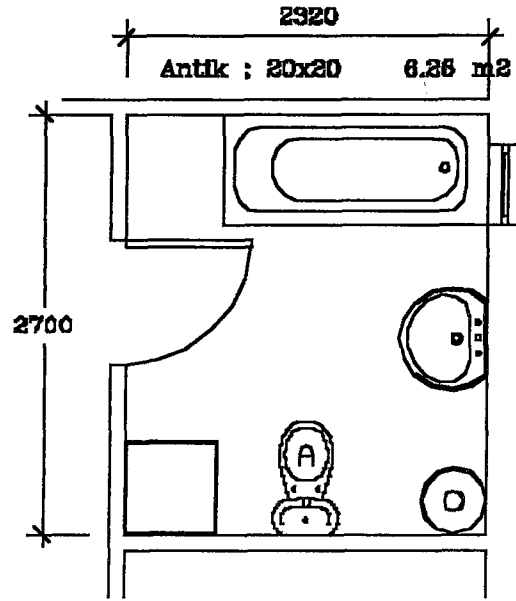
Aşağıda programa ait bir dizi çizim sunulmuştur. Çizimler kullanılan eleman sayısına göre sınıflandırılmışlardır. Orijinal çizimlerde bulunan renk ve döşeme kaplaması, sunumu kolaylaştırmak amacıyla kaldırılmıştır, ve çizimler ölçeksiz olarak basılmıştır.



Şekil G.1 Beş elemanlı çözüm; LA KL TS*CM KU



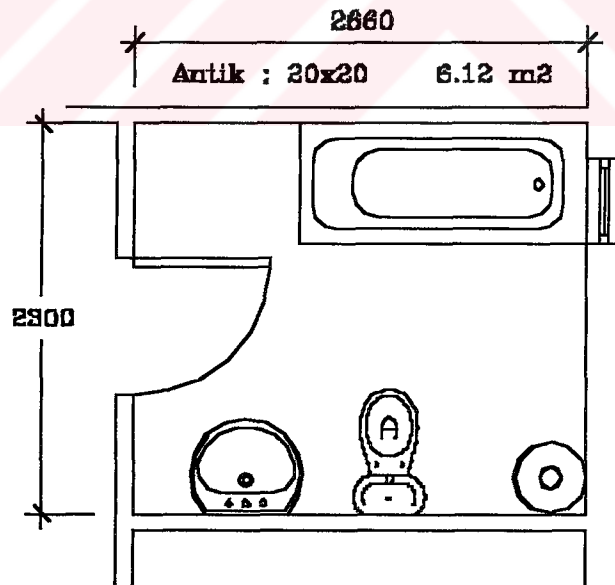
Şekil G.2 Beş elemanlı çözüm; CM LA TS KL KU*



Sıra 1 CM KL*TS LA KU Hata 2

Toplam Maliyet 210.173.900 TL

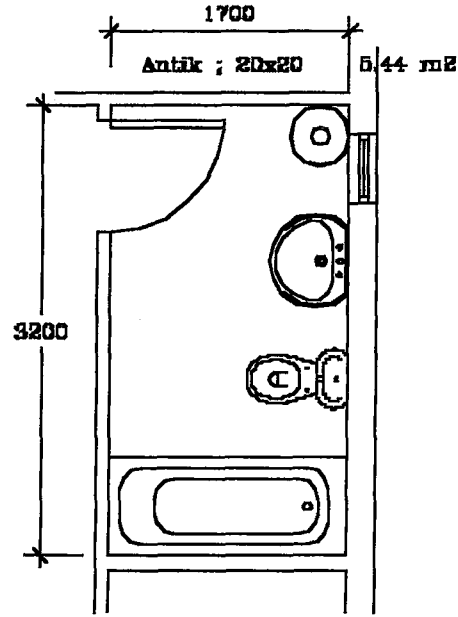
Şekil G.3 Beş elemanlı çözüm; CM KL*TS LA KU



Sıra 1 LA KL*TS KU Hata 2

Toplam Maliyet 159.727.400 TL

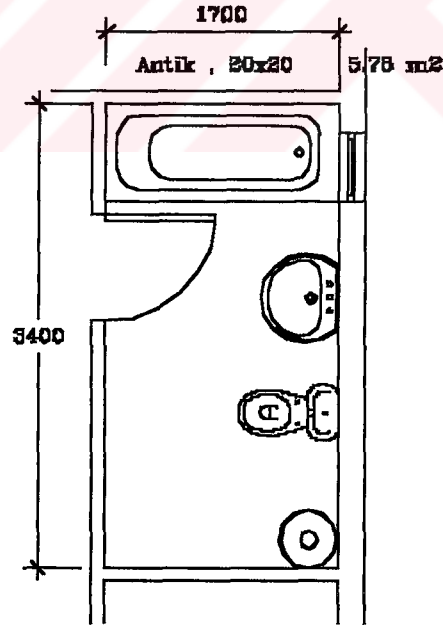
Şekil G.4 Dört elemanlı çözüm; LA KL*TS KU



Sıra 1 *KU KL LA TS Hata 0

Toplam Maliyet 158.775.500 TL

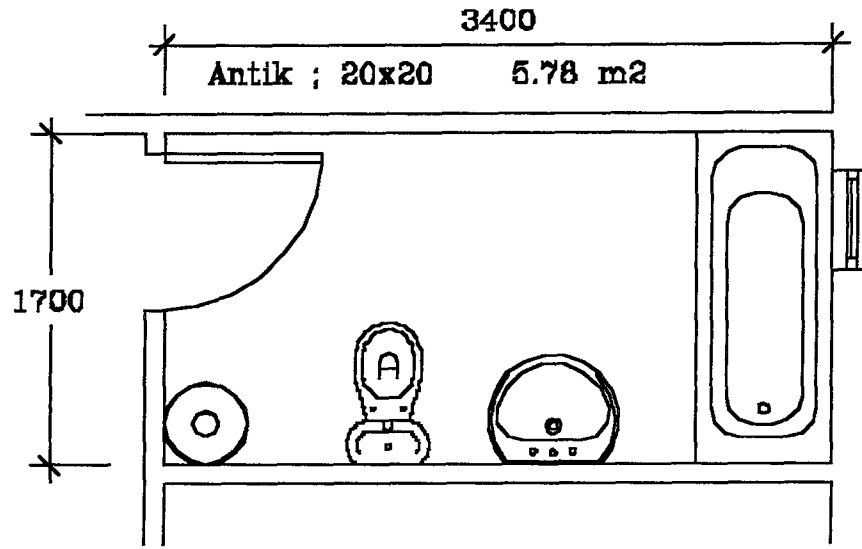
Şekil G.5 Dört elemanlı çözüm; *KU KL LA TS



Sıra 1 *TS KL LA KU Hata 0

Toplam Maliyet 160.124.600 TL

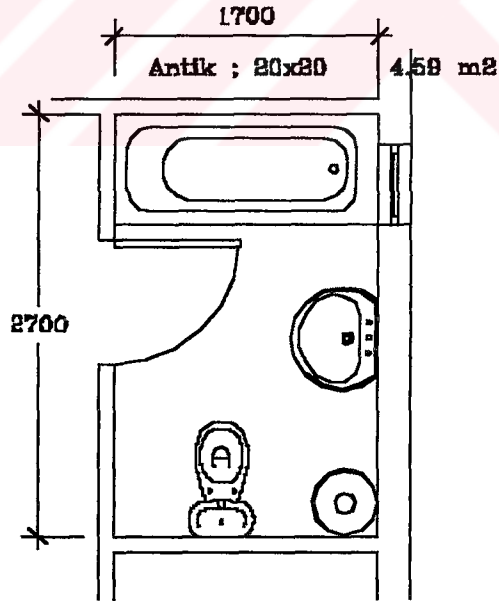
Şekil G.6 Dört elemanlı çözüm; *TS KL LA KU



Sıra 1 TS KL LA KU* Hata 0

Toplam Maliyet 160.124.500 TL

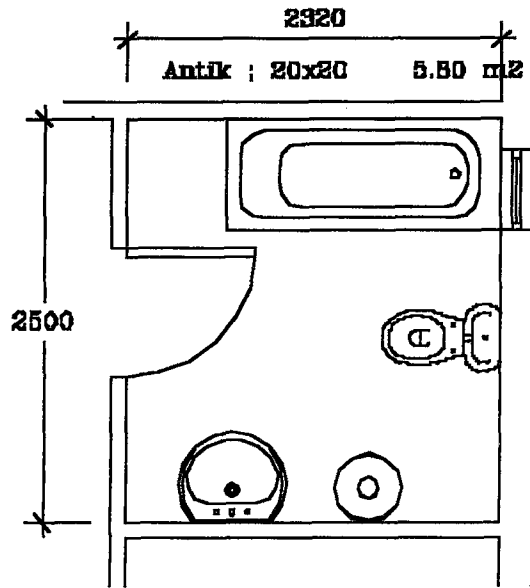
Şekil G.7 Dört elemanlı çözüm; TS KL LA KU*



Sıra 1 KL*TS LA KU Hata 2

Toplam Maliyet 155.403.000 TL

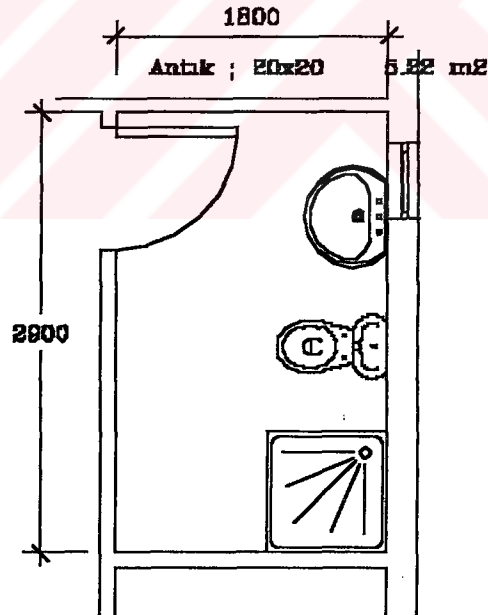
Şekil G.8 Dört elemanlı çözüm; KL*TS LA KU



Sıra 1 LA TS*KL KU Hata 3

Toplam Maliyet 158.707.100 TL

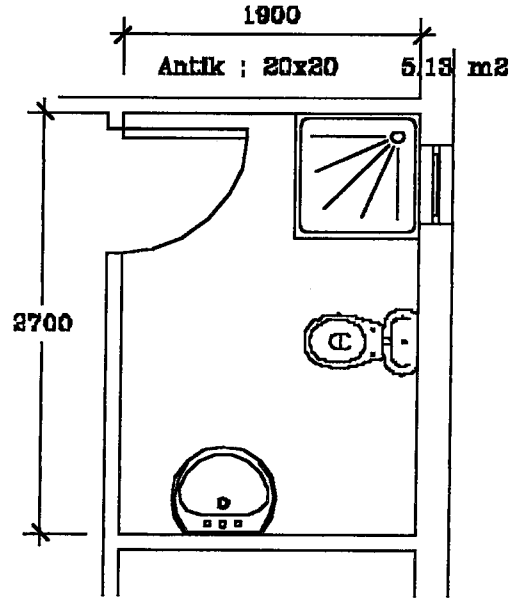
Şekil G.9 Dört elemanlı çözüm; LA TS*KL KU



Sıra 1 *DT KL LA Hata 0

Toplam Maliyet 107.540.500 TL

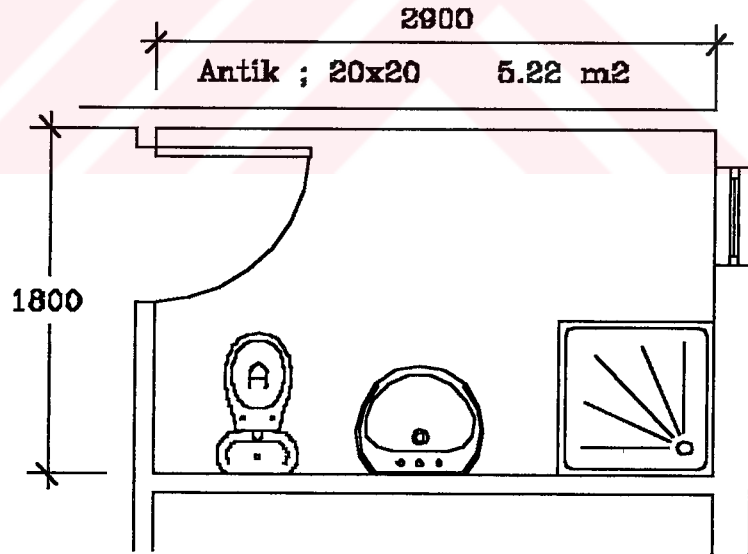
Şekil G.10 Üç elemanlı çözüm; *DT KL LA



Sıra 1 LA*KL DT Hata 2

Toplam Maliyet 106.942.000 TL

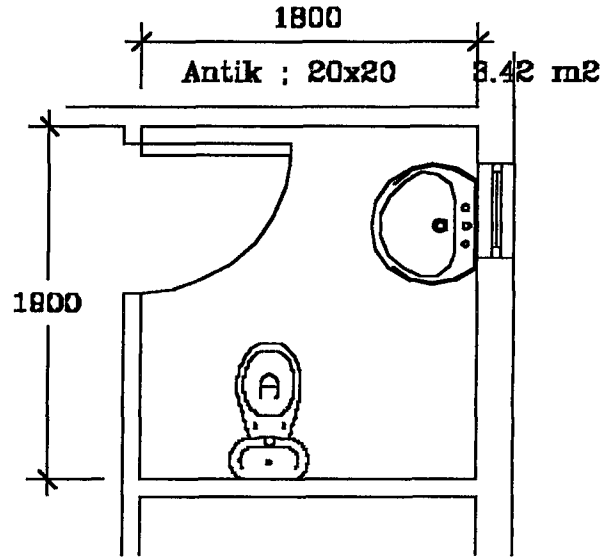
Şekil G.11 Üç elemanlı çözüm; LA*KL DT



Sıra 1 KL LA DT* Hata 0

Toplam Maliyet 107.540.500 TL

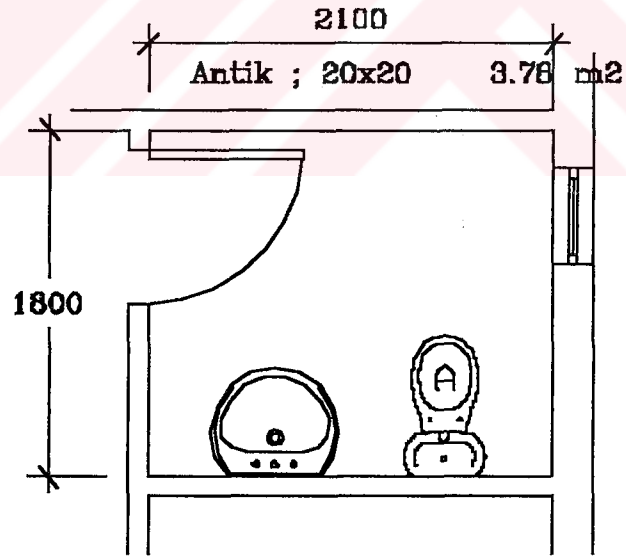
Şekil G.12 Üç elemanlı çözüm; KL LA DT*



Sıra 1 KL*LA Hata 2

Toplam Maliyet 50.700.500 TL

Şekil G.13 İki elemanlı çözüm; KL*LA



Sıra 1 LA KL* Hata 0

Toplam Maliyet 52.088.500 TL

Şekil G.14 İki elemanlı çözüm; LA KL*

SONUÇLAR ve ÖNERİLER

Amaç, kural tabanlı bir US geliştirmektir. Kural tabanının kullanıcı katılımına açık olması önemli bir alt amaç olarak konmuştur.

Aşağıda sıralanan eksiklerin dışında, uygulanabilir, tefrişi yapılmış banyo tasarımları geliştirebildiği göz önüne alınarak, sistemin amaçlarını gerçekleştirebildiği söylenebilir.

Programda tespit edilen hatalar ya da eksikler şöyle sıralanabilir:

1. Kural tabanı kullanıcıya sunulurken doğal dir kullanılması gerekir. Kural tabanını kullanıcıya açmaktaki amaç, kullanıcı katılımını sağlamak, ve kural tabanının güncelleştirilebilmesini kolaylaştırmak olduğu için, kullanıcıyla iletişimin, tıpkı programın diğer ekranlarında olduğu gibi, kullanıcının doğal dili üzerinde olması gerekmektedir.
2. Tanımlanmamış kurallar tasarımda hatalara yol açmaktadır. Programın son durumunda, kural tabanında 105 adet kural bulunmaktadır. Tanımlanabilir kuralların sayısı belirsizdir. Bu nedenle, eksik kuralların saptanması ve kural tabanına girilmesi, ancak programın uzun süreli kullanımı sonunda olabilir.
3. Kural tabanı sadece, elemanların banyo içindeki, ya da birbirlerine göre olan konumlarıyla ilgili yerleşim kuralları içerir. Tefrişin yapılabilmesi için gerekli olan, elemanların birbirlerine göre konumları, ve banyo içindeki konumlarıyla ilişkilendirilmiş boyutsal tefriş kuralları kural tabanı dışında tutulmuştur. Bu durumda, ortaya çıkan hatalı durumların düzeltilmesi ancak programın koduna müdahaleyle mümkün olabilmektedir. Bu sorunu gidermek için, bu durumları içeren kuralları kural tabanına, ve kurallarla ilgili çıkarım mekanizmasını da program koduna eklemek gerekmektedir.

Program geliştirilirken aşağıdaki öneriler dikkate alınabilir:

1. Programın kodu, yeni elemanların eklenmesini (banyo dolabı, vb.), grafik arabirimi aracılığıyla kullanıcı tarafından olanaklı kılacak şekilde geliştirilmelidir.
2. Bilgileri saklamak amacıyla text dosyaları yerine gerçek veritabanı dosyaları kullanılmalıdır. Bu sayede dosyaların diğer programlar tarafından okunabilmesi sağlanır, dosyalardaki verilere ve bilgilere erişim kolaylaşır, program hız kazanır, ve verilerin güvenliği sağlanmış olur.
3. Tesisatla ilgi karar durumları sisteme dahil edilebilir.
4. Kapının konumu ve açılış yönü daha esnek olabilir. Buna ait kurallar tanımlanabilir.
5. Markalar, modeller, ve diğer elemanların ayrıntılı bilgilerini içeren veritabanına (banyo.dta) erişim için bir grafik arabirim geliştirilebilir.
6. Programın varsayılan (default) ayarlarının bilgilerini içeren veritabanına (banyo.def) erişim için bir grafik arabirim geliştirilebilir.
7. Kullanıcıya, tasarımda kullanılan elemanların ayrıntılı teknik bilgileri bir grafik arabirim aracılığıyla sunulabilir.
8. Çizim aşamasında, çizilecek olasılıklar arasında ileriye ve geriye doğru harekete imkan tanıyan bir arabirim tasarlanabilir.

KAYNAKLAR

- ALPHAN, A., (1985), Yapıda Sağlık Donatımı, İstanbul Teknik Üniversitesi Matbaası, İstanbul.
- ALEV, G., (1994), Mimari Tasarlama Eğitiminde Yazlık Konut Tasarımı Amacıyla Kullanılabilecek Bir Uzman Sistem : ZEUS, İstanbul Teknik Üniversitesi Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi.
- AUTODESK, (1988), AutoLISP Release 10 Programers Reference, Switzerland, Autodesk BV.
- AUTODESK, (1992a), AutoLISP Release 12 Reference Manual, Switzerland, Autodesk BV.
- AUTODESK, (1992b), AutoLISP Release 12 Programers Reference, Switzerland, Autodesk BV.
- AUTODESK, (1992c), AutoCAD Release 12 Customization Manual, Switzerland, Autodesk BV.
- AUTODESK, (1992d), AutoCAD Release 12 Extras Manual, Switzerland, Autodesk BV.
- BAHÇECİOĞLU, B., (1995), Bilgisayar Ortamında Üretilmiş Planlar için Otomatik Cephe Üretilmesi, İstanbul Teknik Üniversitesi Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi.
- Banyo ve Mutfak, (1997), Colombo Design, Banyo ve Mutfak, sayı 4, Kış, sayfa 36, 78.
- BARUT, M., (1991), Uzman Sistemler ve Deterministik Yöneylem Araştırması Tekniklerinde Bir Uygulama, İstanbul Teknik Üniversitesi Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi.
- BAYAZIT, N., ALPHAN, A., (1980), Banyo ve Tuvalet Tipolojisi Araştırması, İ.T.Ü. Rektörlüğü Döner Sermaye İşletmesi.
- BAYAZIT, N., (1994), Endüstri Ürünlerinde ve Mimarlıkta Tasarlama Metodlarına Giriş, Literatür Yayınları, İstanbul, Mayıs 1994.
- BAYAZIT, N., KAVAKLI, M., (1992a), Uzman Sistem Metodları I, Bilgisayar Magazin, cilt 1, sayı 14, Mart, sayfa 60-67.
- BAYAZIT, N., KAVAKLI, M., (1992b), Uzman Sistem Metodları II, Bilgisayar Magazin, cilt 1, sayı 15, Nisan, sayfa 57-77.

- BUHANAN, B. G. ve Witchel, T. M., (1977), Model Directed Learning Production Rules, Computer Science Department, Stanford Uni, Stanford, ABD (bu kaynak Hayes-Roth, 1983 tarafından verilmiştir).
- CEBECİ, U., (1994), Hücreyel İmalatın Başlangıç Aşamaları için Uzman Sistem Yaklaşımı, İstanbul Teknik Üniversitesi Fen Bilimleri Enstitüsü, Doktora Tezi.
- ÇAKAR, T., (1991), Uzman Sistem Teknolojisi ve Çizelgeleme Uygulaması, İstanbul Teknik Üniversitesi Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi.
- FAIN, J., GORLIN, D., HAYES-ROTH, F., (1981), The ROSIE Language Reference Manuel, Rand Corp., Santa Monica, California, (bu kaynak Hayes-Roth, 1983 tarafından verilmiştir).
- FEIGENBAUM, E.A., McCOURDUCK, P., (1983) The Fifth Generation, Reading, Massachusetts, Addison -Wesley Publishing Company.
- GÖÇEN, M., (1986), Yıkama Fonksiyonlarının Konut Tasarımına Etkisi ve Yıkama Mekanı Tasarımı, Mimar Sinan Üniversitesi, Fen Bilimleri Enstitüsü, İçmimarlık Yüksek Lisans Tezi.
- GEVARTER, W. B., (1985), Intelligence Machines, An Introductory Perspective of Artificial Intelligence Robotics, Perintice-Hall Inc., Englewood Cliffs, New Jersey, USA, (bu kaynak Çakar, 1991 tarafından verilmiştir).
- HAYES-ROTH, F., (1983), Building Expert Systems, London, Addison-Wesley Publishing Company.
- PASCA, V., (1991), The Bathroom Between Mith and Reakity, Bagno 90-91, cilt 4, sayı 70, sayfa 4-15, 48, 74, 126.
- KAVAKLI, M., (1995), WIND: Aknowledge Based System for The Synthesis of Window Parts, İstanbul Teknik Üniversitesi Fen Bilimleri Enstitüsü, Doktora Tezi (İngilizce Hazırlanmıştır).
- KAVAKLI, M., (1990), U-MuT AutoLISP'le Bilgisayar Destekli Uzman-Mutfak Tasarımı, İstanbul Teknik Üniversitesi Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi.
- SPILER, J., MARINCEK, J., (1995), LISP2C : AutoLISP to C (ADS) Translator User's Manual, Ljubljana, Slovenia, Adresi: www.basic.si/lisp2c.html.
- MACK, J., MANSER, J., (1982), Home Design, Londra.
- MADNI, A. M., (1988), The Role of Human Factors in Expert Systems Design and Acceptance, The Human Factors Society Inc., California, USA, (bu kaynak Çakar, 1991 tarafından verilmiştir).
- MANSER, J., (?), The Kitchen and Bathroom Book.

McCARTHY, J. M., History of LISP, Artificial Intelligence Laboratory, Stanford Üniversitesi, (bu kaynak internetten temin edilmiştir. Adresi: <http://www-formal.stanford.edu/jmc/history/lisp/lisp.html>, şeklindedir).

NEUFERT, E., (1989), Yapı Tasarım Bilgisi, İstanbul.

ÖZKUL, S., (1991), Yapay Zeka Uygulamaları, İstanbul Teknik Üniversitesi Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi.

ÖZMEN, Günay (1992), AutoCAD Kullanıcıları için AutoLISP, İstanbul, yayınlanmamış ders notları.

ROSENMAN, M. A., vd., (1987), Expert Systems for Design Application, Quinlan, J. R. nin, Applications of Expert Systems, kitabında yer almaktadır. London, Addison-Wesley Publishing Company.

QUINLAN, J. Ross, (1987), Applications of Expert Systems, London, Addison - Wesley Publishing Company.

SEREL VİTRİFİYE VE FAYANS MONTAJ KATALOĞU, (1995), Serel Seramik Sıhhi Tesisat Gereçleri San. Ve Tic. AŞ.

SHORTLIFFE, E. H., (1976), Computer Based Medical Consultation: MYCIN, New York, American Elsevier, (bu kaynak Hayes-Roth, 1983 tarafından verilmiştir).

SILVERMAN, Barry G., (1987), Expert Systems for Business, London, Addison - Wesley Publishing Company.

SLAGLE, J. R., (1961), A Hueristic Program That Solves Symbolic Integral Problems in Freshman Calculus: Symbolic Automatic Integrator (SAINT), MIT doktora tezi, ABD (bu kaynak Hayes-Roth, 1983 tarafından verilmiştir).

TOPRAK SERAMİK VİTRİFİYE VE FAYANS MONTAJ KATALOĞU, (1995), Toprak Holding.

VİTRA VİTRİFİYE VE FAYANS MONTAJ KATALOĞU, (1994), Vitra ve DADA-Dağıtım ve Pazarlama AŞ.

WATERMAN, D. A., (1986), A Guide to Expert Systems, Addison-Wesley, Reading, MA.

WEISS, S. M. ve Klukowski, C. A., (1979), EXPERT: A System for Developing Consultation Models, ABD (bu kaynak Hayes-Roth, 1983 tarafından verilmiştir).

WIIG, K., (1990), Expert Systems: A Manegers Guide, Genova, International Labour Office, Genva.

WINSTON, P. H., (1987), Commercial Debut of Artificial Inteligence, Quinlan, J. R. nin, Applications of Expert Systems, kitabında yer almaktadır. London, Addison-Wesley Publishing Company.

WINSTON, P. H., Horn, B. K. P., (1989), LISP, Addison-Wesley, Reading, Massachusetes.

YAPI KATALOĐU, (1986), Yapı Endüstri Merkezi Yayınları.

YAPICIOĐLU, N., (1991), Uzman Sistemler ve Uygulamaları, İstanbul Teknik Üniversitesi Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi.



EKLER

EK A Kombinezon Tablosu

Kullanıcıya, arasından seçim yapabileceği 5 ayrı eleman verilmiştir. Bunlar arasından lavabo ve klozet için model ve marka seçimi olanaklı kılınmıştır. Bu yalnızca tasarımın maliyetini etkilemektedir. Çamaşır makinesi için ise yalnızca fiyatı değiştirme imkanı bulunmaktadır. Yıkama elemanları için duş teknesi ve küvet şeklinde iki ayrı seçenek bulunmaktadır. Aynı şekilde sıcak su için de şofben ve termosifon gibi iki seçenek bulunmaktadır.

Kısaltmalar :

LA	:	Lavabo	
KL	:	Klozet	
CM	:	Çamaşır makinesi	
DT	:	Duş teknesi	
KU	:	Küvet	
SF	:	Şofben	
TS	:	Termosifon	için kullanılmıştır.

Bu kısaltmalar programın kodu içinde de bu şekilde kullanılmıştır.

Kullanıcı 5 elemandan ve bir duvardan oluşan bir tasarım geliştirmek istediğinde elinde 4 ayrı kombinezon bulunur:

CM SF LA KL DT
 CM SF LA KL KU
 CM TS LA KL DT
 CM TS LA KL KU

Aslında 5 in 5 li kombinezonu 1 olduğu halde, yıkama ve sıcak su elemanlarının kendi aralarında değişmelerinden dolayı kombinezon sayısı 4 e çıkmıştır. Program bu kombinezonları kullanarak tüm 5 elemanlı durumları üretir.

CM SF **LA KL DT**
 CM SF **LA DT KL**
 CM SF **KL LA DT**

CM SF **KL DT LA**
 CM SF **DT KL LA**
 CM SF **DT LA KL ...**

şeklinde devam eden permütasyon üretimi sonucunda

$$5! = 5 \times 4 \times 3 \times 2 \times 1 = 120 \quad 120 \times 4 = 480$$

adet 5 elemanlı olasılık buluyoruz. Kullanıcı kullanılan duvar sayısını 2 ye çıkardığında ise program permütasyon hesabını iki duvara dağıtarak yapar:

CM TS LA KL KU

gibi tek bir olasılıktan

*CM TS LA KL KU
 CM*TS LA KL KU
 CM TS*LA KL KU
 CM TS LA*KL KU
 CM TS LA KL*KU
~~CM TS LA KL KU*~~

şeklinde 6 adet olasılık türetilir. Burada “*” işareti iki duvarın birleştiği noktayı gösterir. Bu 6 olasılık içinde sonuncusunda tüm elemanlar ilk duvarda bulunur. Bu olasılık ise zaten türetilmişti. Bunu çıkardığımızda, elimizde:

$$480 \times 5 = 2.400$$

adet, 2 duvara yayılan 5 elemanlı permütasyon bulunur. Bu olasılıkların bir çoğu (küvetin kapı önüne gelmesi durumu gibi...) kullanılamaz durumdadır. Kullanıcı kural tabanına girdiği kurallarla programın bu uygulanamaz olasılıkları elemesini sağlar.

Kullanıcı eğer 4 elemandan oluşan bir tasarım geliştirmek isterse:

CM SF LA KL
 CM SF LA DT
 CM SF LA KU
 CM SF KL DT
 CM SF KL KU
 CM TS LA KL

CM TS LA DT
 CM TS LA KU
 CM TS KL DT
 CM TS KL KU
 CM LA KL DT
 CM LA KL KU
 SF LA KL DT
 SF LA KL KU
 TS LA KL DT
 TS LA KL KU

şeklinde olmak üzere 16 adet kombinezonla işe başlar. Bunlar arasından yaptığı seçimiyle program

CM **SF LA KL**
 CM **SF KL LA**
 CM **LA SF KL**
 CM **LA KL SF**
 CM **KL LA SF**
 CM **KL SF LA ...**

şeklinde devam eden

$$4! = 4 \times 3 \times 2 \times 1 = 24$$

$$24 \times 16 = 384$$

permütasyon elde eder.

*CM SF LA KL
 CM*SF LA DT
 CM SF*LA KU
 CM SF LA*KU
~~CM SF KL DT*~~

şeklinde devam eden 2. duvarın katkısı da eklenince

$$384 \times 4 = 1.536$$

adet 4 elemanlı 2 duvarlı olasılık ortaya çıkar.

Benzer şekilde 3 elemanlı kombinezon grupları aşağıda verilmiştir:

CM SF LA
 CM SF KL
 CM SF DT
 CM SF KU
 CM TS LA
 CM TS KL
 CM TS DT
 CM TS KU
 CM LA KL
 CM LA DT
 CM LA KU
 CM KL DT
 CM KL KU
 SF LA KL
 SF LA DT
 SF LA KU
 SF KL DT
 SF KL KU
 TS LA KL
 TS LA DT
 TS LA KU
 TS KL DT
 TS KL KU
 LA KL DT
 LA KL KU

bu 25 adet kombinezonun

$$3! = 3 \times 2 \times 1 = 6$$

$$25 \times 6 = 150$$

adet tek duvarlı permütasyonu bulunur. İkinci duvarla birlikte:

*CM SF LA
 CM*SF KL
 CM SF*DT
~~CM SF DT*~~

şeklinde her permütasyona ait 3 adet yeni olasılık daha elde edilir, ve

$$150 \times 3 = 450$$

adet yeni 3 elemanlı permütasyon oluşur.

İki elemanlı kombinezonlar:

CM SF
 CM TS
 CM LA
 CM KL
 CM DT
 CM KU
 SF LA
 SF KL
 SF DT
 SF KU
 TS LA
 TS KL
 TS DT
 TS KU
 LA KL
 LA DT
 LA KU
 KL DT
 KL KU

şeklinde sıralanıp, toplam 18 adettirler. Elemanları yer değiştirerek;

$$2! = 2 \times 1 = 2$$

$$18 \times 2 = 36$$

adet yeni olasılık oluşur. İki duvar birden kullanılmaya başlandığında ise

*CM SF
 CM*TS
~~CM-TS*~~

her olasılık için iki adet yeni olasılık oluşur. Buna göre;

$$36 \times 2 = 72$$

adet 2 eleman ve 2 duvar kullanan olasılık meydana gelir.

Elde edilen rakamların toplu dökümü tablo A.1 de verilmiştir.

Tablo A.1 Kombinezon tablosu

5 elemanlı, tek duvarlı olasılıklar		480
5 elemanlı, iki duvarlı olasılıklar	$480 \times 5 =$	2.400
4 elemanlı, tek duvarlı olasılıklar		384
4 elemanlı, iki duvarlı olasılıklar	$384 \times 4 =$	1.536
3 elemanlı, tek duvarlı olasılıklar		150
3 elemanlı, iki duvarlı olasılıklar	$150 \times 3 =$	450
2 elemanlı, tek duvarlı olasılıklar		36
2 elemanlı, iki duvarlı olasılıklar	$36 \times 2 =$	72
Toplam olasılık sayısı:		5.508

Program kural tabanı tarafından sınanmak üzere 5.508 adet olasılık üretir. Fakat bu olasılıklardan ancak çok küçük bir kısmı uygulanabilme imkanına sahiptir. Bunların sayısını kullanıcının programa verdiği kurallar belirler.

EK B Programın Metni

Aşağıda verilen bilgiler Wasco Technical Software'den Steve Waskow tarafından geliştirilen ALLY (A Lisp Analyzer) programının 1992 yılında çıkan 3.0 numaralı sürümü kullanılarak üretilmiştir. 3010 satırlık program kodunda 139 adet fonksiyon, ve 812 adet değişken (tekrar kullanımlar dışında) tanımlanmış, 85 adet AutoLISP ve AutoCAD fonksiyon ve sembolü kullanılmıştır. Bu rakamların ayrıntılı değerlendirmeleri aşağıda verilmiştir:

Tablo B.1 Program Kodunun harf, sözcük, ve satır cinsinden uzunluğu.

	Harf	%	Sözcük	%	Satır	%
Kod:	49.555	73	4.439	98	2.810	93
Yorum:	1.292	2	89	2	90	3
Boşluk:	16.949	25	-	-	164	5
Toplam:	67.796		4.528		3.010	

Tablo B.2 Programda Yer Alan Sözcük ve Satır Uzunluklarının Ayrıntısı.

	Maximum	Ortlama
Harf adedi cinsinden satır uzunluğu:	76	22,52
Satır başına düşen sözcük sayısı:	7	1,58

Tablo B.3 Programda Yer Alan Fonksiyon ve Değişkenlerin Dökümü.

	Global	Local	Toplam
Kullanıcı Fonksiyonları:	139	0	139
Kullanıcı Değişkenleri:	116	696	812
AutoLISP Fonksiyonları:	85	-	85

Tablo B.4 Programda Yer Alan Fonksiyon ve Değişkenlerin, Uzunlukları ve Kullanım Miktarları.

	Ortalama Kullanım Sayısı	Ortalama Harf Uzunluğu
Kullanıcı Fonksiyonları:	3,61	7,06
Kullanıcı Değişkenleri:	4,55	2,50
AutoLISP Fonksiyonları:	47,71	-

Tanımlanan fonksiyonlar ve satır numaraları şöyledir:

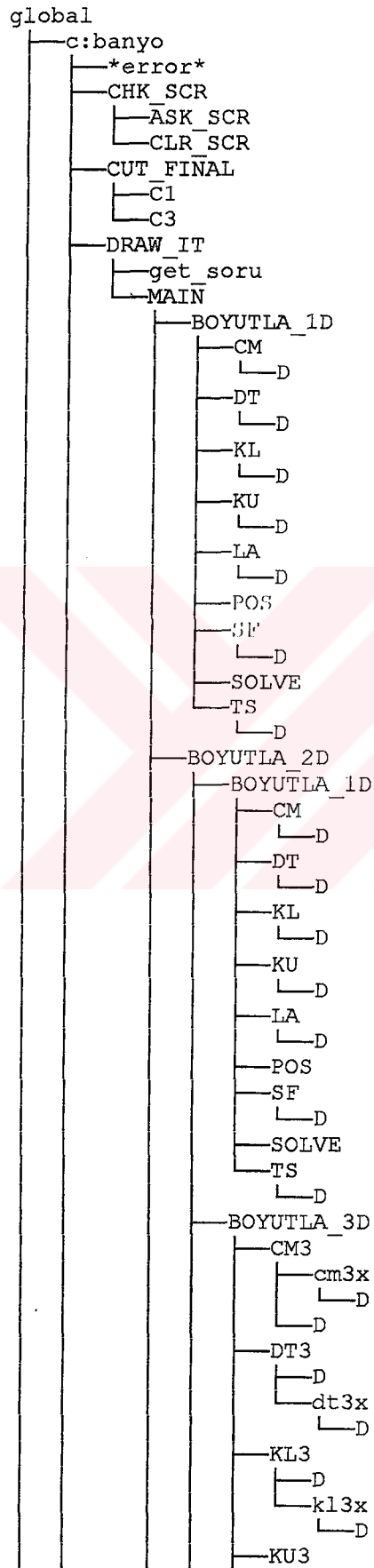
—tp_err (6-12)
 —GET_DETAIL (14-55)
 —FIX_IT (57-62)
 —GET_RULE (64-123)
 —SOLVE (125-132)
 —EXAM1 (134-155)
 —EXAM_1D (157-193)
 —EX5 (195-200)
 —EX6 (202-218)
 —EX8 (220-238)
 —EX9 (240-256)
 —EX11 (258-285)
 —EX12 (287-303)
 —EX14 (305-342)
 —EX15 (344-360)
 —EXAM2 (362-376)
 —EXAM_2D (378-421)
 —EXAM (423-432)
 —DEL_POPUP (434-440)
 —EKSILT (442-452)
 —PER_2 (454-489)
 —PER_3 (491-548)
 —PER_4 (550-618)
 —PER_5 (620-697)
 —DO_PERM (699-707)
 —D (709-709)
 —LA (711-736)
 —KL (738-763)
 —CM (765-790)
 —DT (792-808)
 —KU (810-817)
 —SF (819-844)
 —TS (846-871)
 —POS (873-881)
 —BOYUTLA_1D (883-906)
 —la3x (908-933)
 —LA3 (935-961)
 —kl3x (963-988)
 —KL3 (990-1016)
 —cm3x (1018-1028)
 —CM3 (1030-1057)
 —dt3x (1059-1069)
 —DT3 (1071-1088)
 —KU3 (1090-1103)
 —sf3x (1105-1123)
 —SF3 (1125-1151)
 —ts3x (1153-1164)
 —TS3 (1166-1192)
 —BOYUTLA_3D (1194-1216)
 —BOYUTLA_2D (1218-1236)
 —solve_2d (1238-1260)
 —TEST_1 (1262-1286)
 —TEST (1288-1299)
 —GET_FIRMA (1301-1354)
 —GET_ALL (1356-1417)
 —chk_TP_DIALOG3 (1419-1422)
 —TP_DIALOG3 (1424-1487)
 —UPDATE_X (1489-1496)
 —set_vitr (1498-1502)
 —set_ykap (1504-1512)
 —set_dkap (1514-1522)

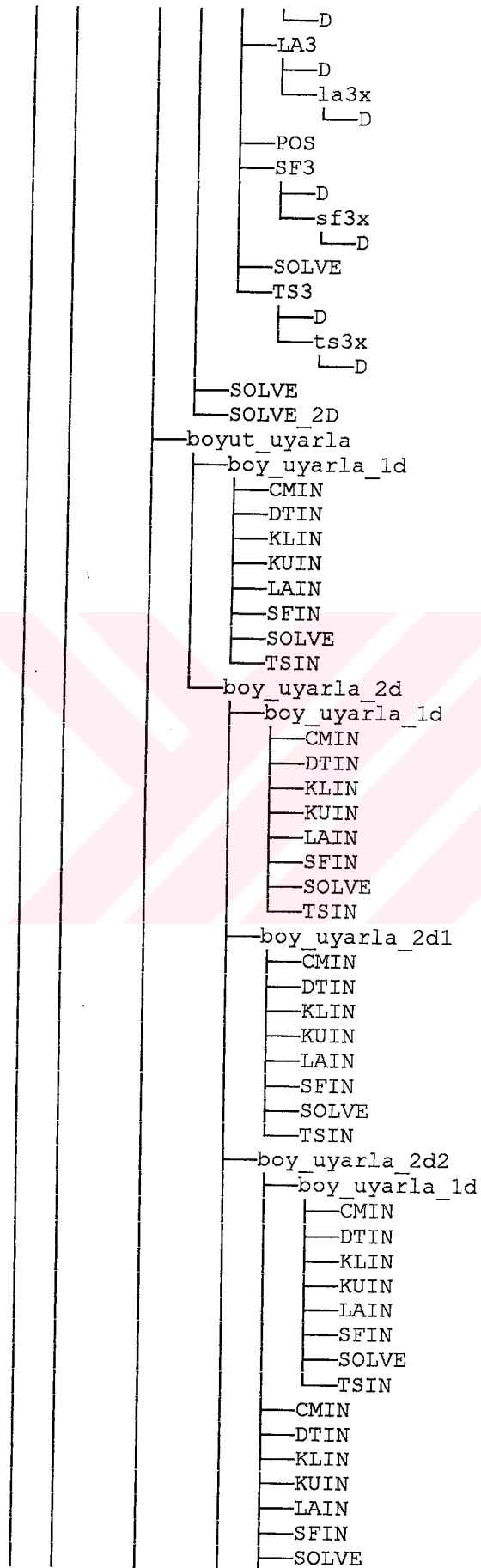
—UPDATE_TP2 (1524-1561)
 —TP_DIALOG2 (1563-1627)
 —SET_THEM_FIRST (1629-1660)
 —PUT_LAST (1662-1692)
 —UPDATE_TP1 (1694-1733)
 —CNG_TOL (1735-1742)
 —CNG_CZS (1744-1750)
 —CNG_CSLD (1752-1758)
 —set_label (1760-1780)
 —TP_DIALOG1 (1782-1828)
 —pre_exam (1830-1841)
 —cz_1 (1843-1847)
 —cz_2 (1849-1853)
 —cz_3 (1855-1860)
 —DOSEME1 (1862-1887)
 —DOSEME2 (1889-1914)
 —DOSEME (1916-1921)
 —LAIN (1923-1928)
 —KLIN (1930-1935)
 —CMIN (1937-1942)
 —DTIN (1944-1949)
 —KUIN (1951-1957)
 —SFIN (1959-1964)
 —TSIN (1966-1971)
 —boy_uyarla_1d (1973-1987)
 —boy_uyarla_2d1 (1989-2015)
 —boy_uyarla_2d2 (2017-2044)
 —boy_uyarla_2d (2046-2059)
 —boyut_uyarla (2061-2072)
 —golgeler (2074-2087)
 —gercekler (2089-2102)
 —set_some_globals (2104-2128)
 —duvarlar (2130-2160)
 —kapi (2162-2191)
 —pencere (2193-2225)
 —son_kontrol (2227-2240)
 —MAIN (2242-2284)
 —olculer_yazilar (2286-2319)
 —noktali (2321-2349)
 —GET_MLYT (2351-2386)
 —get_soru (2388-2395)
 —DRAW_IT (2397-2422)
 —CLR_SCR (2424-2428)
 —ASK_SCR (2430-2439)
 —CHK_SCR (2441-2449)
 —C1 (2451-2457)
 —C3 (2459-2469)
 —CUT_FINAL (2471-2476)
 —ekran_6 (2478-2489)
 —mode_tile_7 (2491-2511)
 —goto_first_line (2513-2519)
 —fill_list_box (2521-2537)
 —do_it (2539-2552)
 —write_rule (2554-2568)
 —change_mode (2570-2578)
 —delete_it (2580-2594)
 —get_p (2596-2619)
 —get_part1 (2621-2628)
 —get_part2 (2630-2639)
 —get_part123 (2641-2648)
 —get_part4 (2650-2659)
 —get_part5 (2661-2668)
 —get_part345 (2670-2677)

- sort_1d_rule (2679-2735)
- get_last (2737-2749)
- sort_2d_rule (2751-2800)
- get_num (2802-2809)
- ask_it (2811-2821)
- delete_rule (2823-2829)
- create_backup (2831-2847)
- add_rule_1d (2849-2869)
- change_rule (2871-2890)
- add_rule_1d1 (2892-2911)
- add_rule_2d (2913-2929)
- add_rule (2931-2936)
- update_krl (2938-2954)
- rule_base (2956-2976)
- c:banyo (2978-3010)



Programda yer alan fonksiyonların çağırılış ağacı aşağıdaki gibidir:

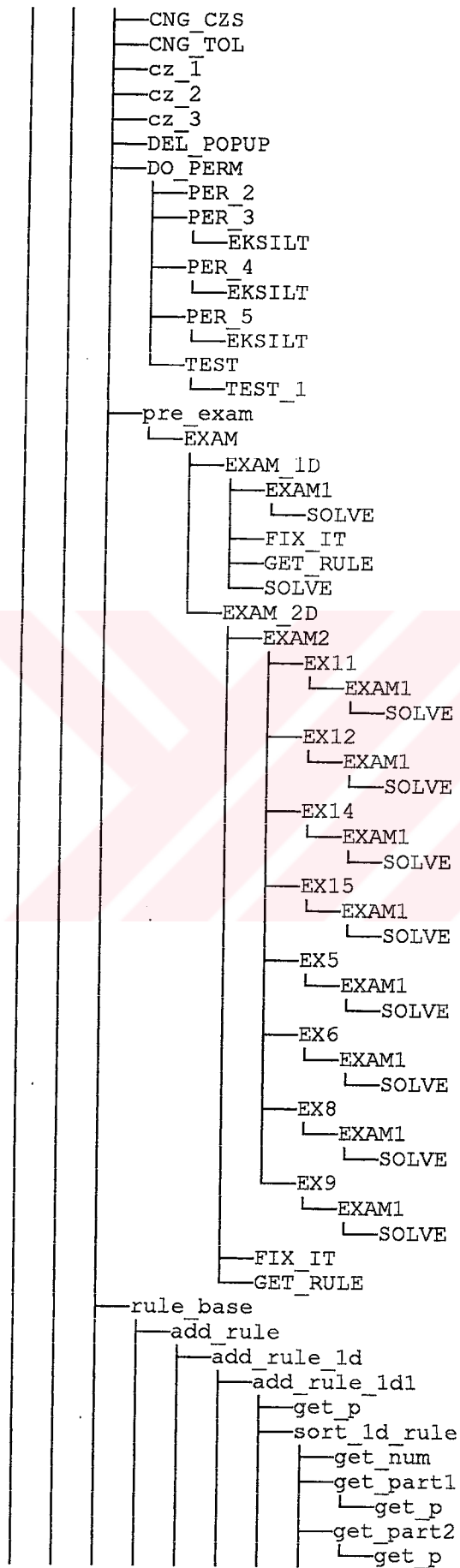


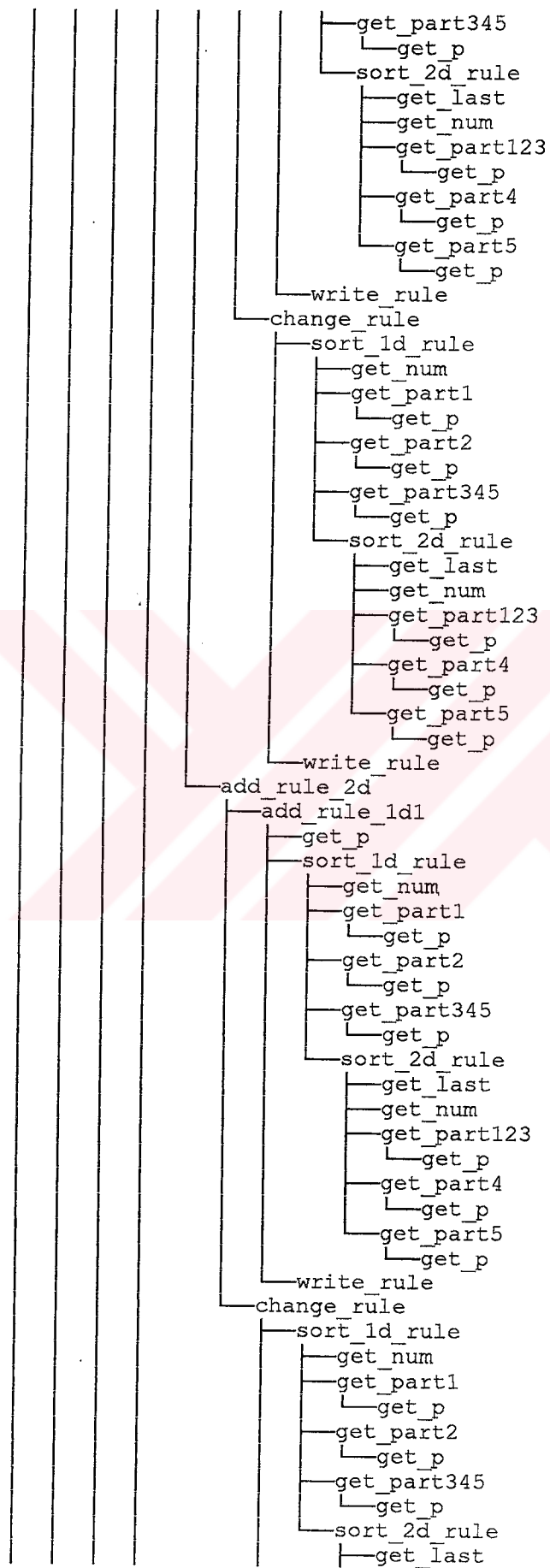


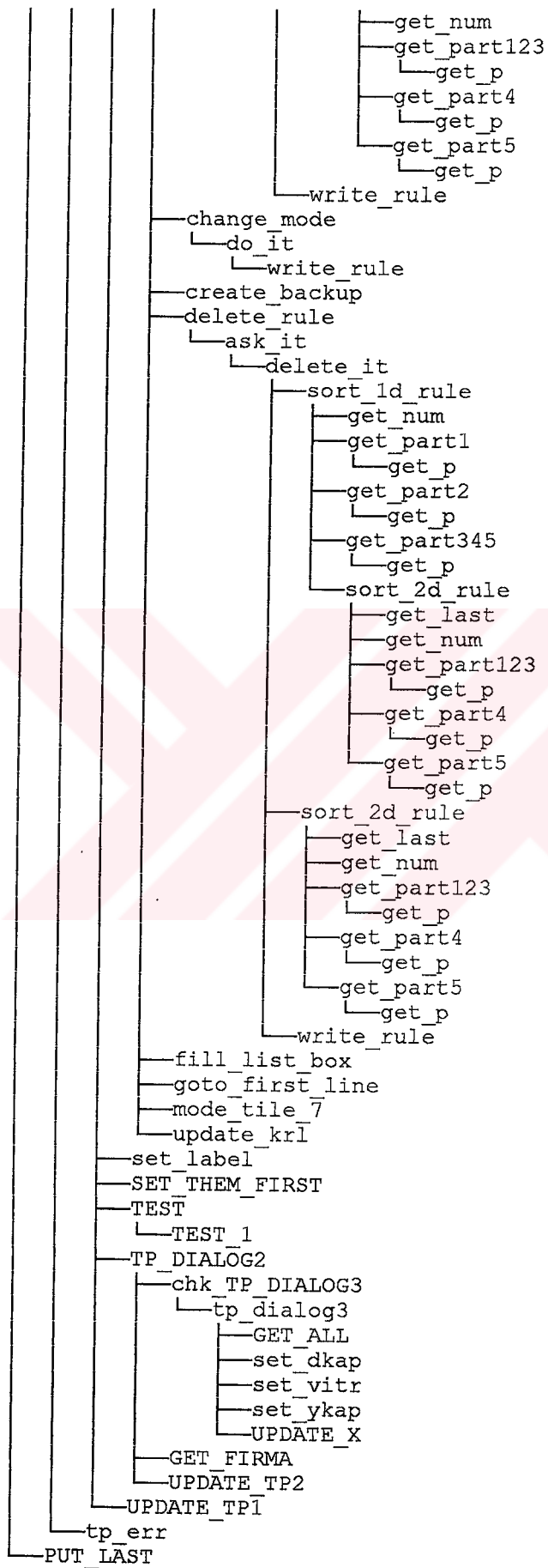
```

      └─TSIN
        └─SOLVE_2D
DOSEME
├─DOSEME1
├─DOSEME2
duvarlar
gercekler
golgeler
kapi
├─SOLVE
├─SOLVE_2D
olculer_yazilar
├─GET_MLYT
│   ├──D
│   ├──SOLVE
│   └─SOLVE_2D
noktali
pencere
set_some_globals
son_kontrol
    ├──SOLVE
    └─SOLVE_2D
ekran_6
GET_DETAIL
GET_FIRMA
pre_exam
├─EXAM
│   ├──EXAM_1D
│   │   ├──EXAM1
│   │   │   └─SOLVE
│   │   ├──FIX_IT
│   │   ├──GET_RULE
│   │   └─SOLVE
│   └─EXAM_2D
│       ├──EXAM2
│       │   ├──EX11
│       │   │   └─EXAM1
│       │   │       └─SOLVE
│       │   ├──EX12
│       │   │   └─EXAM1
│       │   │       └─SOLVE
│       │   ├──EX14
│       │   │   └─EXAM1
│       │   │       └─SOLVE
│       │   ├──EX15
│       │   │   └─EXAM1
│       │   │       └─SOLVE
│       │   ├──EX5
│       │   │   └─EXAM1
│       │   │       └─SOLVE
│       │   ├──EX6
│       │   │   └─EXAM1
│       │   │       └─SOLVE
│       │   ├──EX8
│       │   │   └─EXAM1
│       │   │       └─SOLVE
│       │   └─EX9
│       │       └─EXAM1
│       │           └─SOLVE
│       ├──FIX_IT
│       └─GET_RULE
TP_DIALOG1
└─CNG CSLD

```







Programın metni aşağıda verilmiştir:

```

1  ;;; 29.04.1995 Celik ingin
2  ;;;
3  ;;; Banyo.lsp
4  ;;;
5
6  (defun tp_err (s)
7    (if(/= s "Function cancelled") (princ(strcat "\nError:" s)))
8    (if tp_oe (setq *error* tp_oe))
9    (setvar "CMDECHO" 1)
10   (setvar "BLIPMODE" 1)
11   (princ)
12 )
13
14 (defun GET_DETAIL(/flo LINE A A0 A1 A2 A3 A4 A5 A6 A7 A8 A9)
15   (if(null(setq flo(open(strcat(getvar"dwgprefix")
16                                 "Banyo.dta")"r"))))
17     (progn
18       (princ "\n\"Banyo.dta\" dosyasi acilamadi!")
19       (exit)
20     )
21   (setq DETAY nil)
22   (while (setq LINE (read-line flo))
23     (if (= (substr LINE 1 1) "(")
24       (progn
25         (setq L (eval (read LINE)))
26         (setq A (nth 0 L))
27         (setq B (nth 1 L))
28         (setq C (nth 2 L))
29         (cond
30           ((and (= A SVT) (= B "LA")) (setq A1 L))
31           ((and (= A SVT) (= B "KL")) (setq A2 L))
32           ((= B "CM") (setq A3 L))
33           ((= B "DT") (setq A4 L))
34           ((= B "KU") (setq A5 L))
35           ((= B "SF") (setq A6 L))
36           ((= B "TS") (setq A7 L))
37           ((and (= A SKP) (= B "YK") (= C SKP1)) (setq A8 L))
38           ((and (= A SDK) (= B "DK") (= C SDK1)) (setq A9 L))
39           ((= A "Ekler") (setq A0 L))
40         )
41       )
42     )
43   )
44   (setq DETAY (cons A0 DETAY))
45   (setq DETAY (cons A9 DETAY))
46   (setq DETAY (cons A8 DETAY))
47   (setq DETAY (cons A7 DETAY))
48   (setq DETAY (cons A6 DETAY))
49   (setq DETAY (cons A5 DETAY))
50   (setq DETAY (cons A4 DETAY))
51   (setq DETAY (cons A3 DETAY))
52   (setq DETAY (cons A2 DETAY))
53   (setq DETAY (cons A1 DETAY))
54   (close flo)
55 )
56
57 (defun FIX_IT ( X / )
58   (if (= (strlen (itoa X)) 1) (setq X (strcat "0" (itoa X)))
59       (setq X (itoa X))
60   )

```

```

61 (eval X)
62 )
63
64 (defun GET_RULE ( LEN / R A B C K L X X2 Y Y2 Z FL LINE)
65 (if(null(setq FL(open(strcat(getvar
                                "dwgprefix")"Banyo.krl")"r"))))
66 (progn
67 (princ "\n\"Banyo.krl\" dosyasi açilamadi!")
68 (exit)
69 )
70 )
71 (setq R "")
72 (while (setq LINE (read-line FL))
73 (if (= (substr LINE 1 1) "(")
74 (setq R (strcat R LINE " "))
75 )
76 )
77 (close FL)
78
79 (if(null(setq FL(open(strcat(getvar
                                "dwgprefix")"Banyo.krl")"r"))))
80 (progn
81 (princ "\n\"Banyo.krl\" dosyasi açilamadi!")
82 (exit)
83 )
84 )
85 (while (setq LINE (read-line FL))
86 (cond
87 ;; tek duvar için kurallari alir
88 ( (and (= (substr LINE 11 1) (itoa LEN))
89 (= (substr LINE 1 1) "+")
90 )
91 (progn
92 (setq X (substr LINE 14 2))
93 (setq Y (substr LINE 17 2))
94 (setq K (substr LINE 20 2))
95 (setq L (substr LINE 23 1))
96 (setq Z (substr LINE 28 1))
97 (setq A (strcat "(if (and (= \" X \" \" Y \"\") )"))
98 (setq B (strcat "(= \" K \" \" L \")) \""))
99 (setq C (strcat "(setq A (+ A \" Z \")) )\" ))
100 (setq R (strcat R A B C))
101 )
102 )
103 ;; iki duvar için kurallari alir
104 ( (and (= (substr LINE 8 1) "2") (= LEN 6)
105 (= (substr LINE 1 1) "+")
106 )
107 (progn
108 (setq X (substr LINE 11 2))
109 (setq Y (substr LINE 14 2))
110 (setq X2 (substr LINE 20 2))
111 (setq Y2 (substr LINE 23 2))
112 (setq Z (substr LINE 29 1))
113 (setq A (strcat "(if (and (= \" X \" \" Y \"\") )"))
114 (setq B (strcat "(= \" X2 \" \" Y2 \"\") \""))
115 (setq C (strcat "(setq A (+ A \" Z \")) )\" ))
116 (setq R (strcat R A B C))
117 )
118 )
119 )
120 )

```

```

121 (close FL)
122 (eval (strcat "(progn " R ")") )
123 )
124
125 (defun SOLVE ( STR / A X X1 X2 X3 X4 X5 )
126   (setq A nil)
127   (if(/=(substr STR 13 2)"" ) (setq A (cons (substr STR 13 2) A)))
128   (if(/=(substr STR 10 2)"" ) (setq A (cons (substr STR 10 2) A)))
129   (if(/=(substr STR 7 2)"" ) (setq A (cons (substr STR 7 2) A)))
130   (if(/=(substr STR 4 2)"" ) (setq A (cons (substr STR 4 2) A)))
131   (if(/=(substr STR 1 2)"" ) (setq A (cons (substr STR 1 2) A)))
132 )
133
134 (defun EXAM1 (RULES STR D1 D2 DV AD / LST A A1 A2 A3 A4 A5 P2
135              P3 P4 P5 RULES STR D1 D2 DV AD Ax Ay)
136   (setq LST (SOLVE STR))
137   (setq A1 (nth 0 LST))
138   A2 (nth 1 LST)
139   A3 (nth 2 LST)
140   A4 (nth 3 LST)
141   A5 (nth 4 LST)
142   A 0
143   Ax ""
144   Ay ""
145   P2 (= 2 (length LST))
146   P3 (= 3 (length LST))
147   P4 (= 4 (length LST))
148   P5 (= 5 (length LST))
149 )
150 ;;kose noktasi kontrolu icin kose elemanlari hazirlar...
151 (if (= ad 0) (setq Ax (car lst) Ay (cadr lst)))
152
153 (eval (read RULES))
154 (eval A)
155 )
156
157 (defun EXAM_1D ( / P RULEp LEN M X Y Z TL PERMo)
158   (setq LEN (length (SOLVE (nth 0 PERM))))
159   (setq RULEp (GET_RULE LEN))
160   (setq P nil)
161   (setq Z 0)
162   (setq M 0)
163   (foreach Y PERM
164     (setq X (EXAM1 RULEp Y LEN 0 1 1))
165     (setq P (cons (FIX_IT X) P) )
166     (set_tile "DC4_PR" Y)
167     (set_tile "DC4_PP" (FIX_IT X))
168     (setq Z (+ 1 Z))
169     (set_tile "DC4_PS" (itoa Z))
170     (if (<= X (atoi TOL))
171       (progn
172         (setq M (+ 1 M))
173         (setq TL (strcat (itoa M) "      =      "
174           "Tolerans <= " TOL ))
175         (set_tile "DC4_TL" TL)
176       )
177     )
178   )
179   (setq P (reverse P))
180   (setq PERMo nil)
181   (setq Z 0)
182   (foreach Y PERM
183     (setq PERMo (cons (strcat (nth Z P) Y) PERMo))

```

```

184      (setq Z (+ 1 Z))
185    )
186    (setq PERMo (acad_strlsort PERMo))
187    (setq FINAL nil)
188    (foreach Y PERMo
189      (setq FINAL (cons (list (atoi (substr Y 1 2))
190        (substr Y 3 15) ) FINAL))
191    )
192    (setq FINAL (reverse FINAL))
193  )
194
195  (defun EX5 ( S R1 / S1 R1 A )
196    ;; "LA*KL" ...
197    (setq S1 (strcat (substr S 1 2) " " (substr S 4 2)) )
198    (setq A (EXAM1 R1 S1 1 1 2 0))
199    (eval A)
200  )
201
202  (defun EX6 ( S R2 / S1 R2 A )
203    (cond
204      ;; "*LA KL"...
205      ((= (substr S 1 1) "*") (progn
206        (setq S1 (substr S 2 5))
207        (setq A (EXAM1 R2 S1 0 2 2 2))
208      )
209    )
210    ;; "LA KL*"...
211    ((= (substr S 6 1) "*") (progn
212      (setq S1 (substr S 1 5))
213      (setq A (EXAM1 R2 S1 2 0 2 1))
214    )
215  )
216  )
217  (eval A)
218  )
219
220  (defun EX8 ( S R1 R2 / S S1 S2 R1 R2 A )
221    (cond
222      ;; "LA KL*CM"...
223      ((= (substr S 6 1) "*") (progn
224        (setq S1 (substr S 1 5))
225        (setq S2 (strcat (substr S 4 2) " " (substr S 7 2)))
226        (setq A (EXAM1 R2 S1 2 1 2 1))
227        (setq A (+ A (EXAM1 R1 S2 2 1 2 0)))
228      )
229      ;; "LA*KL CM"...
230      ((= (substr S 3 1) "*") (progn
231        (setq S1 (substr S 4 5))
232        (setq S2 (strcat (substr S 1 2) " " (substr S 4 2)))
233        (setq A (EXAM1 R2 S1 1 2 2 2))
234        (setq A (+ A (EXAM1 R1 S2 1 2 2 0)))
235      )
236    )
237    (eval A)
238  )
239
240  (defun EX9 ( S R3 / S S1 R3 A )
241    (cond
242      ;; "LA KL CM*"...
243      ((= (substr S 9 1) "*") (progn
244        (setq S1 (substr S 1 8))
245        (setq A (EXAM1 R3 S1 3 0 2 1))
246      )

```

```

247     )
248 ;; "LA KL CM"...
249     ((= (substr S 1 1) "") (progn
250         (setq S1 (substr S 2 8))
251         (setq A (EXAM1 R3 S1 3 0 2 2))
252     )
253 )
254 )
255 (eval A)
256 )
257
258 (defun EX11 ( S R1 R2 R3 / S S1 S2 S3 R1 R2 R3 A )
259 (cond
260 ;; "LA KL CM*DT"...
261     ((= (substr S 9 1) "") (progn
262         (setq S1 (substr S 1 8))
263         (setq S2 (strcat (substr S 7 2) " " (substr S 10 2)))
264         (setq A (EXAM1 R3 S1 3 1 2 1))
265         (setq A (+ A (EXAM1 R1 S2 3 1 2 0)))
266     ))
267 ;; "LA KL*CM DT"...
268     ((= (substr S 6 1) "") (progn
269         (setq S1 (substr S 1 5))
270         (setq S2 (substr S 7 5))
271         (setq S3 (strcat (substr S 4 2) " " (substr S 7 2)))
272         (setq A (EXAM1 R2 S1 2 2 2 1))
273         (setq A (+ A (EXAM1 R2 S2 2 2 2 2)))
274         (setq A (+ A (EXAM1 R1 S3 2 2 2 0)))
275     ))
276 ;; "LA*KL CM DT"...
277     ((= (substr S 3 1) "") (progn
278         (setq S1 (substr S 4 8))
279         (setq S2 (strcat (substr S 1 2) " " (substr S 4 2)))
280         (setq A (EXAM1 R3 S1 1 3 2 2))
281         (setq A (+ A (EXAM1 R1 S2 1 3 2 0)))
282     ))
283 )
284 (eval A)
285 )
286
287 (defun EX12 ( S R4 / S S1 R4 A )
288 (cond
289 ;; "LA KL CM DT*"...
290     ((= (substr S 12 1) "") (progn
291         (setq S1 (substr S 1 11))
292         (setq A (EXAM1 R4 S1 4 0 2 1))
293     )
294 )
295 ;; "LA KL CM DT"...
296     ((= (substr S 1 1) "") (progn
297         (setq S1 (substr S 2 11))
298         (setq A (EXAM1 R4 S1 0 4 2 2))
299     )
300 )
301 )
302 (eval A)
303 )
304
305 (defun EX14 ( S R1 R2 R3 R4 / S S1 S2 S3 R1 R2 R3 R4 A )
306 (setq S3 nil)
307 (cond
308 ;; "LA KL CM DT*SF"...
309     ((= (substr S 12 1) "") (progn

```

```

310      (setq S1 (substr S 1 11))
311      (setq S2 (strcat (substr S 10 2) " " (substr S 13 2)))
312      (setq A (EXAM1 R4 S1 4 1 2 1))
313      (setq A (+ A (EXAM1 R1 S2 4 1 2 0)))
314    ))
315  ;; "LA KL CM*DT SF"...
316  ((= (substr S 9 1) "") (progn
317    (setq S1 (substr S 1 8))
318    (setq S2 (substr S 10 5))
319    (setq S3 (strcat (substr S 7 2) " " (substr S 10 2)))
320    (setq A (EXAM1 R3 S1 3 2 2 1))
321    (setq A (+ A (EXAM1 R2 S2 3 2 2 2)))
322    (setq A (+ A (EXAM1 R1 S3 3 2 2 0)))
323  ))
324  ;; "LA KL*CM DT SF"...
325  ((= (substr S 6 1) "") (progn
326    (setq S1 (substr S 1 5))
327    (setq S2 (substr S 7 8))
328    (setq S3 (strcat (substr S 4 2) " " (substr S 7 2)))
329    (setq A (EXAM1 R2 S1 2 3 2 1))
330    (setq A (+ A (EXAM1 R3 S2 2 3 2 2)))
331    (setq A (+ A (EXAM1 R1 S3 2 3 2 0)))
332  ))
333  ;; "LA*KL CM DT SF"...
334  ((= (substr S 3 1) "") (progn
335    (setq S1 (substr S 4 8))
336    (setq S2 (strcat (substr S 1 2) " " (substr S 4 2)))
337    (setq A (EXAM1 R4 S1 1 4 2 2))
338    (setq A (+ A (EXAM1 R1 S2 1 4 2 0)))
339  ))
340  )
341  (eval A)
342  )
343
344  (defun EX15 ( S R5 / S S1 R5 A )
345    (cond
346      ;; "LA KL CM DT SF*"...
347      ((= (substr S 15 1) "") (progn
348        (setq S1 (substr S 1 14))
349        (setq A (EXAM1 R5 S1 5 0 2 1))
350      ))
351    )
352    ;; "*LA KL CM DT SF"...
353    ((= (substr S 1 1) "") (progn
354      (setq S1 (substr S 2 14))
355      (setq A (EXAM1 R5 S1 0 5 2 2))
356    ))
357  )
358  )
359  (eval A)
360  )
361
362  (defun EXAM2 ( STR R1 R2 R3 R4 R5 / STR R1 R2 R3 R4 R5 X A )
363    (setq X (strlen STR))
364    (setq A 0)
365    (cond
366      ((= X 5) (setq A (EX5 STR R1)) )
367      ((= X 6) (setq A (EX6 STR R2)) )
368      ((= X 8) (setq A (EX8 STR R1 R2)) )
369      ((= X 9) (setq A (EX9 STR R3)) )
370      ((= X 11) (setq A (EX11 STR R1 R2 R3)) )
371      ((= X 12) (setq A (EX12 STR R4)) )
372      ((= X 14) (setq A (EX14 STR R1 R2 R3 R4)) )

```

```

373      ((= X 15) (setq A (EX15 STR R5)) )
374    )
375    (eval A)
376  )
377
378  (defun EXAM_2D(/R1 R2 R3 R4 R5 P RULEp LEN M X Y Z TL PERMo)
379    (setq R1 (GET_RULE 6))
380    (setq R2 (GET_RULE 2))
381    (setq R3 (GET_RULE 3))
382    (setq R4 (GET_RULE 4))
383    (setq R5 (GET_RULE 5))
384    ; (setq Rg (GET_RULE 7))
385    (setq P nil)
386    (setq Z 0)
387    (setq M 0)
388    (foreach Y PERM
389      (setq X (EXAM2 Y R1 R2 R3 R4 R5))
390      (setq P (cons (FIX_IT X) P) )
391      (set_tile "DC4_PR" Y)
392      (set_tile "DC4_PP" (FIX_IT X))
393      (setq Z (+ 1 Z))
394      (set_tile "DC4_PS" (itoa Z))
395      (if (<= X (atoi TOL))
396        (progn
397          (setq M (+ 1 M))
398          (setq TL (strcat (itoa M) "      =      ")
399            "Tolerans <= " TOL ))
400          (set_tile "DC4_TL" TL)
401        )
402      )
403    )
404    (setq P (reverse P))
405    (setq PERMo nil)
406    (setq Z 0)
407    (setq X (strlen (nth 0 PERM)))
408    (foreach Y PERM
409      (setq PERMo (cons (strcat (nth Z P) Y)PERMo))
410      (setq Z (+ 1 Z))
411    )
412    (setq PERMo (acad_strlsort (reverse PERMo)))
413    (setq FINAL nil)
414    (foreach Y PERMo
415      (setq FINAL(cons(list(atoi(substr Y 1 2))(substr Y 3 15))
416        FINAL
417      )
418    )
419    )
420    (setq FINAL (reverse FINAL))
421  )
422
423  (defun EXAM ()
424    (setq DCL_ID ( load_dialog "Banyo.DCL"))
425    (if ( not ( new_dialog "DDBanyo4" DCL_ID )) ( exit ) )
426
427    (if (= DV "1") (EXAM_1D) (EXAM_2D) )
428
429    (action_tile "DC4_IP" "(done_dialog)" )
430    (start_dialog)
431    (unload_dialog DCL_ID)
432  )
433
434  ( defun DEL_POPUP ()
435    (start_list "DC_POP")

```

```

436 (add_list "")
437 (end_list)
438 (mode_tile "DC_CZ" 1)
439 (mode_tile "DC_DS" 1)
440 )
441
442 (defun EKSILT ( EKS KOMx )
443 (setq ZZ2 0)
444 (setq KOMy nil)
445 (while (<= ZZ2 (- (length KOMx) 1))
446 (if (/= EKS (nth ZZ2 KOMx))
447 (setq KOMy (cons (nth ZZ2 KOMx) KOMy))
448 )
449 (setq ZZ2 (+ ZZ2 1))
450 )
451 (setq KOMy (reverse KOMy))
452 )
453
454 (defun PER_2 ( / PERMo P P1 P2 )
455 (setq KOMB2 KOMB)
456 (setq PERM nil)
457 (setq PERM(cons(strcat(nth 0 KOMB2)" "(nth 1 KOMB2))PERM))
458 (setq PERM(cons(strcat(nth 1 KOMB2)" "(nth 0 KOMB2))PERM))
459 (if (/= (get_tile "DC_2D") "1")
460 (progn
461 (start_list "DC_POP")
462 (mapcar 'add_list PERM)
463 (end_list)
464 (set_tile "DC_PER" "2")
465 ; (set_tile "DC_PER" "2-5578")
466 )
467 (progn
468 (setq PERMo nil)
469 (setq P 0)
470 (while (<= P 1)
471 (setq P1 (strcat "*" (nth P PERM)))
472 (setq P2 (strcat (substr (nth P PERM) 1 2 )
473 "*" (substr (nth P PERM) 4 5 ) ))
474 (setq P3 (strcat (nth P PERM) "*" ))
475 (setq PERMo (cons P1 PERMo))
476 (setq PERMo (cons P2 PERMo))
477 (setq PERMo (cons P3 PERMo))
478 (setq P (+ P 1))
479 )
480 (setq PERM PERMo)
481 (start_list "DC_POP")
482 (mapcar 'add_list PERM)
483 (end_list)
484 (set_tile "DC_PER" "6")
485 )
486 )
487 (mode_tile "DC_CZ" 0)
488 (mode_tile "DC_DS" 0)
489 )
490
491 (defun PER_3 ( / C COUNT P P1 P2 P3 P4 Z1 Z2 Z3 )
492 (set_tile "DC_PER" "0")
493 (setq Z2 0)
494 (setq Z1 0)
495 (setq P "")
496 (setq COUNT 0)
497 (setq KOMB3 KOMB)
498 (setq PERM nil)

```

```

499 (while (<= Z1 2)
500   (setq P1 (nth Z1 KOMB3))
501   (setq KOMB2 nil)
502   (setq KOMB2 (EKSILT (nth Z1 KOMB3) KOMB3))
503   (while (<= Z2 1)
504     (setq P2 (nth Z2 KOMB2))
505     (setq P3 (nth 0 (EKSILT (nth Z2 KOMB2) KOMB2)))
506     (setq P (strcat P1 " " P2 " " P3))
507     (setq PERM (cons P PERM))
508     (setq Z2 (+ Z2 1))
509     (setq COUNT (+ COUNT 1))
510   )
511   (setq Z2 0)
512   (setq Z1 (+ Z1 1))
513 )
514 (if (/= (get_tile "DC_2D") "1")
515   (progn
516     (setq PERM (reverse PERM))
517     (start_list "DC_POP")
518     (mapcar 'add_list PERM)
519     (end_list)
520   )
521   (progn
522     (setq PERMo nil)
523     (setq P 0)
524     (while (<= P (- COUNT 1))
525       (setq P1 (strcat "*" (nth P PERM)))
526       (setq P2 (strcat (substr (nth P PERM) 1 2 )
527         "*" (substr (nth P PERM) 4 8 ) ))
528       (setq P3 (strcat (substr (nth P PERM) 1 5 )
529         "*" (substr (nth P PERM) 7 8 ) ))
530       (setq P4 (strcat (nth P PERM) "*" ))
531       (setq PERMo (cons P1 PERMo))
532       (setq PERMo (cons P2 PERMo))
533       (setq PERMo (cons P3 PERMo))
534       (setq PERMo (cons P4 PERMo))
535       (setq P (+ P 1))
536     )
537     (setq PERM PERMo)
538     (start_list "DC_POP")
539     (mapcar 'add_list PERM)
540     (end_list)
541     (setq COUNT (* COUNT 4))
542   )
543 )
544 (setq C (itoa COUNT))
545 (set_tile "DC_PER" C)
546 (mode_tile "DC_CZ" 0)
547 (mode_tile "DC_DS" 0)
548 )
549
550 (defun PER_4 ( / C COUNT P P1 P2 P3 P4 Z1 Z2 Z3 )
551   (set_tile "DC_PER" "00")
552   (setq Z3 0)
553   (setq Z2 0)
554   (setq Z1 0)
555   (setq P "")
556   (setq COUNT 0)
557   (setq KOMB4 KOMB)
558   (setq PERM nil)
559   (while (<= Z1 3)
560     (setq P1 (nth Z1 KOMB4))
561     (setq KOMB3 nil)

```

```

562      (setq KOMB3 (EKSILT (nth Z1 KOMB4) KOMB4))
563      (while (<= Z2 2)
564        (setq P2 (nth Z2 KOMB3))
565        (setq KOMB2 nil)
566        (setq KOMB2 (EKSILT (nth Z2 KOMB3) KOMB3))
567        (while (<= Z3 1)
568          (setq P3 (nth Z3 KOMB2))
569          (setq P4 (nth 0 (EKSILT (nth Z3 KOMB2) KOMB2)))
570          (setq P (strcat P1 " " P2 " " P3 " " P4 ))
571          (setq PERM (cons P PERM))
572          (setq Z3 (+ Z3 1))
573          (setq COUNT (+ COUNT 1))
574        )
575        (setq Z3 0)
576        (setq Z2 (+ Z2 1))
577      )
578      (setq Z2 0)
579      (setq Z1 (+ Z1 1))
580    )
581    (if (/= (get_tile "DC_2D") "1")
582      (progn
583        (setq PERM (reverse PERM))
584        (start_list "DC_POP")
585        (mapcar 'add_list PERM)
586        (end_list)
587      )
588      (progn
589        (setq PERMo nil)
590        (setq P 0)
591        (while (<= P (- COUNT 1))
592          (setq P1 (strcat "*" (nth P PERM)))
593          (setq P2 (strcat (substr (nth P PERM) 1 2 )
594            "*" (substr (nth P PERM) 4 11 ) ))
595          (setq P3 (strcat (substr (nth P PERM) 1 5 )
596            "*" (substr (nth P PERM) 7 11 ) ))
597          (setq P4 (strcat (substr (nth P PERM) 1 8 )
598            "*" (substr (nth P PERM) 10 11 ) ))
599          (setq P5 (strcat (nth P PERM) "*" ))
600          (setq PERMo (cons P1 PERMo))
601          (setq PERMo (cons P2 PERMo))
602          (setq PERMo (cons P3 PERMo))
603          (setq PERMo (cons P4 PERMo))
604          (setq PERMo (cons P5 PERMo))
605          (setq P (+ P 1))
606        )
607        (setq PERM PERMo)
608        (start_list "DC_POP")
609        (mapcar 'add_list PERM)
610        (end_list)
611        (setq COUNT (* COUNT 5))
612      )
613    )
614    (setq C (itoa COUNT))
615    (set_tile "DC_PER" C)
616    (mode_tile "DC_CZ" 0)
617    (mode_tile "DC_DS" 0)
618  )
619
620  (defun PER_5 ( / C COUNT P P0 P1 P2 P3 P4 Z1 Z2 Z3 Z4 )
621    (set_tile "DC_PER" "000")
622    (setq Z3 0)
623    (setq Z2 0)
624    (setq Z1 0)

```

```

625 (setq Z0 0)
626 (setq P "")
627 (setq COUNT 0)
628 (setq KOMB5 KOMB)
629 (setq PERM nil)
630 (while (<= Z0 4)
631   (setq P0 (nth Z0 KOMB5))
632   (setq KOMB4 (EKSILT (nth Z0 KOMB5) KOMB5))
633   (while (<= Z1 3)
634     (setq P1 (nth Z1 KOMB4))
635     (setq KOMB3 (EKSILT (nth Z1 KOMB4) KOMB4))
636     (while (<= Z2 2)
637       (setq P2 (nth Z2 KOMB3))
638       (setq KOMB2 (EKSILT (nth Z2 KOMB3) KOMB3))
639       (while (<= Z3 1)
640         (setq P3 (nth Z3 KOMB2))
641         (setq P4 (nth 0 (EKSILT (nth Z3 KOMB2) KOMB2)))
642         (setq P (strcat P0 " " P1 " " P2 " " P3 " " P4 ))
643         (setq PERM (cons P PERM))
644         (setq Z3 (+ Z3 1))
645         (setq COUNT (+ COUNT 1))
646       )
647       (setq Z3 0)
648       (setq Z2 (+ Z2 1))
649     )
650     (setq Z2 0)
651     (setq Z1 (+ Z1 1))
652   )
653   (setq Z1 0)
654   (setq Z0 (+ Z0 1))
655 )
656 (if (/= (get_tile "DC_2D") "1")
657   (progn
658     (setq PERM (reverse PERM))
659     (start_list "DC_POP")
660     (mapcar 'add_list PERM)
661     (end_list)
662   )
663   (progn
664     (setq PERMo nil)
665     (setq P 0)
666     (while (<= P (- COUNT 1))
667       (setq P1 (strcat "*" (nth P PERM)))
668       (setq P2 (strcat (substr (nth P PERM) 1 2 )
669         "*" (substr (nth P PERM) 4 14 ) ))
670       (setq P3 (strcat (substr (nth P PERM) 1 5 )
671         "*" (substr (nth P PERM) 7 14 ) ))
672       (setq P4 (strcat (substr (nth P PERM) 1 8 )
673         "*" (substr (nth P PERM) 10 14 ) ))
674       (setq P5 (strcat (substr (nth P PERM) 1 11 )
675         "*" (substr (nth P PERM) 13 14 ) ))
676       (setq P6 (strcat (nth P PERM) "*" ))
677       (setq PERMo (cons P1 PERMo))
678       (setq PERMo (cons P2 PERMo))
679       (setq PERMo (cons P3 PERMo))
680       (setq PERMo (cons P4 PERMo))
681       (setq PERMo (cons P5 PERMo))
682       (setq PERMo (cons P6 PERMo))
683       (setq P (+ P 1))
684     )
685     (setq PERM PERMo)
686     (start_list "DC_POP")
687     (mapcar 'add_list PERM)

```

```

688         (end_list)
689         (setq COUNT (* COUNT 6))
690     )
691 )
692
693 (setq C (itoa COUNT))
694 (set_tile "DC_PER" C)
695 (mode_tile "DC_CZ" 0)
696 (mode_tile "DC_DS" 0)
697 )
698
699 (defun DO_PERM ( / L1 L2 L3 L4 Z1 Z2)
700 (TEST)
701 (setq L1 (length KOMB))
702 (cond ((= L1 2) (PER_2))
703       ((= L1 3) (PER_3))
704       ((= L1 4) (PER_4))
705       ((= L1 5) (PER_5))
706 )
707 )
708
709 (defun D ( X Y / A ) (setq A (nth Y (nth X DETAY))) (eval A))
710
711 (defun LA ( boy DUZ YER ang / M X Y)
712 (setq la_angle ang)
713 (cond
714   ((= YER "bas")
715     (progn
716       (setq M (+ boy 130 (D 0 3) (* 2 (D 0 4))))
717       (setq X (+ XP1 boy 130 (/ (D 0 3) 2) (D 0 4)))
718     )
719   )
720   ((= YER "ara")
721     (progn
722       (setq M (+ boy (D 0 3) (* 2 (D 0 4)) DUZ))
723       (setq X (+ XP1 boy (/ (D 0 3) 2) (D 0 4) DUZ))
724     )
725   )
726   ((= YER "son")
727     (progn
728       (setq M (+ boy 150 (D 0 3) (* 2 (D 0 4)) DUZ))
729       (setq X (+ XP1 boy (/ (D 0 3) 2) (D 0 4) DUZ))
730     )
731   )
732 )
733 (setq LA_IN (list X YP1 400))
734 (setq LA_IN_S (list X YP1 200))
735 (eval M)
736 )
737
738 (defun KL ( boy DUZ YER ang / M X Y)
739 (setq kl_angle ang)
740 (cond
741   ((= YER "bas")
742     (progn
743       (setq M (+ boy 130 (D 1 3) (* 2 (D 1 4))))
744       (setq X (+ XP1 boy 130 (/ (D 1 3) 2) (D 1 4)))
745     )
746   )
747   ((= YER "ara")
748     (progn
749       (setq M (+ boy (D 1 3) (* 2 (D 1 4)) DUZ))
750       (setq X (+ XP1 boy (/ (D 1 3) 2) (D 1 4) DUZ))

```

```

751     )
752   )
753   ((= YER "son")
754     (progn
755       (setq M (+ boy 150 (D 1 3) (* 2 (D 1 4)) DUZ))
756       (setq X (+ XP1 boy (/ (D 1 3) 2) (D 1 4) DUZ))
757     )
758   )
759 )
760 (setq KL_IN (list X YP1 400))
761 (setq KL_IN_S (list X YP1 200))
762 (eval M)
763 )
764
765 (defun CM ( boy DUZ YER ang / M )
766   (setq cm_angle ang)
767   (cond
768     ((= YER "bas")
769       (progn
770         (setq M (+ (D 2 3) (D 2 4)))
771         (setq X (+ XP1 (/ (D 2 3) 2) ))
772       )
773     )
774     ((= YER "ara")
775       (progn
776         (setq M (+ boy (D 2 3) (* 2 (D 2 4)) DUZ))
777         (setq X (+ XP1 boy (/ (D 2 3) 2) (D 2 4) DUZ))
778       )
779     )
780     ((= YER "son")
781       (progn
782         (setq M (+ boy (D 2 3) (D 2 4) DUZ))
783         (setq X (+ XP1 boy (/ (D 2 3) 2) (D 2 4) DUZ))
784       )
785     )
786   )
787 (setq CM_IN (list X YP1 400))
788 (setq CM_IN_S (list X YP1 200))
789 (eval M)
790 )
791
792 (defun DT ( boy DUZ YER ang / M )
793   (setq dt_angle ang)
794   (if (= YER "bas")
795     (progn
796       (setq M (+ (D 3 3) (D 3 4)))
797       (setq X (+ XP1 (/ (D 3 3) 2)))
798       (setq EN1 2000)
799     )
800     (progn
801       (setq M (+ boy (D 3 3) (D 3 4) DUZ))
802       (setq X (+ XP1 boy (/ (D 3 3) 2) (D 3 4) DUZ))
803     )
804   )
805 (setq DT_IN (list X YP1 400))
806 (setq DT_IN_S (list X YP1 200))
807 (eval M)
808 )
809
810 (defun KU ( boy DUZ YER ang / M )
811   (setq ku_angle ang)
812   (setq M (+ boy (D 4 3) (D 4 4) DUZ))
813   (setq X (+ XP1 boy (/ (D 4 3) 2) (D 4 4) DUZ))

```

```

814 (setq KU_IN (list X YP1 400))
815 (setq KU_IN_S (list X YP1 200))
816 (eval M)
817 )
818
819 (defun SF ( boy DUZ YER ang / M )
820 (setq sf_angle ang)
821 (cond
822 ((= YER "bas")
823 (progn
824 (setq M (+ (D 5 3) (* (D 5 4) 2)))
825 (setq X (+ XP1 (/ (D 5 3) 2)))
826 )
827 )
828 ((= YER "ara")
829 (progn
830 (setq M (+ boy (D 5 3) (* 2 (D 5 4)) DUZ))
831 (setq X (+ XP1 boy (/ (D 5 3) 2) (D 5 4) DUZ))
832 )
833 )
834 ((= YER "son")
835 (progn
836 (setq M (+ boy (D 5 3) (D 5 4) DUZ))
837 (setq X (+ XP1 boy (/ (D 5 3) 2) (D 5 4) DUZ))
838 )
839 )
840 )
841 (setq SF_IN (list X YP1 400))
842 (setq SF_IN_S (list X YP1 200))
843 (eval M)
844 )
845
846 (defun TS ( boy DUZ YER ang / M )
847 (setq ts_angle ang)
848 (cond
849 ((= YER "bas")
850 (progn
851 (setq M (+ (D 6 3) (* (D 6 4) 2)))
852 (setq X (+ XP1 (/ (D 6 3) 2)))
853 )
854 )
855 ((= YER "ara")
856 (progn
857 (setq M (+ boy (D 6 3) (* 2 (D 6 4)) DUZ))
858 (setq X (+ XP1 boy (/ (D 6 3) 2) (D 6 4) DUZ))
859 )
860 )
861 ((= YER "son")
862 (progn
863 (setq M (+ boy (D 6 3) (D 6 4) DUZ))
864 (setq X (+ XP1 boy (/ (D 6 3) 2) (D 6 4) DUZ))
865 )
866 )
867 )
868 (setq TS_IN (list X YP1 400))
869 (setq TS_IN_S (list X YP1 200))
870 (eval M)
871 )
872
873 (defun POS ( L sira / YER Y )
874 (setq Y (length L))
875 (cond
876 ((= sira 1) (setq YER "bas")))

```

```

877      ((and (> sira 1) (< sira Y)) (setq YER "ara"))
878      ((= sira Y) (setq YER "son"))
879    )
880    (eval YER)
881  )
882
883  (defun BOYUTLA_1D ( P ang P1 / p1 p ang L A X DUZ boy )
884    (setq L (SOLVE P))
885    (setq XP1 (car P1))
886    (setq YP1 (cadr P1))
887    (setq EN1 1700)
888    (setq sira 1) ;düzeltmeyi 2.den sonra kullan
889    (setq DUZ 0) ;düzeltme
890    (setq boy 0) ;yatay boyut
891    (foreach A L
892      (setq YER (POS L sira))
893      (cond
894        ((= A "LA") (setq boy (LA boy DUZ YER ang)) )
895        ((= A "KL") (setq boy (KL boy DUZ YER ang)) )
896        ((= A "CM") (setq boy (CM boy DUZ YER ang)) )
897        ((= A "DT") (setq boy (DT boy DUZ YER ang)) )
898        ((= A "KU") (setq boy (KU boy DUZ YER ang)) )
899        ((= A "SF") (setq boy (SF boy DUZ YER ang)) )
900        ((= A "TS") (setq boy (TS boy DUZ YER ang)) )
901      )
902      (setq DUZ (- 0 100))
903      (setq sira (+ sira 1))
904    )
905    (setq BO1 boy)
906  )
907
908  (defun la3x ( son en / )
909    (cond
910      ((= son "KL") (setq en (+ en 700) bol (+ bol 700)
911        xp1 (+ xp1 700)))
912      ((= son "CM") (setq en (+ en (d 2 5) (d 2 8))))
913      ((and (= son "DT") (< bol 1700))
914        (setq en (+ en (d 3 5) (d 3 8))
915          bol (+ bol 1000)
916          xp1 (+ xp1 1000)
917        )
918      )
919      ((and (= son "DT") (>= bol 1700))
920        (setq en (+ en (d 3 5) (d 3 8))))
921      ((and (= son "SF") (< bol 1700))
922        (setq en (+ en (d 5 5) (d 5 8))
923          bol (+ bol 1000)
924          xp1 (+ xp1 1000)
925        )
926      )
927      ((and (= son "SF") (>= bol 1700))
928        (setq en (+ en (d 5 5) (d 5 8))))
929      ((= son "TS") (setq en (+ en 600) bol (+ bol 600)
930        xp1 (+ xp1 600)))
931    )
932    (eval en)
933  )
934
935  (defun LA3 ( en DUZ YER ang son / M X Y)
936    (setq la_angle ang)
937    (cond
938      ((= YER "bas")
939        (progn

```

```

940      (if son (setq en (la3x son en)))
941      (setq M (+ en 130 (D 0 3) (* 2 (D 0 4))))
942      (setq X (+ YP1 en 130 (/ (D 0 3) 2) (D 0 4)))
943    )
944  )
945  ((= YER "ara")
946    (progn
947      (setq M (+ en (D 0 3) (* 2 (D 0 4)) DUZ))
948      (setq X (+ YP1 en (/ (D 0 3) 2) (D 0 4) DUZ))
949    )
950  )
951  ((= YER "son")
952    (progn
953      (setq M (+ en 150 (D 0 3) (* 2 (D 0 4)) DUZ))
954      (setq X (+ YP1 en (/ (D 0 3) 2) (D 0 4) DUZ))
955    )
956  )
957  )
958  (setq LA_IN (list XP1 X 400))
959  (setq LA_IN_S (list XP1 X 200))
960  (eval M)
961  )
962
963  (defun kl3x ( son en / )
964    (cond
965      ((= son "LA") (setq en (+ en 700) bol (+ bol 700)
966        xp1 (+ xp1 700)))
967      ((= son "CM") (setq en (+ en (d 2 5) (d 2 8))))
968      ((and (= son "DT") (< bol 1700))
969        (setq en (+ en (d 3 5) (d 3 8))
970          bol (+ bol 1000)
971          xp1 (+ xp1 1000)
972        )
973      )
974      ((and (= son "DT") (>= bol 1700))
975        (setq en (+ en (d 3 5) (d 3 8))))
976      ((and (= son "SF") (< bol 1700))
977        (setq en (+ en (d 5 5) (d 5 8))
978          bol (+ bol 1000)
979          xp1 (+ xp1 1000)
980        )
981      )
982      ((and (= son "SF") (>= bol 1700))
983        (setq en (+ en (d 5 5) (d 5 8))))
984      ((= son "TS") (setq en (+ en 600) bol (+ bol 600)
985        xp1 (+ xp1 600)))
986    )
987    (eval en)
988  )
989
990  (defun KL3 ( en DUZ YER ang son / M X Y)
991    (setq kl_angle ang)
992    (cond
993      ((= YER "bas")
994        (progn
995          (if son (setq en (kl3x son en)))
996          (setq M (+ en 130 (D 1 3) (* 2 (D 1 4))))
997          (setq X (+ YP1 en 130 (/ (D 1 3) 2) (D 1 4)))
998        )
999      )
1000    ((= YER "ara")
1001      (progn
1002        (setq M (+ en (D 1 3) (* 2 (D 1 4)) DUZ))

```

```

1003      (setq X (+ YP1 en (/ (D 1 3) 2) (D 1 4) DUZ))
1004    )
1005  )
1006  ((= YER "son")
1007    (progn
1008      (setq M (+ en 150 (D 1 3) (* 2 (D 1 4)) DUZ))
1009      (setq X (+ YP1 en (/ (D 1 3) 2) (D 1 4) DUZ))
1010    )
1011  )
1012 )
1013 (setq KL_IN (list XP1 X 400))
1014 (setq KL_IN_S (list XP1 X 200))
1015 (eval M)
1016 )
1017
1018 (defun cm3x ( son en / )
1019   (cond
1020     ((or (= son "LA") (= son "KL") (= son "SF") (= son "TS")))
1021       (setq bol (+ bol (d 2 5) (d 2 8))
1022         xpl (+ xpl (d 2 5) (d 2 8))
1023       )
1024     )
1025     ((= son "DT") (setq en (+ en (d 3 5) (d 3 8))))
1026   )
1027   (eval en)
1028 )
1029
1030 (defun CM3 ( en DUZ YER ang son / M )
1031   (setq cm_angle ang)
1032   (cond
1033     ((= YER "bas")
1034       (progn
1035         (if son (setq en (cm3x son en)))
1036         (setq M (+ (D 2 3) (D 2 4)))
1037         (setq X (+ YP1 (/ (D 2 3) 2) ))
1038         (setq EN1 1800)
1039       )
1040     )
1041     ((= YER "ara")
1042       (progn
1043         (setq M (+ en (D 2 3) (* 2 (D 2 4)) DUZ))
1044         (setq X (+ YP1 en (/ (D 2 3) 2) (D 2 4) DUZ))
1045       )
1046     )
1047     ((= YER "son")
1048       (progn
1049         (setq M (+ en (D 2 3) (D 2 4) DUZ))
1050         (setq X (+ YP1 en (/ (D 2 3) 2) (D 2 4) DUZ))
1051       )
1052     )
1053   )
1054   (setq CM_IN (list XP1 X 400))
1055   (setq CM_IN_S (list XP1 X 200))
1056   (eval M)
1057 )
1058
1059 (defun dt3x ( son en / )
1060   (cond
1061     ((or (= son "LA") (= son "KL") (= son "SF") (= son "TS")))
1062       (setq bol (+ bol (d 3 5) (d 3 8))
1063         xpl (+ xpl (d 3 5) (d 3 8))
1064       )
1065     )

```

```

1066      ((= son "CM") (setq en (+ en (d 2 5) (d 2 8))))
1067    )
1068    (eval en)
1069  )
1070
1071  (defun DT3 ( en DUZ YER ang son / M )
1072    (setq dt_angle ang)
1073    (if (= YER "bas")
1074      (progn
1075        (if son (setq en (dt3x son en)))
1076        (setq M (+ (D 3 3) (D 3 4)))
1077        (setq X (+ YP1 (/ (D 3 3) 2)))
1078        (setq EN1 2000)
1079      )
1080      (progn
1081        (setq M (+ en (D 3 3) (D 3 4) DUZ))
1082        (setq X (+ YP1 en (/ (D 3 3) 2) (D 3 4) DUZ))
1083      )
1084    )
1085    (setq DT_IN (list XP1 X 400))
1086    (setq DT_IN_S (list XP1 X 200))
1087    (eval M)
1088  )
1089
1090  (defun KU3 ( en DUZ YER ang son / M x )
1091    (setq ku_angle ang)
1092    (if (= YER "bas")
1093      (setq M (+ (D 4 3) (D 4 4))
1094            X (+ YP1 (/ (D 4 3) 2))
1095      )
1096      (setq M (+ en (D 4 3) (D 4 4) DUZ)
1097            X (+ YP1 en (/ (D 4 3) 2) (D 4 4) DUZ)
1098      )
1099    )
1100    (setq KU_IN (list XP1 X 400))
1101    (setq KU_IN_S (list XP1 X 200))
1102    (eval M)
1103  )
1104
1105  (defun sf3x ( son en / )
1106    (cond
1107      ((or (= son "LA") (= son "KL"))
1108        (setq bol (+ bol (d 5 5) (d 5 8))
1109              xpl (+ xpl (d 5 5) (d 5 8))
1110        )
1111      )
1112      ((= son "CM") (setq en (+ en (d 2 5) (d 2 8))))
1113      ((and (= son "DT") (< bol 1700))
1114        (setq en (+ en (d 3 5) (d 3 8))
1115              bol (+ bol 1000)
1116              xpl (+ xpl 1000)
1117        )
1118      )
1119      ((and (= son "DT") (>= bol 1700))
1120        (setq en (+ en (d 3 5) (d 3 8))))
1121    )
1122    (eval en)
1123  )
1124
1125  (defun SF3 ( en DUZ YER ang son / M )
1126    (setq sf_angle ang)
1127    (cond
1128      ((= YER "bas")

```

```

1129      (progn
1130        (if son (setq en (sf3x son en)))
1131        (setq M (+ (D 5 3) (* (D 5 4) 2)))
1132        (setq X (+ YP1 (/ (D 5 3) 2)))
1133      )
1134    )
1135    ((= YER "ara")
1136      (progn
1137        (setq M (+ en (D 5 3) (* 2 (D 5 4)) DUZ))
1138        (setq X (+ YP1 en (/ (D 5 3) 2) (D 5 4) DUZ))
1139      )
1140    )
1141    ((= YER "son")
1142      (progn
1143        (setq M (+ en (D 5 3) (D 5 4) DUZ))
1144        (setq X (+ YP1 en (/ (D 5 3) 2) (D 5 4) DUZ))
1145      )
1146    )
1147  )
1148  (setq SF_IN (list XP1 X 400))
1149  (setq SF_IN_S (list XP1 X 200))
1150  (eval M)
1151 )
1152
1153 (defun ts3x ( son en / )
1154   (cond
1155     ((or (= son "LA") (= son "KL"))
1156       (setq en (+ en 600) bol (+ bol 600)
1157         xpl (+ xpl 600))
1158     )
1159   )
1160   ((= son "CM") (setq en (+ en (d 2 5) (d 2 8))))
1161   ((= son "DT") (setq en (+ en (d 3 5) (d 3 8))))
1162 )
1163 (eval en)
1164 )
1165
1166 (defun TS3 ( en DUZ YER ang son / M )
1167   (setq ts_angle ang)
1168   (cond
1169     ((= YER "bas")
1170       (progn
1171         (if son (setq en (ts3x son en)))
1172         (setq M (+ (D 6 3) (* (D 6 4) 2)))
1173         (setq X (+ YP1 (/ (D 6 3) 2)))
1174       )
1175     )
1176     ((= YER "ara")
1177       (progn
1178         (setq M (+ en (D 6 3) (* 2 (D 6 4)) DUZ))
1179         (setq X (+ YP1 en (/ (D 6 3) 2) (D 6 4) DUZ))
1180       )
1181     )
1182     ((= YER "son")
1183       (progn
1184         (setq M (+ en (D 6 3) (D 6 4) DUZ))
1185         (setq X (+ YP1 en (/ (D 6 3) 2) (D 6 4) DUZ))
1186       )
1187     )
1188   )
1189   (setq TS_IN (list XP1 X 400))
1190   (setq TS_IN_S (list XP1 X 200))
1191   (eval M)

```

```

1192 )
1193
1194 (defun BOYUTLA_3D ( P ang P1 Ax / p1 p ang Ax L A X DUZ en )
1195   (setq L (SOLVE P))
1196   (setq XP1 (+ BO1 (car P1)))
1197   (setq YP1 (cadr P1))
1198   (setq siras 1) ;düzeltmeyi 2.den sonra kullan
1199   (setq DUZ 0) ;düzeltme
1200   (setq en 0) ;düsey boyut
1201   (foreach A L
1202     (setq YER (POS L siras))
1203     (cond
1204       ((= A "LA") (setq en (LA3 en DUZ YER ang Ax)) )
1205       ((= A "KL") (setq en (KL3 en DUZ YER ang Ax)) )
1206       ((= A "CM") (setq en (CM3 en DUZ YER ang Ax)) )
1207       ((= A "DT") (setq en (DT3 en DUZ YER ang Ax)) )
1208       ((= A "KU") (setq en (KU3 en DUZ YER ang Ax)) )
1209       ((= A "SF") (setq en (SF3 en DUZ YER ang Ax)) )
1210       ((= A "TS") (setq en (TS3 en DUZ YER ang Ax)) )
1211     )
1212     (setq DUZ (- 0 100))
1213     (setq siras (+ siras 1))
1214   )
1215   (setq EN1 en)
1216 )
1217
1218 (defun BOYUTLA_2D ( P P1 / p1 p l1 l2 z ang L A X DUZ boy Ax)
1219   (setq z (SOLVE_2D P))
1220   (setq son nil l1 nil)
1221   (if (= "" (car z))
1222     (setq l1 nil)
1223     (setq l1 (car z) Ax (car (reverse (solve l1)))))
1224   )
1225   (if (= "" (cadr z))
1226     (setq l2 nil)
1227     (setq l2 (cadr z))
1228   )
1229   (setq ang 0)
1230   (if l1
1231     (boyutla_1d l1 ang p1)
1232     (setq BO1 1700)
1233   )
1234   (setq ang 90)
1235   (if l2 (boyutla_3d l2 ang p1 Ax))
1236 )
1237
1238 (defun solve_2d ( l / kk mx z y nn nm z1 z2 )
1239   (setq mx (strlen l))
1240   z nil
1241   y 1
1242   nn ""
1243 )
1244 (repeat mx
1245   (setq z (cons (substr l y 1) z))
1246   (setq y (+ 1 y))
1247 )
1248 (setq z1 (reverse (cdr (member "*" z))))
1249 (setq z2 (cdr (member "*" (reverse z))))
1250 (if (length z1) (foreach nm z1 (setq nn (strcat nn nm)))
1251   (setq nn nil)
1252 )
1253 (setq z1 nn)
1254 (setq nn "")

```

```

1255     (if (length z2) (foreach nm z2 (setq nn (strcat nn nm)))
1256         (setq nn nil)
1257     )
1258     (setq z2 nn)
1259     (setq nn (list z1 z2))
1260 )
1261
1262 (defun TEST_1 ()
1263     (setq KOMB nil)
1264     (if (= LA1 1) (setq SAY (+ 1 SAY)
1265         KOMBIO (strcat KOMBIO "LA ")
1266         KOMB (cons "LA" KOMB) ))
1267     (if (= KL1 1) (setq SAY (+ 1 SAY)
1268         KOMBIO (strcat KOMBIO "KL ")
1269         KOMB (cons "KL" KOMB) ))
1270     (if (= CM1 1) (setq SAY (+ 1 SAY)
1271         KOMBIO (strcat KOMBIO "CM ")
1272         KOMB (cons "CM" KOMB) ))
1273     (if (= YK1 1) (progn (setq SAY (+ 1 SAY))
1274         (if (= DT1 1) (setq KOMBIO (strcat KOMBIO "DT ")
1275             KOMB (cons "DT" KOMB) )
1276             (setq KOMBIO (strcat KOMBIO "KU ")
1277                 KOMB (cons "KU" KOMB) ) ) )
1278     (if (= SS1 1) (progn (setq SAY (+ 1 SAY))
1279         (if (= SF1 1) (setq KOMBIO (strcat KOMBIO "SF ")
1280             KOMB (cons "SF" KOMB) )
1281             (setq KOMBIO (strcat KOMBIO "TS ")
1282                 KOMB (cons "TS" KOMB) ) ) )
1283     (setq KOMBIO (substr KOMBIO 1 (- (strlen KOMBIO) 1)) )
1284     (setq KOMB (reverse KOMB))
1285     (set_tile "DC_KOM" KOMBIO)
1286 )
1287
1288 (defun TEST ()
1289     (setq SAY 0 KOMBIO "")
1290     LA1(atoi(get_tile "DC_LA")) KL1 (atoi (get_tile "DC_KL"))
1291     CM1(atoi(get_tile "DC_CM")) YK1 (atoi (get_tile "DC_YK"))
1292     DT1(atoi(get_tile "DC_DT")) KU1 (atoi (get_tile "DC_KU"))
1293     SS1(atoi(get_tile "DC_SS")) SF1 (atoi (get_tile "DC_SF"))
1294     TS1 (atoi (get_tile "DC_TS")) )
1295     (setq SAY_1 (+ LA1 KL1 CM1 YK1 SS1 ))
1296     (if (>= SAY_1 2)
1297         (TEST_1)
1298     )
1299 )
1300
1301 (defun GET_FIRMA ( / flo NM LNM fil Lfil c )
1302     (setq acad_ver (substr (getvar "acadver") 1 2))
1303     (setq c "Programin calismasi icin AutoCAD R12
1304         ya da R13 gereklidir.")
1305     (if (and (/= acad_ver "12") (/= acad_ver "13"))
1306         (progn
1307             (print c)
1308             (exit)
1309         )
1310     (setq NM (getvar "dwgname"))
1311     (if (and (< (strlen NM) 7) (= acad_ver "12"))
1312         (progn
1313             (print "\nLütfen Banyol2.dwg dosyasini açınız...")
1314             (exit)
1315         )
1316     )

```

```

1317 (if (and (< (strlen NM) 7) (= acad_ver "13"))
1318     (progn
1319         (print "\nLütfen Banyo13.dwg dosyasini açiniz...")
1320         (exit)
1321     )
1322 )
1323 (setq LNM (strcase (substr NM (- (strlen NM) 6) 7) T))
1324 (if (and (/= "banyo12" LNM) (= acad_ver "12"))
1325     (progn
1326         (print "\nLütfen Banyo12.dwg dosyasini açiniz...")
1327         (exit)
1328     )
1329 )
1330 (if (and (/= "banyo13" LNM) (= acad_ver "13"))
1331     (progn
1332         (print "\nLütfen Banyo13.dwg dosyasini açiniz...")
1333         (exit)
1334     )
1335 )
1336 (if (null (setq flo
1337     (open (strcat (getvar "dwgprefix") "Banyo.dta") "r")))
1338     (progn
1339         (princ "\n\"Banyo.dta\" dosyasi açilamadi!")
1340         (exit)
1341     )
1342 )
1343 (setq FIRMA nil)
1344 (while (setq LINE (read-line flo))
1345     (if (= (substr LINE 1 1) "*")
1346         (progn
1347             (setq FIRMA
1348                 (cons (substr LINE 2 (strlen LINE)) FIRMA))
1349         )
1350     )
1351 )
1352 (close flo)
1353 (setq FIRMA (reverse FIRMA))
1354 )
1355
1356 (defun GET_ALL ( / Temp1 Temp2 )
1357     (if (null (setq flo
1358         (open (strcat (getvar "dwgprefix") "Banyo.dta") "r")))
1359         (progn
1360             (princ "\n\"Banyo.dta\" dosyasi açilamadi!")
1361             (exit)
1362         )
1363     )
1364     (setq Temp1 (strcat F "_Vitrifiye_"))
1365     (setq Temp2 (strlen Temp1))
1366     (setq VTR nil)
1367     (while (setq LINE (read-line flo))
1368         (if (= (substr LINE 1 Temp2) Temp1)
1369             (progn
1370                 (setq VTR (cons (substr LINE (+ Temp2 1)
1371                     (strlen LINE)) VTR))
1372             )
1373         )
1374     )
1375     (close flo)
1376     (if (null (setq flo
1377         (open (strcat (getvar "dwgprefix") "Banyo.dta") "r")))
1378         (progn
1379             (princ "\n\"Banyo.dta\" dosyasi açilamadi!")

```

```

1380      (exit)
1381    )
1382  )
1383  (setq Temp1 (strcat F "_Yer Karosu_"))
1384  (setq Temp2 (strlen Temp1))
1385  (setq YKAP nil)
1386  (while (setq LINE (read-line flo))
1387    (if (= (substr LINE 1 Temp2) Temp1)
1388      (progn
1389        (setq YKAP (cons (substr LINE (+ Temp2 1)
1390          (strlen LINE)) YKAP))
1391      )
1392    )
1393  )
1394  (close flo)
1395  (if (null (setq flo
1396    (open (strcat (getvar "dwgprefix") "Banyo.dta") "r"))))
1397    (progn
1398      (princ "\n\"Banyo.dta\" dosyasi açilamadi!")
1399      (exit)
1400    )
1401  )
1402  (setq Temp1 (strcat F "_Duvar Karosu_"))
1403  (setq Temp2 (strlen Temp1))
1404  (setq DKAP nil)
1405  (while (setq LINE (read-line flo))
1406    (if (= (substr LINE 1 Temp2) Temp1)
1407      (progn
1408        (setq DKAP (cons (substr LINE (+ Temp2 1)
1409          (strlen LINE)) DKAP) )
1410      )
1411    )
1412  )
1413  (close flo)
1414  (setq VITR (reverse VITR))
1415  (setq YKAP (reverse YKAP))
1416  (setq DKAP (reverse DKAP))
1417 )
1418
1419 (defun chk_TP_DIALOG3 ( fir / F )
1420   (setq f (nth FIR FIRMA))
1421   (if (/= nil f) (tp_dialog3 f))
1422 )
1423
1424 (defun TP_DIALOG3 ( f / f x y xx tile_list tile_list1
1425   tile_list2 tile_list3)
1426   (setq DCL_ID ( load_dialog "Banyo.DCL"))
1427   (if (not (new_dialog "DDBanyo3" DCL_ID)) (exit))
1428   (setq TILE_LIST '("icx11" "icx12" "icx13" "icx14" "icx15"))
1429   (GET_ALL)
1430   (setq TILE_LIST1 (list "icx311" "icx312" "icx313"
1431     "icx314" "icx315" ))
1432   (setq TILE_LIST2 (list "icx321" "icx322" "icx323"
1433     "icx324" "icx325" ))
1434   (setq TILE_LIST3 (list "icx331" "icx332" "icx333"
1435     "icx334" "icx335" ))
1436
1437   (setq XX 0)
1438   (while (<= XX (~ (length VITR) 1))
1439     (set_tile(strcat (nth XX TILE_LIST1) "x") (nth XX VITR) )
1440     ( setq x ( dimx_tile (nth XX TILE_LIST1))
1441       y ( dimy_tile (nth XX TILE_LIST1)) )
1442     ( start_image (nth XX TILE_LIST1) )

```

```

1439      ( slide_image 0 0 x y (strcat F "(" (nth XX VITR) ")") )
1440      ( end_image )
1441      (setq XX (+ 1 XX))
1442    )
1443    (setq XX 0)
1444    (while (<= XX (- (length YKAP) 1))
1445      (set_tile(strcat (nth XX TILE_LIST2) "x") (nth XX YKAP) )
1446      (setq x ( dimx_tile (nth XX TILE_LIST2))
1447            y ( dimy_tile (nth XX TILE_LIST2)) )
1448      (start_image (nth XX TILE_LIST2) )
1449      (slide_image 0 0 x y (strcat "Karo(" (nth XX YKAP) ")") )
1450      (end_image)
1451      (setq XX (+ 1 XX))
1452    )
1453    (setq XX 0)
1454    (while (<= XX (- (length DKAP) 1))
1455      (set_tile(strcat (nth XX TILE_LIST3) "x") (nth XX DKAP) )
1456      (setq x ( dimx_tile (nth XX TILE_LIST3))
1457            y ( dimy_tile (nth XX TILE_LIST3)) )
1458      (start_image (nth XX TILE_LIST3) )
1459      (slide_image 0 0 x y (strcat "Karo(" (nth XX DKAP) ")") )
1460      (end_image)
1461      (setq XX (+ 1 XX))
1462    )
1463    (set_tile "DC_TP3" F)
1464
1465    (action_tile "icx311" "(set_vitr 0)")
1466    (action_tile "icx312" "(set_vitr 1)")
1467    (action_tile "icx313" "(set_vitr 2)")
1468    (action_tile "icx314" "(set_vitr 3)")
1469    (action_tile "icx315" "(set_vitr 4)")
1470
1471    (action_tile "icx321" "(set_ykap 0)")
1472    (action_tile "icx322" "(set_ykap 1)")
1473    (action_tile "icx323" "(set_ykap 2)")
1474    (action_tile "icx324" "(set_ykap 3)")
1475    (action_tile "icx325" "(set_ykap 4)")
1476
1477    (action_tile "icx331" "(set_dkap 0)")
1478    (action_tile "icx332" "(set_dkap 1)")
1479    (action_tile "icx333" "(set_dkap 2)")
1480    (action_tile "icx334" "(set_dkap 3)")
1481    (action_tile "icx335" "(set_dkap 4)")
1482
1483    (action_tile "DC3_IP" "(setq cnc1 \"1\") (done_dialog)")
1484    (action_tile "DC3_TM" "(UPDATE_X f) (done_dialog)")
1485    (start_dialog)
1486    (unload_dialog DCL_ID)
1487  )
1488
1489  (defun UPDATE_X ( f / )
1490    (setq A1 (get_tile "DC3_VT"))
1491    (setq A2 (get_tile "DC3_YK"))
1492    (setq A3 (get_tile "DC3_DK"))
1493    (if(and(= A1 "1") (/= nil T_SVT))(setq SVT T_SVT SVT1 F))
1494    (if(and(= A2 "1") (/= nil T_SKP))(setq SKP T_SKP SKP1 F))
1495    (if(and(= A3 "1") (/= nil T_SDK))(setq SDK T_SDK SDK1 F))
1496  )
1497
1498  (defun set_vitr ( num / )
1499    (setq T_SVT (nth num VITR))
1500    (set_tile "DC_TP31" (nth num VITR))
1501    (set_tile "DC3_VT" "1")

```

```

1502 )
1503
1504 (defun set_ykap (num / )
1505   (if (/= nil (nth num YKAP))
1506     (progn
1507       (setq T_SKP (nth num YKAP))
1508       (set_tile "DC_TP32" (nth num YKAP))
1509       (set_tile "DC3_YK" "1")
1510     )
1511   )
1512 )
1513
1514 (defun set_dkap ( num / )
1515   (if (/= nil (nth num DKAP))
1516     (progn
1517       (setq T_SDK (nth num DKAP))
1518       (set_tile "DC_TP33" (nth num DKAP))
1519       (set_tile "DC3_DK" "1")
1520     )
1521   )
1522 )
1523
1524 (defun UPDATE_TP2 ()
1525   (if (= A1 "1")
1526     (progn
1527       (set_tile "icx21x" SVT1)
1528       (set_tile "icx21y" SVT)
1529       (setq x (dimx_tile "icx21" )
1530             y (dimy_tile "icx21" ))
1531       (start_image "icx21" )
1532       (fill_image 0 0 x y -2 )
1533       (slide_image 0 0 x y (strcat SVT1 "(" SVT ")") )
1534       (end_image )
1535     )
1536   )
1537   (if (= A2 "1")
1538     (progn
1539       (set_tile "icx22x" SKP1)
1540       (set_tile "icx22y" SKP)
1541       (setq x (dimx_tile "icx22" )
1542             y (dimy_tile "icx22" ))
1543       (start_image "icx22" )
1544       (fill_image 0 0 x y -2 )
1545       (slide_image 0 0 x y (strcat "Karo(" SKP ")") )
1546       (end_image )
1547     )
1548   )
1549   (if (= A3 "1")
1550     (progn
1551       (set_tile "icx23x" SDK1)
1552       (set_tile "icx23y" SDK)
1553       (setq x (dimx_tile "icx23" )
1554             y (dimy_tile "icx23" ))
1555       (start_image "icx23" )
1556       (fill_image 0 0 x y -2 )
1557       (slide_image 0 0 x y (strcat "Karo(" SDK ")") )
1558       (end_image )
1559     )
1560   )
1561 )
1562
1563 (defun TP_DIALOG2 ( / X Y1 Y2 Y3)
1564   (GET_FIRMA)

```

```

1565 (setq DCL_ID ( load_dialog "Banyo.DCL"))
1566 ( if ( not ( new_dialog "DDbanyo2" DCL_ID )) ( exit ) )
1567 (setq TILE_LIST ("icx11" "icx12" "icx13" "icx14" "icx15"))
1568 (setq X 0)
1569 (setq SLD_LIST nil)
1570 (while (< X (length FIRMA))
1571   (setq Y1 (nth X TILE_LIST))
1572   (setq Y2 (nth X FIRMA))
1573   (setq Y2 (strcat Y2 "(" Y2 " "))
1574   (setq SLD_LIST ( cons (list Y1 Y2) SLD_LIST ) )
1575   (setq X (+ 1 X))
1576 )
1577 (setq SLD_LIST (reverse SLD_LIST))
1578 ( foreach ls SLD_LIST
1579   ( setq ls1 (car ls)
1580     ls2 (cadr ls)
1581     x ( dimx_tile ls1 )
1582     y ( dimy_tile ls1 )
1583     ( start_image ls1 )
1584     ( slide_image 0 0 x y ls2 )
1585     ( end_image )
1586   )
1587   (setq X 0)
1588   (while (< X (length FIRMA))
1589     (set_tile (strcat (nth X TILE_LIST) "x") (nth X FIRMA) )
1590     (setq X (+ 1 X))
1591   )
1592   (set_tile "icx21x" SVT1)
1593   (set_tile "icx21y" SVT)
1594   (set_tile "icx22x" SKP1)
1595   (set_tile "icx22y" SKP)
1596   (set_tile "icx23x" SDK1)
1597   (set_tile "icx23y" SDK)
1598
1599   (setq x (dimx_tile "icx21")
1600     y (dimy_tile "icx21"))
1601   (start_image "icx21")
1602   (slide_image 0 0 x y (strcat SVT1 "(" SVT " "))
1603   (end_image)
1604
1605   (setq x (dimx_tile "icx22")
1606     y (dimy_tile "icx22"))
1607   (start_image "icx22")
1608   (slide_image 0 0 x y (strcat "Karo(" SKP " "))
1609   (end_image)
1610
1611   (setq x (dimx_tile "icx23")
1612     y (dimy_tile "icx23"))
1613   (start_image "icx23")
1614   (slide_image 0 0 x y (strcat "Karo(" SDK " "))
1615   (end_image)
1616
1617   (action_tile "icx11" "(progn (chk_TP_DIALOG3 0)
1618     (UPDATE_TP2))")
1619   (action_tile "icx12" "(progn (chk_TP_DIALOG3 1)
1620     (UPDATE_TP2))")
1621   (action_tile "icx13" "(progn (chk_TP_DIALOG3 2)
1622     (UPDATE_TP2))")
1623   (action_tile "icx14" "(progn (chk_TP_DIALOG3 3)
1624     (UPDATE_TP2))")
1625   (action_tile "icx15" "(progn (chk_TP_DIALOG3 4)
1626     (UPDATE_TP2))")

```

```

1623 (action_tile "DC2_IP" "(done_dialog)" )
1624 (action_tile "DC2_TM" "(done_dialog)" )
1625 (start_dialog )
1626 (unload_dialog DCL_ID )
1627 )
1628
1629 (defun SET_THEM_FIRST ( / FILE flo LINE TILE K )
1630 (if (= ILK "evet")
1631 (setq FILE "Banyo.def")
1632 (setq FILE "Banyo.tmp")
1633 )
1634 (if (null (setq flo
1635 (open (strcat (getvar "dwgprefix") FILE) "r")))
1636 (progn
1637 (princ (strcat "\n" FILE " dosyasi açilamadi!"))
1638 (exit)
1639 )
1640 )
1641 (while (setq LINE (read-line flo))
1642 (setq K (substr LINE 1 1) )
1643 (cond
1644 ((= K "-") (setq TILE (substr LINE 2 (strlen LINE))))
1645 ((= K "*") (set_tile TILE (substr LINE 2 (strlen LINE))))
1646 )
1647 )
1648 (close flo)
1649 (if (= FILE "Banyo.def")
1650 (setq SVT1 "Toprak"
1651 SVT "Antik"
1652 SKP1 "Toprak"
1653 SKP "20x20"
1654 SDK1 "Toprak"
1655 SDK "20x20"
1656 KP1 1
1657 TOL (get_tile "DC_SLD")
1658 )
1659 )
1660 )
1661
1662 (defun PUT_LAST ( / FL_TEMP LINET flo LINE TILE K )
1663 (if (null (setq flo
1664 (open (strcat (getvar "dwgprefix") "Banyo.def") "r")))
1665 (progn
1666 (princ "\n\"Banyo.def\" dosyasi açilamadi!")
1667 (exit)
1668 )
1669 )
1670 (if (null (setq FL_TEMP
1671 (open (strcat (getvar "dwgprefix") "Banyo.tmp") "w")))
1672 (progn
1673 (princ "\n\"Banyo.tmp\" dosyasi açilamadi!")
1674 (exit)
1675 )
1676 )
1677 (while (setq LINE (read-line flo))
1678 (setq K (substr LINE 1 1) )
1679 (cond
1680 ((= K "-") (progn
1681 (setq TILE (substr LINE 2 (strlen LINE)))
1682 (write-line LINE FL_TEMP ) ) )
1683 ((= K "*") (write-line (strcat "*" (get_tile TILE))
1684 FL_TEMP)
1685 )

```

```

1686      ((and (/= K "-") (/= K "*") (/= K ".") (/= K "+")))
1687      (write-line LINE FL_TEMP ) )
1688    )
1689  )
1690  (close flo)
1691  (close FL_TEMP)
1692 )
1693
1694 (defun UPDATE_TP1 ( / X Y )
1695   (if (/= nil SVT) (progn
1696     (set_tile "DC_MKVx" SVT1)
1697     (set_tile "DC_VTx" SVT)
1698     (setq x (dimx_tile "ic11" )
1699           y (dimy_tile "ic11" ))
1700     (start_image "ic11" )
1701     (fill_image 0 0 x y -2 )
1702     (slide_image 0 0 x y (strcat SVT1 "(" SVT ")") )
1703     (end_image )
1704   ))
1705   (if (/= nil SKP) (progn
1706     (set_tile "DC_MKKx" SKP1)
1707     (set_tile "DC_KPx" SKP)
1708     (setq x (dimx_tile "ic21" )
1709           y (dimy_tile "ic21" ))
1710     (start_image "ic21" )
1711     (fill_image 0 0 x y -2 )
1712     (slide_image 0 0 x y (strcat "Karo(" SKP ")") )
1713     (end_image )
1714   ))
1715   (if (/= nil SDK) (progn
1716     (set_tile "DC_MKDx" SDK1)
1717     (set_tile "DC_KPy" SDK)
1718   ))
1719   (if (= (get_tile "DC_1D") "1")
1720     (setq DV "1")
1721     (setq DV "2")
1722   )
1723   (if (= (get_tile "DC_CZ1") "1")
1724     (setq CZT 1)
1725     (if (= (get_tile "DC_CZ2") "1")
1726       (setq CZT 2)
1727       (setq CZT 3)
1728     )
1729   )
1730   (mode_tile "DC_CZ3x" (- 1 (atoi (get_tile "DC_CZ3"))))
1731   (mode_tile "den1" (- 1 (atoi (get_tile "DC_SS"))))
1732   (mode_tile "den2" (- 1 (atoi (get_tile "DC_YK"))))
1733 )
1734
1735 (defun CNG_TOL ( VAL WHY / )
1736   (if (or (= WHY 3) (= WHY 1))
1737     (progn
1738       (set_tile "DC_TO" VAL)
1739       (setq TOL VAL)
1740     )
1741   )
1742 )
1743
1744 (defun CNG_CZS ( VAL WHY / )
1745   (if (or (= WHY 3) (= WHY 1))
1746     (progn
1747       (set_tile "DC_CZ3x" VAL)
1748     )

```

```

1749 )
1750 )
1751
1752 (defun CNG_CSLD ( VAL WHY / )
1753   (if (or (= WHY 1) (= WHY 2))
1754     (progn
1755       (set_tile "DC_CSLD" VAL)
1756     )
1757   )
1758 )
1759
1760 (defun set_label ( / lb1 )
1761   (setq lb1 "A      ü")
1762   (setq lb1 (strcat lb1 "
1763   (setq lb1 (strcat lb1 "
1764   (setq x 100
1765   y " 1997 "
1766   y (strcat y (chr ( + x 99)))
1767   y (strcat y (chr ( + x 1)))
1768   y (strcat y (chr ( + x 8)))
1769   y (strcat y (chr ( + x 5)))
1770   y (strcat y (chr ( + x 7)))
1771   y (strcat y " ")
1772   y (strcat y (chr ( + x 121)))
1773   y (strcat y (chr ( + x 10)))
1774   y (strcat y (chr ( + x 3)))
1775   y (strcat y (chr ( + x 5)))
1776   y (strcat y (chr ( + x 10)))
1777 )
1778 (setq lb1 (strcat lb1 (chr 169) y))
1779 (set_tile "DCL_LABEL" lb1)
1780 )
1781
1782 (defun TP_DIALOG1 ( / DCL_ID Y Y1 Y2 X X1 X2 A B )
1783   (setq DCL_ID (load_dialog "Banyo.DCL"))
1784   (if (not (new_dialog "DDBanyo1" DCL_ID)) (exit))
1785   (set_label)
1786   (SET THEM FIRST)
1787   (UPDATE TP1)
1788   (DEL_POPUP)
1789   (TEST)
1790   (mode_tile "DC_PRM" 2)
1791   (action_tile "ic11" "(TP_DIALOG2) (UPDATE_TP1)")
1792   (action_tile "ic21" "(TP_DIALOG2) (UPDATE_TP1)")
1793   (action_tile "DC_LA" "(TEST) (DEL_POPUP)")
1794   (action_tile "DC_KL" "(TEST) (DEL_POPUP)")
1795   (action_tile "DC_CM" "(TEST) (DEL_POPUP)")
1796   (action_tile "DC_KU" "(TEST) (DEL_POPUP)")
1797   (action_tile "DC_DT" "(TEST) (DEL_POPUP)")
1798   (action_tile "DC_TS" "(TEST) (DEL_POPUP)")
1799   (action_tile "DC_SF" "(TEST) (DEL_POPUP)")
1800   (action_tile "DC_1D" "(progn (setq DV \"1\") (DEL_POPUP))")
1801   (action_tile "DC_2D" "(progn (setq DV \"2\") (DEL_POPUP))")
1802   (action_tile "DC_PRM" "(DO_PERM)")
1803   (action_tile "DC_SS"
1804     "(progn (TEST) (mode_tile \"den1\" (- 1 (atoi $value)))
1805       (DEL_POPUP))")
1806   (action_tile "DC_YK"
1807     "(progn (TEST) (mode_tile \"den2\" (- 1 (atoi $value)))
1808       (DEL_POPUP))")
1809   (action_tile "DC_DS" "(pre_exam (get_tile \"DC_POP\"))")
1810   (action_tile "DC_CZ1" "(cz_1)")
1811   (action_tile "DC_CZ2" "(cz_2)")

```

```

1810 (action_tile "DC_CZ3" "(cz_3)")
1811 (action_tile "DC_KR" "(rule_base)")
1812 (action_tile "DC_CZ3x" "(CNG_CSLD $value $reason)")
1813 (action_tile "DC_CSLD" "(CNG_CZS $value $reason)")
1814 (action_tile "DC_SLD" "(CNG_TOL $value $reason)")
1815 (action_tile "DC_IP" "(done_dialog 0) (exit)" )
1816 (action_tile "DC_CZ"
1817   ( strcat
1818     "(progn"
1819       "(setq CZS (atoi(get_tile \"DC_CZ3x\")))"
1820       "(PUT_LAST)"
1821       "(setq dc_pop (get_tile \"DC_POP\"))"
1822       "(done_dialog)"
1823     ")")
1824   )
1825 )
1826 (start_dialog)
1827 (unload_dialog DCL_ID)
1828 )
1829
1830 (defun pre_exam ( pop / perm_temp pop )
1831   (setq pop (atoi pop))
1832   (if (= czt 2)
1833     (progn
1834       (setq perm_temp perm)
1835       (setq perm (list (nth pop perm)))
1836       (exam)
1837       (setq perm perm_temp)
1838     )
1839     (exam)
1840   )
1841 )
1842
1843 (defun cz_1 ()
1844   (setq CZT 1)
1845   (set_tile "DC_CZ3" "0")
1846   (mode_tile "DC_CZ3x" 1)
1847 )
1848
1849 (defun cz_2 ()
1850   (setq CZT 2)
1851   (set_tile "DC_CZ3" "0")
1852   (mode_tile "DC_CZ3x" 1)
1853 )
1854
1855 (defun cz_3 ()
1856   (setq CZT 3)
1857   (set_tile "DC_CZ1" "0")
1858   (set_tile "DC_CZ2" "0")
1859   (mode_tile "DC_CZ3x" 0)
1860 )
1861
1862 ( defun DOSEME1 ( XM p1 / p1 DONGU PXX1 PXX2 INC )
1863   (if (= (rem EN XM) 0)
1864     (setq dongu (- (fix (/ EN XM)) 1))
1865     (setq dongu ( fix (/ EN XM)))
1866   )
1867   (setq INC XM)
1868   (repeat dongu
1869     (setq PXX1 (list (car P1) (+ (cadr P1) INC )) )
1870     (setq PXX2 (list (+ (car P1) BOY)
1871       (+ (cadr P1) INC))
1872   )

```

```

1873      (command "_line" PXX1 PXX2 "")
1874      (setq INC (+ INC XM))
1875    )
1876    (if (= (rem BOY XM) 0)
1877      (setq dongu (- (fix (/ BOY XM)) 1))
1878      (setq dongu (fix (/ BOY XM)))
1879    )
1880    (setq INC XM)
1881    (repeat dongu
1882      (setq PXX1 (list (+ (car P1) INC) (cadr P1)))
1883      (setq PXX2 (list (+ (car P1) INC) (+ (cadr P1) EN)))
1884      (command "_line" PXX1 PXX2 "")
1885      (setq INC (+ INC XM))
1886    )
1887  )
1888
1889  (defun DOSEME2 ( XM YM p1 / p1 DONGU PXX1 PXX2 INC )
1890    (if (= (rem EN XM) 0)
1891      (setq dongu (- (fix (/ EN XM)) 1))
1892      (setq dongu (fix (/ EN XM)))
1893    )
1894    (setq INC XM)
1895    (repeat dongu
1896      (setq PXX1 (list (car P1) (+ (cadr P1) INC)))
1897      (setq PXX2 (list (+ (car P1) BOY)
1898                      (+ (cadr P1) INC)))
1899    )
1900    (command "_line" PXX1 PXX2 "")
1901    (setq INC (+ INC XM))
1902  )
1903  (if (= (rem BOY YM) 0)
1904    (setq dongu (- (fix (/ BOY YM)) 1))
1905    (setq dongu (fix (/ BOY YM)))
1906  )
1907  (setq INC YM)
1908  (repeat dongu
1909    (setq PXX1 (list (+ (car P1) INC) (cadr P1)))
1910    (setq PXX2 (list (+ (car P1) INC) (+ (cadr P1) EN)))
1911    (command "_line" PXX1 PXX2 "")
1912    (setq INC (+ INC YM))
1913  )
1914 )
1915
1916 (defun DOSEME ( p1 / XM YM )
1917   (command "_layer" "s" "DOSEME" "")
1918   (setq XM (* 10 (atoi (substr SKP 1 2))))
1919   (setq YM (* 10 (atoi (substr SKP 4 2))))
1920   (if (= XM YM) (DOSEME1 XM p1) (DOSEME2 XM YM p1))
1921 )
1922
1923 (defun LAIN ( duzx duzy / duzx duzy x1 y1 )
1924   (setq x1 (+ DUZX (nth 0 LA_IN)))
1925   (setq y1 (+ DUZY (nth 1 LA_IN)))
1926   (setq LA_IN (list x1 y1 400))
1927   (setq LA_IN_S (list x1 y1 200))
1928 )
1929
1930 (defun KLIN ( duzx duzy / duzx duzy x1 y1 )
1931   (setq x1 (+ DUZX (nth 0 KL_IN)))
1932   (setq y1 (+ DUZY (nth 1 KL_IN)))
1933   (setq KL_IN (list x1 y1 400))
1934   (setq KL_IN_S (list x1 y1 200))
1935 )

```

```

1936
1937 (defun CMIN ( duzx duzy / duzx duzy x1 y1 )
1938   (setq x1 (+ DUZx (nth 0 CM_IN)))
1939   (setq y1 (+ DUZy (nth 1 CM_IN)))
1940   (setq CM_IN (list x1 y1 400))
1941   (setq CM_IN_S (list x1 y1 200))
1942 )
1943
1944 (defun DTIN ( duzx duzy / duzx duzy x1 y1 )
1945   (setq x1 (+ DUZx (nth 0 DT_IN)))
1946   (setq y1 (+ DUZy (nth 1 DT_IN)))
1947   (setq DT_IN (list x1 y1 400))
1948   (setq DT_IN_S (list x1 y1 200))
1949 )
1950
1951 (defun KUIN ( duzx duzy / duzx duzy x1 y1 )
1952   (setq x1 (+ DUZx (nth 0 KU_IN)))
1953   (setq y1 (+ DUZy (nth 1 KU_IN)))
1954   (setq KU_IN (list x1 y1 400))
1955   (setq KU_IN_S (list x1 y1 200))
1956 )
1957 )
1958
1959 (defun SFIN ( duzx duzy / duzx duzy x1 y1 )
1960   (setq x1 (+ DUZx (nth 0 SF_IN)))
1961   (setq y1 (+ DUZy (nth 1 SF_IN)))
1962   (setq SF_IN (list x1 y1 400))
1963   (setq SF_IN_S (list x1 y1 200))
1964 )
1965
1966 (defun TSIN ( duzx duzy / duzx duzy x1 y1 )
1967   (setq x1 (+ DUZx (nth 0 TS_IN)))
1968   (setq y1 (+ DUZy (nth 1 TS_IN)))
1969   (setq TS_IN (list x1 y1 400))
1970   (setq TS_IN_S (list x1 y1 200))
1971 )
1972
1973 (defun boy_uyarla_1d ( L / L duzx duzy tmp1 )
1974   (setq duzx 0 duzy 0)
1975   (setq tmp1 (solve (cadr L)))
1976   (setq tmp1 (car (reverse tmp1)))
1977   (setq duzx (- boy bo1))
1978   (cond
1979     ((= tmp1 "LA") (LAIN duzx duzy))
1980     ((= tmp1 "KL") (KLIN duzx duzy))
1981     ((= tmp1 "CM") (CMIN duzx duzy))
1982     ((= tmp1 "DT") (DTIN duzx duzy))
1983     ((= tmp1 "KU") (KUIN duzx duzy))
1984     ((= tmp1 "SF") (SFIN duzx duzy))
1985     ((= tmp1 "TS") (TSIN duzx duzy))
1986   )
1987 )
1988
1989 (defun boy_uyarla_2d1 ( y / y duzx duzy tmp1 )
1990   (setq duzx 0 duzy 0)
1991   (setq duzy (- en en1))
1992   (setq tmp1 (car (reverse (solve y))))
1993   (cond
1994     ;ikinci duvar, son eleman
1995     ((= tmp1 "LA") (LAIN duzx duzy))
1996     ((= tmp1 "KL") (KLIN duzx duzy))
1997     ((= tmp1 "CM") (CMIN duzx duzy))
1998     ((= tmp1 "DT") (DTIN duzx duzy))
1999     ((= tmp1 "KU") (KUIN duzx duzy))

```

```

1999      ((= tmp1 "SF") (SFIN duzx duzy))
2000      ((= tmp1 "TS") (TSIN duzx duzy))
2001    )
2002    (setq duzx 0 duzy 0)
2003    (setq duzx (- boy bol))
2004    (foreach tmp1 (solve y)
2005      (cond ;ikinci duvar, tüm elemanlar
2006        ((= tmp1 "LA") (LAIN duzx duzy))
2007        ((= tmp1 "KL") (KLIN duzx duzy))
2008        ((= tmp1 "CM") (CMIN duzx duzy))
2009        ((= tmp1 "DT") (DTIN duzx duzy))
2010        ((= tmp1 "KU") (KUIN duzx duzy))
2011        ((= tmp1 "SF") (SFIN duzx duzy))
2012        ((= tmp1 "TS") (TSIN duzx duzy))
2013      )
2014    )
2015  )
2016
2017  (defun boy_uyarla_2d2 ( x y / x y duzx duzy tmp1 )
2018    (boy_uyarla_1d (list "" x)) ;ilk duvar, son eleman
2019    (setq duzx 0 duzy 0)
2020    (setq duzy (- en en1))
2021    (setq tmp1 (car (reverse (solve y))))
2022    (cond ;ikinci duvar, son eleman
2023      ((= tmp1 "LA") (LAIN duzx duzy))
2024      ((= tmp1 "KL") (KLIN duzx duzy))
2025      ((= tmp1 "CM") (CMIN duzx duzy))
2026      ((= tmp1 "DT") (DTIN duzx duzy))
2027      ((= tmp1 "KU") (KUIN duzx duzy))
2028      ((= tmp1 "SF") (SFIN duzx duzy))
2029      ((= tmp1 "TS") (TSIN duzx duzy))
2030    )
2031    (setq duzx 0 duzy 0)
2032    (setq duzx (- boy bol))
2033    (foreach tmp1 (solve y)
2034      (cond ;ikinci duvar, tüm elemanlar
2035        ((= tmp1 "LA") (LAIN duzx duzy))
2036        ((= tmp1 "KL") (KLIN duzx duzy))
2037        ((= tmp1 "CM") (CMIN duzx duzy))
2038        ((= tmp1 "DT") (DTIN duzx duzy))
2039        ((= tmp1 "KU") (KUIN duzx duzy))
2040        ((= tmp1 "SF") (SFIN duzx duzy))
2041        ((= tmp1 "TS") (TSIN duzx duzy))
2042      )
2043    )
2044  )
2045
2046  (defun boy_uyarla_2d ( L / x y L tmp1 )
2047    (setq tmp1 (solve_2d (cadr L)))
2048    (setq x (car tmp1))
2049    (setq y (cadr tmp1))
2050    (if (and (= x "") (/= y ""))
2051      (boy_uyarla_2d1 y)
2052    )
2053    (if (and (/= x "") (/= y ""))
2054      (boy_uyarla_2d2 x y)
2055    )
2056    (if (and (/= x "") (= y ""))
2057      (boy_uyarla_1d L)
2058    )
2059  )
2060
2061  (defun boyut_uyarla ( L / L tmp1 tmp2 duzx duzy )

```

```

2062 (setq BL_EN (/ (* (atoi (substr SKP 1 2)) 10) 2))
2063 (setq BL_BY (/ (* (atoi (substr SKP 4 2)) 10) 2))
2064 (setq EN (* BL_EN (+ (fix (/ EN BL_EN)) 1)))
2065 (setq BOY (* BL_BY (+ (fix (/ BOY BL_BY)) 1)))
2066 (if (or (/= boy bol) (/= en en1))
2067     (if (= dv "1")
2068         (boy_uyarla_1d L)
2069         (boy_uyarla_2d L)
2070     )
2071 )
2072 )
2073
2074 (defun golgeler ()
2075     (command "layer" "s" "SURFACE" "")
2076     (if(= LA1 1)(command "_insert" LAVS LA_IN_S "" ""
2077         la_angle))
2078     (if (= KL1 1) (command "_insert" KLOS KL_IN_S "" ""
2079         kl_angle))
2080     (if (= CM1 1) (command "_insert" "CM_S" CM_IN_S "" ""
2081         cm_angle))
2082     (if (and (= YK1 1) (= DT1 1))
2083         (command "_insert" "DT_S" DT_IN_S "" "" dt_angle))
2084     (if (and (= YK1 1) (= KU1 1))
2085         (command "_insert" "KU_S" KU_IN_S "" "" ku_angle))
2086     (if (and (= SS1 1) (= SF1 1))
2087         (command "_insert" "SF_S" SF_IN_S "" "" sf_angle))
2088     (if (and (= SS1 1) (= TS1 1))
2089         (command "_insert" "TS_S" TS_IN_S "" "" ts_angle))
2090 )
2091
2092 (defun gercekler ()
2093     (command "layer" "s" "VITRIFIYE" "")
2094     (if (= LA1 1)(command "_insert" LAV LA_IN "" "" la_angle))
2095     (if (= KL1 1)(command "_insert" KLO KL_IN "" "" kl_angle))
2096     (if (= CM1 1)(command "_insert" "CM" CM_IN "" "" cm_angle))
2097     (if (and (= YK1 1) (= DT1 1))
2098         (command "_insert" "DT" DT_IN "" "" dt_angle))
2099     (if (and (= YK1 1) (= KU1 1))
2100         (command "_insert" "KU" KU_IN "" "" ku_angle))
2101     (if (and (= SS1 1) (= SF1 1))
2102         (command "_insert" "SF" SF_IN "" "" sf_angle))
2103     (if (and (= SS1 1) (= TS1 1))
2104         (command "_insert" "TS" TS_IN "" "" ts_angle))
2105 )
2106
2107 (defun set_some_globals ( p1 / )
2108     (setq la_angle 0
2109         kl_angle 0
2110         cm_angle 0
2111         dt_angle 0
2112         ku_angle 0
2113         sf_angle 0
2114         ts_angle 0
2115     )
2116     (setq la_in p1
2117         la_in_s p1
2118         kl_in p1
2119         kl_in_s p1
2120         cm_in p1
2121         cm_in_s p1
2122         dt_in p1
2123         dt_in_s p1
2124         ku_in p1

```

```

2122      ku_in_s p1
2123      sf_in    p1
2124      sf_in_s  p1
2125      ts_in    p1
2126      ts_in_s  p1
2127    )
2128  )
2129
2130  (defun duvarlar ( p1 / a0 a1 a2 a3
2131                  p1 p2 p3 p4 p5 p6 p7 p8
2132                  p9 p10 p11 p12 p13 p14 p15 p16 )
2133
2134      (setq A0 0 ;; 0
2135            A1 ( / pi 2 ) ;; 90
2136            A2 pi ;; 180
2137            A3 (* 3 ( / pi 2 ) ) ;; 270
2138            P2 (polar P1 A0 BOY)
2139            P3 (polar P2 A1 EN)
2140            P4 (polar P3 A2 BOY)
2141            P5 (polar P1 A3 ( + CC ID))
2142            P6 (polar P5 A1 CC)
2143            P7 (polar P6 A0 BOY)
2144            P8 (polar P7 A3 CC)
2145            P9 (polar P8 A0 DD)
2146            P10 (polar P9 A1 ( + CC ID EN ID CC))
2147            P11 (polar P10 A2 DD)
2148            P12 (polar P11 A3 CC)
2149            P13 (polar P12 A2 ( + BOY ID CC))
2150            P14 (polar P13 A3 ID)
2151            P15 (polar P14 A0 CC)
2152            P16 (polar P15 A3 ( + EN ID CC))
2153      )
2154      (command "_layer" "_s" "DUVAR" "")
2155      (command "_line" P1 P2 P3 P4 "c")
2156      (command "_line" P5 P6 P7 P8 "")
2157      (command "_line" P9 P10 "")
2158      (command "_line" P11 P12 P13 "")
2159      (command "_line" P14 P15 P16 "")
2160  )
2161
2162  (defun kapi ( p1 id kp kk L / p1 p2 p3 p4 id kp a0 a1 a2 a3
2163              p3 p4 L pk1 pk2 pk3 pk4 m )
2164      (setq m (car (reverse (solve (cadr (solve_2d (cadr L)))))))
2165      ;;2. duvarin son elemani KU ise kapiyi KU kadar kaydir
2166      (if (and (= m "KU") (= dv "2"))
2167          (setq p1 (list (+ (car p1) 0.0) (- (cadr p1) 700.0) ))
2168      )
2169      (setq A0 0 ;; 0
2170            A1 ( / pi 2 ) ;; 90
2171            A2 pi ;; 180
2172            A3 (* 3 ( / pi 2 ) ) ;; 270
2173            P2 (polar P1 A0 BOY)
2174            P3 (polar P2 A1 EN)
2175            P4 (polar P3 A2 BOY)
2176            PK1 (polar P4 A3 ID)
2177            PK2 (polar PK1 A2 ID)
2178            PK3 (polar PK2 A3 KP)
2179            PK4 (polar PK3 A0 ID)
2180            KK1 (polar PK1 A3 KK) ; Kapi
2181            KK2 (polar KK1 A0 KP)
2182            KK3 (polar KK2 A1 KK)
2183      )
2184      (command "_break" PK1 PK4) ;kapi araligi

```

```

2185 (command "_break" PK2 PK3)
2186 (command "_line" PK1 PK2 "")
2187 (command "_line" PK3 PK4 "")
2188 (command "_layer" "s" "KAPI_PENCERE" "")
2189 (command "_line" PK1 KK1 KK2_KK3 "_c") ; Kapi
2190 (command "_arc" "_c" PK1 PK4 KK2)
2191 )
2192
2193 (defun pencere ( p1 / bp0 bp1 bp3 bp3 bp4 bp5 bp6 bp7
2194 a0 a1 a2 a3 pp1 pp2 pp3 pp4
2195 p1 p2 p3)
2196 (setq A0 0 ;;; 0
2197 A1 ( / pi 2 ) ;;; 90
2198 A2 pi ;;; 180
2199 A3 (* 3 ( / pi 2 ) ) ;;; 270
2200 P2 (polar P1 A0 BOY)
2201 P3 (polar P2 A1 EN)
2202 PP1 (polar P3 A3 DD) ; Pencere araliGi
2203 PP2 (polar PP1 A0 DD)
2204 PP3 (polar PP2 A3 PP)
2205 PP4 (polar PP3 A2 DD)
2206 BP1 (polar PP1 A0 ( / ( - DD PK) 2)) ; Pencere
2207 BP2 (polar BP1 A0 PK)
2208 BP3 (polar BP2 A3 PP)
2209 BP4 (polar BP3 A2 PK)
2210 BP5 (polar BP1 A3 PK)
2211 BP8 (polar BP5 A0 PK)
2212 BP6 (polar BP4 A1 PK)
2213 BP9 (polar BP6 A0 PK)
2214 BP7 (polar BP6 A0 ( / PK 2))
2215 BP0 (polar BP7 A1 ( - PP ( * PK 2)))
2216 )
2217 (command "_layer" "s" "DUVAR" "")
2218 (command "_line" PP1 PP2 "") ; Pencere araliGi
2219 (command "_line" PP3 PP4 "")
2220 (command "_layer" "s" "KAPI_PENCERE" "")
2221 (command "_line" BP1 BP2 BP3 BP4 "_c") ; Pencere
2222 (command "_line" BP5 BP8 "")
2223 (command "_line" BP6 BP9 "")
2224 (command "_line" BP7 BP0 "")
2225 )
2226
2227 (defun son_kontrol ( L / m1 m2 )
2228 ;;2. duvarin son elemani KU ise ...
2229 (setq m1(car(reverse(solve(cadr(solve_2d(cadr L)))))))
2230 ;;1. duvarda eleman varsa ...
2231 (setq m2 (solve (car (solve_2d (cadr L)))))
2232 (if (and (= m1 "KU") (/= m2 "") (= dv "2") (< en 2200))
2233 (setq en 2200)
2234 )
2235 ;;1. duvarin son elemani KU ise ...
2236 (setq m1 (car (reverse (solve (cadr L)))))
2237 (if (and (= dv "1") (= m1 "KU") (< en 1700))
2238 (setq EN 1700)
2239 )
2240 )
2241
2242 (defun MAIN ( L sx sy m / lav lavs klo klos L sx sy m
2243 p1 x1 y1 scale
2244 cc id dd kp kk pp pk )
2245 (setq LAV (nth 2 (car DETAY)))
2246 (setq LAVS (strcat LAV "S"))
2247 (setq KLO (nth 2 (cadr DETAY)))

```

```

2248 (setq KLOS (strcat KLO "_S"))
2249 (setq ang 0)
2250 (setq CC 300
2251      ID 100      ; ic duvar
2252      DD 200      ; Dis duvar
2253      KP 800      ; Kapi araligi
2254      KK 50       ; Kapi kalınligi
2255      PP 500      ; Pencere araligi
2256      PK 50       ; Pencere kalınligi
2257      EN1 1700
2258      BO1 3000
2259 )
2260 ;; Çizimin sifir noktası
2261 (setq P1 (list (+ SX ID CC 450) (+ SY ID CC 100)))
2262 (set_some_globals p1)
2263 (cond
2264   ((= DV "1") (BOYUTLA_1D (cadr L) ang P1))
2265   ((= DV "2") (BOYUTLA_2D (cadr L) P1))
2266 )
2267 (setq EN EN1 BOY BO1)
2268 (if (< en 1700) (setq en 1700))
2269 (if (< boy 1700) (setq boy 1700))
2270 (son_kontrol L)
2271 (boyut_uyarla l)
2272 (setq X1 (+ SX (/ (+ BOY (* CC 3.5))2))
2273      Y1 (+ SY (/ (+ EN (* CC 2))2))
2274      SCALE (+ EN (* CC 7))
2275 )
2276 (command "_zoom" "c" (list X1 Y1) SCALE)
2277 (duvarlar p1)
2278 (kapi p1 id kp kk L)
2279 (pencere p1)
2280 (doseme p1)
2281 (golgeler)
2282 (gercekler)
2283 (olculer_yazilar p1 m L)
2284 )
2285
2286 (defun olculer_yazilar ( p1 m L / xx1 yy1 dx1 dy1 dp11 dp12
2287                        dp13 p1 L dp21 dp22 dp23 tp1 tp2 tp3 m m2 ml)
2288
2289   (command "_layer" "_s" "OLCU" "") ; Ölçü
2290   (setq XX1 (car P1)
2291         YY1 (cadr P1)
2292         DX1 (- XX1 (+ 100 (/ BOY 19.0 ) ) )
2293         DY1 (+ YY1 EN (+ 100 (/ EN 18.0) ) )
2294         DP11 (list DX1 YY1)
2295         DP12 (list DX1 (+ YY1 EN ) )
2296         DP13 (list (- DX1 300) YY1)
2297         DP21 (list XX1 DY1)
2298         DP22 (list (+ XX1 BOY ) DY1)
2299         DP23 (list XX1 (+ DY1 250)) )
2300   (command "dim" "ver" DP11 DP12 DP13 "" )
2301   (command "hor" DP21 DP22 DP23 "" "exit")
2302
2303   (command "_layer" "_s" "YAZI" "" ) ; Yazı
2304   (setq TP1 (list (+ XX1 220) (+ YY1 250 EN ) )
2305         TP2 (list (+ XX1 50) (- YY1 700))
2306         TP3 (list (+ XX1 50) (- YY1 1100))
2307   )
2308   (setq M2 (rtos (* (/ EN 1000.00) (/ BOY 1000.00)) 2 2))
2309   (command "_text" TP1 "" (strcat SVT " ; " SKP " " M2 " m2"))
2310   (command "_text" TP2 ""

```

```

2311      (strcat "Sira " (itoa M) " " (cadr L)
2312      "      Hata " (itoa (car L))
2313      )
2314      )
2315      (setq ml (GET_MLYT L KP PP))
2316      (setq ml (noktali ML))
2317      (command "_text" tp3 "" (strcat "Toplam Maliyet " ml " TL"))
2318      (command "_layer" "_s" "VITRIFIYE" "")
2319      )
2320
2321      (defun noktali ( m / y m )
2322      (setq y (strlen m))
2323      (if (= y 7) (setq m (strcat (substr m 1 1) "."
2324      (substr m 2 3) "."
2325      (substr m 5 3)
2326      )
2327      )
2328      )
2329      (if (= y 8) (setq m (strcat (substr m 1 2) "."
2330      (substr m 3 3) "."
2331      (substr m 6 3)
2332      )
2333      )
2334      )
2335      (if (= y 9) (setq m (strcat (substr m 1 3) "."
2336      (substr m 4 3) "."
2337      (substr m 7 3)
2338      )
2339      )
2340      )
2341      (if (= y 10) (setq m (strcat (substr m 1 1) "."
2342      (substr m 2 3) "."
2343      (substr m 5 3) "."
2344      (substr m 8 3)
2345      )
2346      )
2347      )
2348      (eval m)
2349      )
2350
2351      (defun GET_MLYT ( L KP PP / L lp kk ml x x1 x2 y )
2352      (setq ml 0
2353      x (cadr L)
2354      y nil
2355      )
2356      (if (= dv "1")
2357      (setq y (solve x))
2358      (setq x1 (solve (car (solve_2d x)))
2359      x2 (solve (cadr (solve_2d x)))
2360      y (append x1 x2)
2361      )
2362      )
2363      (foreach lm y
2364      (cond
2365      ((= LM "LA") (setq ML (+ ML (D 0 7)))) )
2366      ((= LM "KL") (setq ML (+ ML (D 1 7)))) )
2367      ((= LM "CM") (setq ML (+ ML (D 2 7)))) )
2368      ((= LM "DT") (setq ML (+ ML (D 3 7)))) )
2369      ((= LM "KU") (setq ML (+ ML (D 4 7)))) )
2370      ((= LM "SF") (setq ML (+ ML (D 5 7)))) )
2371      ((= LM "TS") (setq ML (+ ML (D 6 7)))) )
2372      )
2373      )

```

```

2374 (setq yer_alani (* EN BOY 0.000001))
2375 (setq yer_mali (* yer_alani (D 7 7)))
2376 (setq ML (+ ML yer_mali))
2377 (setq duvar_alani (* (+ EN BOY) (D 9 6) 0.000002))
2378 (setq kapi_alani (* KP (D 9 4) 0.000001))
2379 (setq pencere_alani (* PP (D 9 5) 0.000001))
2380 (setq duvar_alani (- duvar_alani kapi_alani pencere_alani))
2381 (setq duvar_mali (* duvar_alani (D 8 7)))
2382 (setq ML (+ ML duvar_mali))
2383 (setq tavan_mali (* yer_alani (D 9 7)))
2384 (setq ML (+ ML tavan_mali))
2385 (eval (rtos ML 2 0))
2386 )
2387
2388 (defun get_soru ( L m / L m s1 s2 s3 s4 )
2389 (setq s1 (itoa (car L)))
2390 (setq s2 (cadr L))
2391 (setq s3 (strcat s1 " " s2))
2392 (setq s5 (strcat (itoa (length final)) "/" (itoa m)))
2393 (setq s4 (strcat "\n" s5 "Sonraki ? " s3 " <E/h/b/c> :"))
2394 (eval s4)
2395 )
2396
2397 (defun DRAW_IT ( / ans soru m x )
2398 (setq SX 9000) ; Baslangic noktasi X
2399 (setq SY 2000) ; Baslangic noktasi Y
2400 (setq M 0)
2401 (setq x (- (length final) 1))
2402 (foreach L FINAL
2403 (setq soru (get_soru L (+ 1 m)))
2404 (if (<= m x) (setq ANS (getstring soru)))
2405 (if (or (= ans "c") (= ans "C")) (exit))
2406 (if (or (= ans "b") (= ans "B")) (setq m (+ x 1)))
2407 (if (and(<= m x) (or(= ANS "E") (= ANS "e") (= ANS ""))))
2408 (progn
2409 (MAIN L SX SY (+ 1 M))
2410 (command "hide")
2411 (if (= (rem (+ 1 M) 4) 0)
2412 (setq SX 9000 SY (- SY 7000)) ;satirbasi
2413 (setq SX (+ SX 7000)) ;bir saga
2414 )
2415 )
2416 )
2417 (setq M (+ M 1))
2418 )
2419 (getstring "")
2420 (command "_zoom" "e")
2421 (if (<= (length FINAL) 2) (command "_hide"))
2422 )
2423
2424 (defun CLR_SCR ( x / x)
2425 (command "_zoom" "_a")
2426 (command "_erase" x "")
2427 (setq x nil)
2428 )
2429
2430 (defun ASK_SCR ( / CLR )
2431 (setq DCL_ID (load_dialog "Banyo.DCL"))
2432 (if (not (new_dialog "DDbanyo5" DCL_ID)) (exit))
2433 (action_tile "DC4_TM" "(done_dialog) (setq CLR 1)" )
2434 (action_tile "DC4_IP" "(done_dialog) (exit)" )
2435 (start_dialog)
2436 (unload_dialog DCL_ID)

```

```

2437 (done_dialog)
2438 (eval CLR)
2439 )
2440
2441 (defun CHK_SCR ( / XX x)
2442 (setq XX (ssget "x"))
2443 (if (= acad_ver "12")
2444 (setq x 8)
2445 (setq x 14)
2446 )
2447 (if (= (if (> (sslength XX) x) (ASK_SCR)) 1) (CLR_SCR xx))
2448 (setq XX nil)
2449 )
2450
2451 (defun C1 ( / L temp)
2452 (setq temp nil)
2453 (foreach L FINAL
2454 (if (<= (nth 0 L) (atoi TOL)) (setq temp (cons L temp))))
2455 )
2456 (setq FINAL (reverse temp))
2457 )
2458
2459 (defun C3 ( / L temp X)
2460 (setq temp nil)
2461 (setq X 1)
2462 (foreach L FINAL
2463 (if (and (<= (nth 0 L) (atoi TOL)) (<= X CZS))
2464 (setq temp (cons L temp))
2465 )
2466 (setq X (+ X 1))
2467 )
2468 (setq FINAL (reverse temp))
2469 )
2470
2471 (defun CUT_FINAL ()
2472 (cond
2473 ((= CZT 1) (C1))
2474 ((= CZT 3) (C3))
2475 )
2476 )
2477
2478 (defun ekran_6 ()
2479 (setq DCL_ID (load_dialog "Banyo.DCL"))
2480 (if (not (new_dialog "DDBanyo6" DCL_ID)) (exit))
2481
2482 (set_tile "DC6_TO2" tol)
2483 (set_tile "DC6_AL2" (cadr (car final)))
2484 (set_tile "DC6_AP2" (itoa (car (car final))))
2485
2486 (action_tile "DC6_IP" "(done_dialog)" )
2487 (start_dialog)
2488 (unload_dialog DCL_ID)
2489 )
2490
2491 (defun mode_tile_7 ( / a)
2492 (setq a (atoi (get_tile "DC7_1D"))))
2493 (mode_tile "DC7_10" (- 1 a))
2494 (mode_tile "DC7_11" (- 1 a))
2495 (mode_tile "DC7_12" (- 1 a))
2496 (mode_tile "DC7_13" (- 1 a))
2497 (mode_tile "DC7_14" (- 1 a))
2498 (mode_tile "DC7_15" (- 1 a))
2499 (mode_tile "DC7_16" (- 1 a))

```

```

2500 (mode_tile "DC7_17" (- 1 a))
2501 (mode_tile "DC7_18" (- 1 a))
2502 (mode_tile "DC7_19" (- 1 a))
2503 (mode_tile "DC7_20" a)
2504 (mode_tile "DC7_21" a)
2505 (mode_tile "DC7_22" a)
2506 (mode_tile "DC7_23" a)
2507 (mode_tile "DC7_24" a)
2508 (mode_tile "DC7_25" a)
2509 (mode_tile "DC7_26" a)
2510 (mode_tile "DC7_27" a)
2511 )
2512
2513 (defun goto_first_line ( / x m )
2514   (setq x 0)
2515   (while (/= (substr (nth x fl-lst) 1 1) "+")
2516     (setq x (+ 1 x)))
2517   )
2518   (set_tile "DC7_LS" (itoa (+ 5 x)))
2519 )
2520
2521 (defun fill_list_box ( / fl )
2522   (if (null (setq fl (open (strcat (getvar "dwgprefix")
2523                                     "Banyo.krx") "r"))))
2524     (progn
2525       (princ "\n\"Banyo.krx\" dosyasi açilamadi!")
2526       (exit)
2527     )
2528     (setq fl-lst nil)
2529     (start_list "DC7_LS")
2530     (while (setq line (read-line fl))
2531       (setq fl-lst (cons line fl-lst))
2532       (add_list line)
2533     )
2534     (end_list)
2535     (setq fl-lst (reverse fl-lst))
2536     (close fl)
2537   )
2538
2539 (defun do_it ( ll a b / ll a b c d x )
2540   (setq a (strcat b (substr a 2 (- (strlen a) 1) )))
2541   (setq x 0 c nil)
2542   (foreach c fl-lst
2543     (if (/= x ll)
2544       (setq d (cons c d))
2545       (setq d (cons a d))
2546     )
2547     (setq x (+ 1 x))
2548   )
2549   (setq fl-lst (reverse d))
2550   (write_rule)
2551   (set_tile "DC7_LS" (itoa ll))
2552 )
2553
2554 (defun write_rule ( / c )
2555   (if (null (setq fl (open (strcat (getvar
2556                                     "dwgprefix") "Banyo.krx") "w"))))
2557     (progn
2558       (princ "\n\"Banyo.krx\" dosyasi açilamadi!")
2559       (exit)
2560     )

```

```

2561 (start_list "DC7_LS")
2562 (foreach c fl-lst
2563   (write-line c fl)
2564   (add_list c)
2565 )
2566 (end_list)
2567 (close fl)
2568 )
2569
2570 (defun change_mode ( / ll a )
2571   (setq ll (atoi (get_tile "DC7_LS")))
2572   (setq a (nth ll fl-lst))
2573   (cond
2574     ((= (substr a 1 1) "+") (do_it ll a "-"))
2575     ((= (substr a 1 1) "-") (do_it ll a "+"))
2576   )
2577   (mode_tile "DC7_LS" 2)
2578 )
2579
2580 (defun delete_it ( ll a / ll a c d x )
2581   (setq x 0 c nil)
2582   (foreach c fl-lst
2583     (if (/= x ll) (setq d (cons c d)))
2584     (setq x (+ 1 x))
2585   )
2586   (setq fl-lst (reverse d))
2587   (if (= (substr a 8 1) "1")
2588     (sort_1d_rule)
2589     (sort_2d_rule)
2590   )
2591   (done_dialog)
2592   (write_rule)
2593   (set_tile "DC7_LS" (itoa ll))
2594 )
2595
2596 (defun get_p ( y / a x y z )
2597 ; y 1 birinci bolumun ilk elemaninin numarasi
2598 ; y 2 birinci bolumun son elemaninin numarasi
2599 ; y 3 ikinci bolumun ilk elemaninin numarasi
2600 ; y 4 ikinci bolumun son elemaninin numarasi
2601 (setq x 0)
2602 (if (or (= y 2) (= y 4))
2603   (setq a (reverse fl-lst))
2604   (setq a fl-lst)
2605 )
2606 (if (or (= y 1) (= y 2))
2607   (setq z "1")
2608   (setq z "2")
2609 )
2610 (while (and (or (/=(substr (nth x a) 1 1) "+")
2611                (/=(substr (nth x a) 1 1) "-")
2612              )
2613          (/=(substr (nth x a) 8 1) z)
2614        )
2615   (setq x (+ 1 x))
2616 )
2617 (if (or (= y 2) (= y 4)) (setq x (- (length fl-lst) x 1)))
2618 (eval x)
2619 )
2620
2621 (defun get_part1 ( / x y p)
2622 (setq x 0 y (- (get_p 1) 1) p nil)
2623 (while (<= x y)

```

```

2624      (setq p (cons (nth x fl-lst) p))
2625      (setq x (+ 1 x))
2626    )
2627    (reverse p)
2628  )
2629
2630  (defun get_part2 ( / x y p)
2631    (setq x (get_p 1) y (get_p 2) p nil)
2632    (while (<= x y)
2633      (if (/= (nth x fl-lst) "")
2634        (setq p (cons (nth x fl-lst) p))
2635      )
2636      (setq x (+ 1 x))
2637    )
2638    (reverse p)
2639  )
2640
2641  (defun get_part123 ( / x y p)
2642    (setq x 0 y (- (get_p 3) 1) p nil)
2643    (while (<= x y)
2644      (setq p (cons (nth x fl-lst) p))
2645      (setq x (+ 1 x))
2646    )
2647    (reverse p)
2648  )
2649
2650  (defun get_part4 ( / x y p)
2651    (setq x (get_p 3) y (get_p 4) p nil)
2652    (while (<= x y)
2653      (if (/= (nth x fl-lst) "")
2654        (setq p (cons (nth x fl-lst) p))
2655      )
2656      (setq x (+ 1 x))
2657    )
2658    (reverse p)
2659  )
2660
2661  (defun get_part5 ( / x y p)
2662    (setq x (+ (get_p 4) 1) y (- (length fl-lst) 1) p nil)
2663    (while (<= x y)
2664      (setq p (cons (nth x fl-lst) p))
2665      (setq x (+ 1 x))
2666    )
2667    (reverse p)
2668  )
2669
2670  (defun get_part345 ( / x y p)
2671    (setq x (+ (get_p 2) 1) y (- (length fl-lst) 1) p nil)
2672    (while (<= x y)
2673      (setq p (cons (nth x fl-lst) p))
2674      (setq x (+ 1 x))
2675    )
2676    (reverse p)
2677  )
2678
2679  (defun sort_ld_rule ( / L n a r2 m ll)
2680    (setq r2 (get_part2))
2681    (foreach a r2
2682      (setq n (cons
2683        (strcat (substr a ll 2)
2684          (substr a 23 1)
2685          (substr a 14 2)
2686          (substr a 17 2)

```

```

2687             (substr a 28 1)
2688             (substr a 1 1)
2689         ) n
2690     )
2691 )
2692 )
2693 (setq m (acad_strlsort n))
2694 (setq n nil a nil L 0)
2695 (setq l1 (substr (car m) 3 1))
2696 (foreach a m
2697   (setq L (+ 1 L))
2698   (if (= (substr a 3 1) l1)
2699     (setq n (cons
2700       (strcat (substr a 9 1) " "
2701         (get_num L) " "
2702         "1D "
2703         (substr a 1 2) " "
2704         (substr a 4 2)
2705         "="
2706         (substr a 6 2)
2707         " AD="
2708         (substr a 3 1)
2709         " => "
2710         (substr a 8 1)
2711       ) n
2712     )
2713   )
2714   (setq n (cons "" n)
2715     n (cons
2716       (strcat (substr a 9 1) " "
2717         (get_num L) " "
2718         "1D "
2719         (substr a 1 2) " "
2720         (substr a 4 2)
2721         "="
2722         (substr a 6 2)
2723         " AD="
2724         (substr a 3 1)
2725         " => "
2726         (substr a 8 1)
2727       ) n
2728     )
2729   )
2730 )
2731 (setq l1 (substr a 3 1))
2732 )
2733 (setq fl-lst (append (get_part1) (reverse n) (get_part345)))
2734 (sort_2d_rule)
2735 )
2736
2737 (defun get_last ( / a b l x )
2738   (setq a (reverse fl-lst))
2739   (setq x 0)
2740   (while (and (or (/=(substr (nth x a) 1 1) "+")
2741     (/=(substr (nth x a) 1 1) "-")
2742   )
2743     (/=(substr (nth x a) 8 1) "1")
2744   )
2745     (setq x (+ 1 x))
2746   )
2747   (setq l (atoi (substr (nth x a) 3 3)))
2748   (eval l)
2749 )

```

```

2750
2751 (defun sort_2d_rule ( / L n a r2 m l1)
2752   (setq L (get_last))
2753   (setq n nil)
2754   (setq r2 (get_part4))
2755   (foreach a r2
2756     (setq n (cons
2757       (strcat (substr a 14 2)
2758         (substr a 23 2)
2759         (substr a 29 1)
2760         (substr a 1 1)
2761       ) n
2762     )
2763   )
2764 )
2765 (setq m (acad_strlsort n))
2766 (setq n nil a nil)
2767 (setq l1 (substr (car m) 1 2))
2768 (foreach a m
2769   (setq L (+ 1 L))
2770   (if (= (substr a 1 2) l1)
2771     (setq n (cons
2772       (strcat (substr a 6 1) " "
2773         (get_num L) " "
2774         "2D Ax="
2775         (substr a 1 2)
2776         " ve Ay="
2777         (substr a 3 2)
2778         " => "
2779         (substr a 5 1)
2780       ) n
2781     )
2782   )
2783   (setq n (cons "" n)
2784     n (cons
2785       (strcat (substr a 6 1) " "
2786         (get_num L) " "
2787         "2D Ax="
2788         (substr a 1 2)
2789         " ve Ay="
2790         (substr a 3 2)
2791         " => "
2792         (substr a 5 1)
2793       ) n
2794     )
2795   )
2796 )
2797 (setq l1 (substr a 1 2))
2798 )
2799 (setq fl-lst(append(get_part123)(reverse n)(get_part5)))
2800 )
2801
2802 (defun get_num (x / a)
2803   (setq a (itoa x))
2804   (cond
2805     ((= (strlen a) 1) (setq a (strcat "00" a)))
2806     ((= (strlen a) 2) (setq a (strcat "0" a)))
2807   )
2808   (eval a)
2809 )
2810
2811 (defun ask_it ( l1 a / l1 a )
2812   (setq dcl_id (load_dialog "Banyo.DCL"))

```

```

2813 (if (not (new_dialog "DDbanyo8" dcl_id)) (exit))
2814 (set_tile "DC8_T1" a)
2815
2816 (action_tile "DC8_EV" "(delete_it 11 a)")
2817 (action_tile "DC8_HY" "(done_dialog)")
2818
2819 (start_dialog)
2820 (unload_dialog dcl_id)
2821 )
2822
2823 (defun delete_rule ( / 11 dcl_id a )
2824 (setq 11 (atoi (get_tile "DC7_LS")))
2825 (setq a (nth 11 fl-lst))
2826 (if (or (= (substr a 1 1) "+") (= (substr a 1 1) "+"))
2827 (ask_it 11 a)
2828 )
2829 )
2830
2831 (defun create_backup ( / )
2832 (if (null (setq fl (open (strcat (getvar
                                "dwgprefix") "Banyo.krl") "r")))
2833 (progn
2834 (princ "\n\"Banyo.krl\" dosyasi açilamadi!")
2835 (exit)
2836 )
2837 )
2838 (if (null (setq fx (open (strcat (getvar
                                "dwgprefix") "Banyo.krx") "w")))
2839 (progn
2840 (princ "\n\"Banyo.krx\" dosyasi açilamadi!")
2841 (exit)
2842 )
2843 )
2844 (while (setq line (read-line fl)) (write-line line fx))
2845 (close fl)
2846 (close fx)
2847 )
2848
2849 (defun add_rule_ld ( / a a1 a2 a3 a5 a9 12 13 15 x st)
2850 (setq 12 (list "2E" "3E" "4E" "5E"))
2851 (setq 13 (list "A1" "A2" "A3" "A4" "A5"))
2852 (setq 15 (list "CM" "DT" "KL" "KU" "LA" "SF" "TS"))
2853 (setq a2 (nth (atoi (get_tile "DC7_12")) 12))
2854 (setq a3 (nth (atoi (get_tile "DC7_13")) 13))
2855 (setq a5 (nth (atoi (get_tile "DC7_15")) 15))
2856 (setq a7 (itoa (+ (atoi (get_tile "DC7_17")) 1)))
2857 (setq a9 (itoa (+ (atoi (get_tile "DC7_19")) 1)))
2858 (setq a (strcat "1D " a2 " " a3 "=" a5 " AD=" a7 " => " a9))
2859 (setq a1 (strcat "1D " a2 " " a3 "=" a5 " AD=" a7 " => "))
2860 (setq x -1 st 0)
2861 (foreach m fl-lst
2862 (setq x (+ 1 x))
2863 (if (= a1 (substr m 8 20)) (setq st x))
2864 )
2865 (if (= st 0)
2866 (add_rule_ld1 a 1 0)
2867 (change_rule a st 0)
2868 )
2869 )
2870
2871 (defun change_rule ( a st tx / a x y n m st tx)
2872 (setq y -1 m nil)
2873 (foreach n fl-lst

```

```

2874      (setq y (+ 1 y))
2875      (if (/= st y)
2876          (setq m (cons n m))
2877          (setq m (cons (strcat "+"      " a) m))
2878      )
2879  )
2880  (setq fl-lst (reverse m))
2881  (sort_ld_rule)
2882  (write_rule)
2883  (setq x -1 st 0)
2884  (foreach n fl-lst
2885      (setq x (+ 1 x))
2886      (if (= a (substr n 8 (+ 21 tx))) (setq st x))
2887  )
2888  (set_tile "DC7_LS" (itoa st))
2889  (mode_tile "DC7_LS" 2)
2890  )
2891
2892  (defun add_rule_ldl ( a p tx / a x y n m p st tx)
2893      (setq x (get_p p) y -1 m nil)
2894      (foreach n fl-lst
2895          (setq y (+ 1 y))
2896          (if (/= x y)
2897              (setq m (cons n m))
2898              (setq m (cons n m) m (cons (strcat "+"      " a) m))
2899          )
2900      )
2901      (setq fl-lst (reverse m))
2902      (sort_ld_rule)
2903      (write_rule)
2904      (setq x -1 st 0)
2905      (foreach n fl-lst
2906          (setq x (+ 1 x))
2907          (if (= a (substr n 8 (+ 21 tx))) (setq st x))
2908      )
2909      (set_tile "DC7_LS" (itoa st))
2910      (mode_tile "DC7_LS" 2)
2911  )
2912
2913  (defun add_rule_2d ( / a a1 a3 a5 l3 x st)
2914      (setq l3 (list "CM" "DT" "KL" "KU" "LA" "SF" "TS"))
2915      (setq a3 (nth (atoi (get_tile "DC7_23")) l3))
2916      (setq a5 (nth (atoi (get_tile "DC7_25")) l3))
2917      (setq a7 (itoa (+ (atoi (get_tile "DC7_27")) 1)))
2918      (setq a (strcat "2D Ax=" a3 " ve Ay=" a5 " => " a7))
2919      (setq a1 (strcat "2D Ax=" a3 " ve Ay=" a5 " => "))
2920      (setq x -1 st 0)
2921      (foreach m fl-lst
2922          (setq x (+ 1 x))
2923          (if (= a1 (substr m 8 21)) (setq st x))
2924      )
2925      (if (= st 0)
2926          (add_rule_ldl a 3 1)
2927          (change_rule a st 1)
2928      )
2929  )
2930
2931  (defun add_rule ()
2932      (if (= (get_tile "DC7_1D") "1")
2933          (add_rule_ld)
2934          (add_rule_2d)
2935      )
2936  )

```

```

2937
2938 (defun update_krl ( / )
2939   (if (null (setq fl(open(strcat(getvar
                                "dwgprefix") "Banyo.krl") "w"))))
2940     (progn
2941       (princ "\n\"Banyo.krl\" dosyasi açilamadi!")
2942       (exit)
2943     )
2944   )
2945   (if (null (setq fx(open(strcat(getvar
                                "dwgprefix") "Banyo.krx") "r"))))
2946     (progn
2947       (princ "\n\"Banyo.krx\" dosyasi açilamadi!")
2948       (exit)
2949     )
2950   )
2951   (while (setq line (read-line fx)) (write-line line fl))
2952   (close fl)
2953   (close fx)
2954 )
2955
2956 (defun rule_base ( / dcl_id )
2957   (setq dcl_id (load_dialog "Banyo.DCL"))
2958   (if (not (new_dialog "DDbanyo7" dcl_id)) (exit))
2959   (create_backup)
2960   (set_tile "DC7_1D" "1")
2961   (mode_tile 7)
2962   (fill_list_box)
2963   (goto_first_line)
2964   (mode_tile "DC7_LS" 2)
2965
2966   (action_tile "DC7_EK" "(add_rule)")
2967   (action_tile "DC7_MD" "(change_mode)")
2968   (action_tile "DC7_SI" "(delete_rule) (mode_tile \"DC7_LS\" 2)")
2969   (action_tile "DC7_TM" "(done_dialog) (update_krl)")
2970   (action_tile "DC7_IP" "(done_dialog)")
2971   (action_tile "DC7_1D" "(mode_tile 7)")
2972   (action_tile "DC7_2D" "(mode_tile 7)")
2973
2974   (start_dialog)
2975   (unload_dialog dcl_id)
2976 )
2977
2978 (defun c:banyo ( / sor ans )
2979   (GET_FIRMA)
2980   (setvar "CMDECHO" 0)
2981   (setvar "BLIPMODE" 0)
2982   (if *error* (setq tp_oe *error* *error* tp_err))
2983   (setq ans "E")
2984   (setq ILK "evet")
2985   (while (or (= ans "E") (= ans "e") (= ans ""))
2986     (CHK_SCR)
2987     (TP_DIALOG1)
2988     (pre_exam dc_pop)
2989     (setq sor "1")
2990     (if (> (car (car final)) (atoi tol))
2991       (progn
2992         (ekran_6)
2993         (setq sor "0")
2994       )
2995     )
2996     (progn
2997       (GET_DETAIL)
2998       (CUT_FINAL)

```

```
2998          (DRAW_IT)
2999          (setq sor "1")
3000      )
3001  )
3002  (setq FINAL nil)
3003  (setq ILK "hayir")
3004  (if(= sor "1") (setq ans (getstring "\n Devam ?<E/h> :")))
3005  (princ)
3006  )
3007  (setvar "BLIPMODE" 1 )
3008  (setvar "CMDECHO" 1 )
3009  (princ)
3010  )
```



EK C Programlanabilir Diyalog Kutuları

AutoCAD programı bu özelliği 12. sürümünden itibaren kullanıma sunmuştur. Bu özellik programcıyla kullanıcı arasında daha "kullanıcı dostu" bir diyalog kurulabilmesi imkanını sağlamaktadır. Bu çalışmada bu özellik kullanılmıştır. Aşağıda dökümü verilen kod 8 adet diyalog penceresi içermektedir.

```
dcl_settings : default_dcl_settings { audit_level = 0; }
```

Ana ekran (birinci ekran):

```
DDbanyo1 : dialog
{key = "DC1_LABEL";
  :row {
    : column {
      : boxed_row {
        label = "Elemanlar";
        fixed_height = true;
        alignment = top;
        : column {
          fixed_width = true;
          : toggle {
            label = "Lavabo";
            mnemonic="L";
            key = "DC_LA";
          }
          : toggle {
            label = "Camasir Mak.";
            mnemonic="a";
            key = "DC_CM";
          }
          : toggle {
            label = "Sicak Su";
            mnemonic="u";
            key = "DC_SS";
          }
        }
      }
    }
    : column {
      fixed_width = true;
      : toggle {
        label = "Klozet";
        mnemonic="o";
        key = "DC_KL";
      }
      : toggle {
        label = "Yikanma";
        mnemonic="n";
      }
    }
  }
}
```

```

        key = "DC_YK";
    }
}
: boxed_radio_row {
    label = "Yikanma";
    key = "den2";
    alignment = top;
    : radio_button {
        label = " Kuvet";
        mnemonic="v";
        key = "DC_KU";
        fixed_width=true;
        width=12;
    }
    : radio_button {
        label = " Dus T.";
        mnemonic="D";
        key = "DC_DT";
    }
}
: boxed_radio_row {
    label = "Su Isitma";
    key = "den1";
    alignment = top;
    : radio_button {
        label = " Termosifon";
        mnemonic="m";
        key = "DC_TS";
        fixed_width=true;
        width=12;
    }
    : radio_button {
        label = " Sofben";
        mnemonic="f";
        key = "DC_SF";
    }
}
: boxed_radio_row {
    label = "Duvar";
    fixed_height = true;
    alignment = top;
    : radio_button {
        label = " 1 Duvar";
        key = "DC_1D";
        mnemonic="1" ;
        fixed_width=true;
        width=12;
    }
    : radio_button {
        label = " 2 Duvar";
        mnemonic="2" ;
        key = "DC_2D";
    }
}

```

```

    }
    : row {
        : text {label = "Tolerans" = "};
        : text {label = ""; key = "DC_TO";}
        : slider {
            mnemonic="r";
            key = "DC_SLD";
            min_value = 0;
            max_value = 7;
            width = 15;
            height = 6;
            small_increment = 1;
            big_increment = 1;
            fixed_width = true;
            fixed_height = true;
        }
    }
    spacer_1;
    : row {
        fixed_width = true;
        alignment = centered;
        : retirement_button {
            label="Ciz";
            key="DC_CZ";
            mnemonic="C" ;
            is_default=true;
        }
        : spacer {width = 2;}
        : retirement_button {
            label="iptal";
            key="DC_IP";
            mnemonic="i" ;
            is_cancel=true;
        }
        : spacer {width = 2;}
        : retirement_button {
            label = "Test...";
            key = "DC_DS";
            mnemonic = "T";
        }
    }
}
: column {
    : text {
        label = "";
        width = 0;
    }
}
: column {
    : boxed_column {
        label = "Vitrifiye";
        fixed_height = true;
        : row {
            : column {

```

```

        : text {
            label = "Marka";
            key = "DC_MKVx";
        }
        : text {
            label = "Antik";
            key = "DC_VTx";
        }
        : icon_image {
            key = "ic11";
            mnemonic = "V";
        }
    }
    : column {
        : text {
            label = "Marka";
            key = "DC_MKKx";
        }
        : text {
            label = "20X20";
            key = "DC_KPx";
        }
        : icon_image {key = "ic21";}
    }
}
: row {
    alignment = left;
    : text {label = "Duvar : ";}
    : text {
        label = "Marka";
        key = "DC_MKDx";
    }
    : text {label = " / ";}
    : text {
        label = "Olcu";
        key = "DC_KPy";
    }
}
}
: row {
    fixed_width = true;
    : text {label = "  Kombinasyon      ";}
    : text {
        label = "LA KL CM DT SF";
        key = "DC_KOM";
    }
}
}
: row {
    fixed_width = true;
    : button {
        label = "Permutasyon";
        key="DC_PRM";
        mnemonic="P";
        fixed_width = true;
    }
}

```


Firmalar menüsü ekranı (ikinci ekran):

```

DDbanyo2 : dialog
{label = "Firmalar";
  : column {
    : row {
      label = "Ureticiler";
      fixed_height = true;
      : column {
        : text {
          label = "*";
          key = "icx11x";
        }
        : icon_image {key = "icx11";}
      }
      : column {
        : text {
          label = "*";
          key = "icx12x";
        }
        : icon_image {key = "icx12";}
      }
      : column {
        : text {
          label = "*";
          key = "icx13x";
        }
        : icon_image {key = "icx13";}
      }
      : column {
        : text {
          label = "*";
          key = "icx14x";
        }
        : icon_image {key = "icx14";}
      }
      : column {
        : text {
          label = "*";
          key = "icx15x";
        }
        : icon_image {key = "icx15";}
      }
    }
  }
  : column { : text {label = ""}; : text {label = ""};}
  : row {
    label = "Secim";
    : column {
      : text {
        label = "Marka";
        key = "icx21x";
      }
      : text {
        label = "10X20";
      }
    }
  }
}

```

```

        key = "icx21y";
    }
    : icon_image {key = "icx21";}
}
: column {
    : text {
        label = "Marka";
        key = "icx22x";
    }
    : text {
        label = "20X20";
        key = "icx22y";
    }
    : icon_image {key = "icx22";}
}
: column {
    : text {
        label = "Marka";
        key = "icx23x";
    }
    : text {
        label = "30X30";
        key = "icx23y";
    }
    : icon_image {key = "icx23";}
}
}
: column {
    : text {label = "";}
}
: row {
    fixed_width = true;
    alignment = centered;
    : retirement_button {
        label="Tamam";
        key="DC2_TM";
        mnemonic="T";
        is_default=true;
    }
    : spacer {width = 2;}
    : retirement_button {
        label="iptal";
        key="DC2_IP";
        mnemonic="i";
        is_cancel=true;
    }
}
}
: column {
    : text {label = "";}
}
}
}

```

Firma bazında seçim ekranı (üçüncü ekran)

```

Ddbanyo3 : dialog
{label = "*";
  key = "DC_TP3";
  : column {
    : boxed_row {
      label = "Vitrifiye";
      : column {
        : text {
          label = "*";
          key = "icx311x";
        }
        : icon_image {
          key = "icx311";
        }
      }
    }
    : column {
      : text {
        label = "";
        key = "icx312x";
      }
      : icon_image {
        key = "icx312";
      }
    }
    : column {
      : text {
        label = "*";
        key = "icx313x";
      }
      : icon_image {
        key = "icx313";
      }
    }
  }
  : column {
    : text {
      label = "*";
      key = "icx314x";
    }
    : icon_image {
      key = "icx314";
    }
  }
  : column {
    : text {
      label = "*";
      key = "icx315x";
    }
    : icon_image {
      key = "icx315";
    }
  }
}

```

```

}
: boxed_row {
    label = "Yer Karosu";
    : column {
        : text {
            label = "*";
            key = "icx321x";
        }
        : icon_image {
            key = "icx321";
        }
    }
    : column {
        : text {
            label = "*";
            key = "icx322x";
        }
        : icon_image {
            key = "icx322";
        }
    }
    : column {
        : text {
            label = "*";
            key = "icx323x";
        }
        : icon_image {
            key = "icx323";
        }
    }
    : column {
        : text {
            label = "*";
            key = "icx324x";
        }
        : icon_image {
            key = "icx324";
        }
    }
    : column {
        : text {
            label = "*";
            key = "icx325x";
        }
        : icon_image {
            key = "icx325";
        }
    }
}
: boxed_row {
    label = "Duvar Karosu";
    : column {
        : text {
            label = "*";

```

```

        key = "icx331x";
    }
    : icon_image {
        key = "icx331";
    }
}
: column {
    : text {
        label = "*";
        key = "icx332x";
    }
    : icon_image {
        key = "icx332";
    }
}
: column {
    : text {
        label = "*";
        key = "icx333x";
    }
    : icon_image {
        key = "icx333";
    }
}
: column {
    : text {
        label = "*";
        key = "icx334x";
    }
    : icon_image {
        key = "icx334";
    }
}
: column {
    : text {
        label = "*";
        key = "icx335x";
    }
    : icon_image {
        key = "icx335";
    }
}
}
: column {
    : text {
        label = "";
    }
}
}
: row {
    : row {
        fixed_width = true;
        alignment = centered;
        : retirement_button {
            label="Tamam";

```


Uzun sürecek olan hesaplamalar sırasında kullanıcıya programın çalışmakta olduğunu hatırlatan geçici ekran (dördüncü ekran):

```

DDbanyo4 : dialog
{ label = "Kurallar uygulanıyor...";
  : column {
    : row {
      : text {
        label = "";
        key = "DC4_PS";
        fixed_width = true;
        width = 3;
      }
      : text {
        label = "";
        key = "DC4_PR";
        fixed_width = true;
        width = 17;
      }
      : text {
        label = "";
        key = "DC4_PP";
        fixed_width = true;
        width = 2;
      }
    }
    : text {
      label = "";
      key = "DC4_TL";
    }
    spacer_3;
    : retirement_button {
      label="Tamam...";
      key="DC4_IP";
      mnemonic="T";
      is_default=true;
    }
  }
}

```

Program çalıştırıldığında banyo.dwg dosyasında yapılmış çizimler var ise kullanıcı uyarmak için kullanılan ekran (beşinci ekran):

```
DDbanyo5 : dialog
{ label = "Dosya Bos Degil!";
  : column {
    : text {label = "Dosyada daha once yapilmis";}
    : text {label = "cizimler bulunuyor";}
    spacer_3;
    : retirement_button {
      label="Sil ve Devam Et.";
      key="DC4_TM";
      mnemonic="S";
      fixed_width = true;
      width = 22;
      is_default=true;
    }
    : retirement_button {
      label="Programdan Cik";
      key="DC4_IP";
      mnemonic="C";
      fixed_width = true;
      width = 22;
      is_cancel=true;
    }
  }
}
```

Seçilen tek bir alternatif çizilmek istenmişse ve bu alternatifin hata puanı toleranstan daha büyük ise, kullanıcıyı uyarmak için kullanılan ekran (altıncı ekran):

```

DDbanyo6 : dialog
{label = "Alternatif Cizilebilir Degil!";
  : column {
    fixed_highth = true;
    fixed_width = true;
    : row {
      fixed_width = true;
      : text {
        label = "    Tolerans :";
        key = "DC6_TO1";
        width = 19;
      }
      : text {
        key = "DC6_TO2";
        width = 2;
      }
    }
    : row {
      fixed_width = true;
      : text {
        label = "    Alternatif :";
        key = "DC6_AL1";
        width = 19;
      }
      : text {
        key = "DC6_AL2";
        width = 17;
      }
    }
    : row {
      fixed_width = true;
      : text {
        label = "    Alternatifin Puanı :";
        key = "DC6_AP1";
      }
      : text {
        key = "DC6_AP2";
        width = 2;
      }
    }
  }
  spacer_1;
  : text {
    label = "Secilen Alternatifin Hata Puanı
    Toleranstan Daha Buyuk";
  }
  : text {
    label = "Oldugu Icin Bu Alternatif Cizilemez.";
    alignment = centered;
  }
  spacer_1;
}

```

```
        : text {
            label = "Ana Ekrana Donmek icin Tamam Tusuna
            Basin.";
        }
    spacer_2;
    : retirement_button {
        label="Tamam...";
        key="DC6_IP";
        mnemonic="T";
        is_default=true;
    }
}
}
```



Kural tabanı erişim ekranı (yedinci ekran):

```

DDbanyo7 : dialog
{label = "Kural Tabanı";
  : row {
    : column {
      fixed_highth = true;
      fixed_width = true;
      : text {
        label = "Tek Duvarlı Kurallar:";
        key = "DC7_10";
      }
    }
    : row {
      : text {
        label = "1D";
        key = "DC7_11";
        width = 3;
      }
      : popup_list {
        key = "DC7_12";
        alignment = "left";
        fixed_width = true;
        width = 2;
        list = "2E\n3E\n4E\n5E";
      }
      : popup_list {
        key = "DC7_13";
        alignment = "left";
        fixed_width = true;
        width = 2;
        list = "A1\nA2\nA3\nA4\nA5";
      }
      : text {
        label = "=";
        width = 2;
        key = "DC7_14";
      }
      : popup_list {
        key = "DC7_15";
        alignment = "left";
        width = 2;
        list = "CM\nDT\nKL\nKU\nLA\nSF\nTS";
      }
      : text {
        label = "AD";
        width = 2;
        key = "DC7_16";
      }
      : popup_list {
        key = "DC7_17";
        alignment = "left";
        width = 2;
        list = "1\n2";
      }
    }
  }
}

```

```

    }
    : text {
        label = ">";
        width = 2;
        key = "DC7_18";
    }
    : popup_list {
        key = "DC7_19";
        alignment = "left";
        width = 2;
        list = "1\n2\n3\n4\n5\n6\n7\n8";
    }
}
spacer_1;
: text {
    label = "iki Duvarli Kurallar:";
    key = "DC7_20";
}
: row {fixed_width = true;
    : text {
        label = "2D";
        key = "DC7_21";
    }
    : text {
        label = "Ax =";
        key = "DC7_22";
    }
    : popup_list {
        key = "DC7_23";
        alignment = "left";
        width = 2;
        list = "CM\nDT\nKL\nKU\nLA\nSF\nTS";
    }
    : text {
        label = "Ay =";
        key = "DC7_24";
    }
    : popup_list {
        key = "DC7_25";
        alignment = "left";
        width = 2;
        list = "CM\nDT\nKL\nKU\nLA\nSF\nTS";
    }
    : text {
        label = "=>";
        key = "DC7_26";
    }
    : popup_list {
        key = "DC7_27";
        alignment = "left";
        width = 2;
        list = "1\n2\n3\n4\n5\n6\n7\n8";
    }
}
}

```

```

spacer_1;
: boxed_radio_row {label = "Aktif Kural";
    : radio_button {
        label = "1 Duvarli Kurallar";
        mnemonic="1";
        key = "DC7_1D";
    }
    : radio_button {
        label = "2 Duvarli Kurallar";
        mnemonic="2";
        key = "DC7_2D";
    }
}
}
: column {
    alignment = right;
    fixed_width = true;
    spacer_1;
    : retirement_button {
        label = "Ekle ";
        key = "DC7_EK";
        mnemonic = "E" ;
        fixed_width = 7;
    }
    : retirement_button {
        label = "Sil";
        key = "DC7_SL";
        mnemonic = "S" ;
        fixed_width = 7;
    }
    : retirement_button {
        label = "Mode";
        key = "DC7_MD";
        mnemonic = "M" ;
        fixed_width = 7;
    }
    : retirement_button {
        label="iptal";
        key="DC7_IP";
        mnemonic="i";
        is_cancel=true;
        fixed_width = 7;
    }
    : retirement_button {
        label="Tamam";
        key="DC7_TM";
        mnemonic="T";
        is_cancel=true;
        fixed_width = 7;
    }
}
}
: text {
    label = "Kurallar:";

```

```
        key = "DC7_KA";  
    }  
    : list_box {  
        mnemonic = "y";  
        key = "DC7_LS";  
        alignment = "left";  
        fixed_width = true;  
        width = 70;  
        height = 12;  
        fixed_width_font = true;  
    }  
}
```



Bir kural silinmeden önce kullanıcıyı uyarın ekran (sekizinci ekran):

```

DDbanyo8 : dialog
{label = "Kural Silinecek!";
  : column {width = 35;
    spacer_1;
    : text {
      fixed_width_font = true;
      key = "DC8_T1";
    }
    spacer_1;
    : text {
      label = "Bu kurali silmek istiyormusunuz?";
    }
    spacer_1;
    : row {
      : retirement_button {
        label="Evet";
        key="DC8_EV";
        mnemonic="E";
        fixed_width = 6;
      }
      : retirement_button {
        label="Hayir";
        key="DC8_HY";
        mnemonic="H";
        is_default=true;
        is_default=true;
        fixed_width = 6;
      }
    }
  }
}

```

EK D Banyo.krl Dosyası

Bu bölümde kural tabanını içeren Banyo.krl isimli dosyanın içeriği bulunur.

```

;;; 20.01.1997 Çelik ingin
;;;
;;; Banyo.krl
;;;
;;;
;;; bölüm 1
;;;
;;; 001 : kural numarası
;;; 1D : olasılığın ilgili olduğu duvar sayısı
;;; 2E : olasılıktaki eleman sayısı
;;;
;;; DV : kullanılan duvar sayısı
;;; AD : o an aktif olan duvar
;;;
;;; A1 : Aktif duvarın 1. elemanı
;;; A2 : Aktif duvarın 2. elemanı
;;; A3 : Aktif duvarın 3. elemanı
;;; A4 : Aktif duvarın 4. elemanı
;;; A5 : Aktif duvarın 5. elemanı
;;;
;;; D1 : birinci duvardaki eleman sayısı
;;; D2 : ikinci duvardaki eleman sayısı
;;;
;;; Ax : birinci duvardaki son eleman
;;; Ay : ikinci duvardaki ilk eleman
;;;
;;; LA : Lavabo için kullanılan kısaltma.
;;; KL : Klozet için kullanılan kısaltma.
;;; CM : Çamaşır makinesi için kullanılan kısaltma.
;;; DT : Dus teknesi için kullanılan kısaltma.
;;; KU : Küvet için kullanılan kısaltma.
;;; SF : Sofben için kullanılan kısaltma.
;;; TS : Termosifon için kullanılan kısaltma.
;;;
;;; =>x : hata puani
;;; =>8 : bulunduğu satiri etkisizleştirir

```

```

+ 001 1D 2E A1=DT AD=1 => 1
+ 002 1D 2E A1=KU AD=1 => 8
+ 003 1D 2E A2=SF AD=1 => 1
+ 004 1D 2E A2=TS AD=1 => 1

```

```

+ 005 1D 2E A2=SF AD=2 => 1
+ 006 1D 2E A2=TS AD=2 => 1

```

+ 007 1D 3E A1=DT AD=1 => 1
 + 008 1D 3E A1=KU AD=1 => 8
 + 009 1D 3E A2=CM AD=1 => 3
 + 010 1D 3E A2=DT AD=1 => 8
 + 011 1D 3E A2=KU AD=1 => 8
 + 012 1D 3E A2=SF AD=1 => 2
 + 013 1D 3E A2=TS AD=1 => 1
 + 014 1D 3E A3=TS AD=1 => 1

+ 015 1D 3E A2=CM AD=2 => 3
 + 016 1D 3E A2=DT AD=2 => 8
 + 017 1D 3E A2=KU AD=2 => 8
 + 018 1D 3E A2=TS AD=2 => 1
 + 019 1D 3E A3=TS AD=2 => 1

+ 020 1D 4E A1=DT AD=1 => 1
 + 021 1D 4E A1=KU AD=1 => 8
 + 022 1D 4E A2=CM AD=1 => 3
 + 023 1D 4E A2=DT AD=1 => 8
 + 024 1D 4E A2=KU AD=1 => 8
 + 025 1D 4E A2=SF AD=1 => 2
 + 026 1D 4E A2=TS AD=1 => 1
 + 027 1D 4E A3=CM AD=1 => 3
 + 028 1D 4E A3=DT AD=1 => 8
 + 029 1D 4E A3=KU AD=1 => 8
 + 030 1D 4E A3=SF AD=1 => 2
 + 031 1D 4E A3=TS AD=1 => 1

+ 032 1D 4E A2=CM AD=2 => 3
 + 033 1D 4E A2=DT AD=2 => 8
 + 034 1D 4E A2=KU AD=2 => 8
 + 035 1D 4E A2=SF AD=2 => 2
 + 036 1D 4E A2=TS AD=2 => 1
 + 037 1D 4E A3=CM AD=2 => 3
 + 038 1D 4E A3=DT AD=2 => 8
 + 039 1D 4E A3=KU AD=2 => 8
 + 040 1D 4E A3=SF AD=2 => 2
 + 041 1D 4E A3=TS AD=2 => 1

+ 042 1D 5E A1=DT AD=1 => 1
 + 043 1D 5E A1=KU AD=1 => 8
 + 044 1D 5E A2=CM AD=1 => 3
 + 045 1D 5E A2=DT AD=1 => 8
 + 046 1D 5E A2=KU AD=1 => 8
 + 047 1D 5E A2=SF AD=1 => 2
 + 048 1D 5E A2=TS AD=1 => 1

+ 049 1D 5E A3=CM AD=1 => 3
 + 050 1D 5E A3=DT AD=1 => 8
 + 051 1D 5E A3=KU AD=1 => 8
 + 052 1D 5E A3=SF AD=1 => 2
 + 053 1D 5E A3=TS AD=1 => 1
 + 054 1D 5E A4=CM AD=1 => 3
 + 055 1D 5E A4=DT AD=1 => 8
 + 056 1D 5E A4=KU AD=1 => 8
 + 057 1D 5E A4=SF AD=1 => 2
 + 058 1D 5E A4=TS AD=1 => 1

+ 059 1D 5E A2=CM AD=2 => 3
 + 060 1D 5E A2=DT AD=2 => 8
 + 061 1D 5E A2=KU AD=2 => 8
 + 062 1D 5E A2=SF AD=2 => 2
 + 063 1D 5E A2=TS AD=2 => 1
 + 064 1D 5E A3=CM AD=2 => 3
 + 065 1D 5E A3=DT AD=2 => 8
 + 066 1D 5E A3=KU AD=2 => 8
 + 067 1D 5E A3=SF AD=2 => 2
 + 068 1D 5E A3=TS AD=2 => 1
 + 069 1D 5E A4=CM AD=2 => 3
 + 070 1D 5E A4=DT AD=2 => 8
 + 071 1D 5E A4=KU AD=2 => 8
 + 072 1D 5E A4=SF AD=2 => 2
 + 073 1D 5E A4=TS AD=2 => 1

///

/// iki duvarli olasiliklarda köse noktasinin
 /// analizi için kurallar

///

/// Ax: birinci duvardaki son eleman

/// Ay: ikinci duvardaki ilk eleman

///

+ 074 2D Ax=CM ve Ay=DT => 3
 + 075 2D Ax=CM ve Ay=KL => 3
 + 076 2D Ax=CM ve Ay=LA => 3
 + 077 2D Ax=CM ve Ay=SF => 4
 + 078 2D Ax=CM ve Ay=TS => 3

+ 079 2D Ax=DT ve Ay=CM => 4
 + 080 2D Ax=DT ve Ay=KL => 3
 + 081 2D Ax=DT ve Ay=LA => 3
 + 082 2D Ax=DT ve Ay=SF => 4
 + 083 2D Ax=DT ve Ay=TS => 3

```
+ 084 2D Ax=KL ve Ay=CM => 3
+ 085 2D Ax=KL ve Ay=DT => 3
+ 086 2D Ax=KL ve Ay=LA => 2
+ 087 2D Ax=KL ve Ay=SF => 3
+ 088 2D Ax=KL ve Ay=TS => 2
```

```
+ 089 2D Ax=LA ve Ay=CM => 3
+ 090 2D Ax=LA ve Ay=DT => 3
+ 091 2D Ax=LA ve Ay=KL => 2
+ 092 2D Ax=LA ve Ay=SF => 3
+ 093 2D Ax=LA ve Ay=TS => 2
```

```
+ 094 2D Ax=SF ve Ay=CM => 3
+ 095 2D Ax=SF ve Ay=DT => 3
+ 096 2D Ax=SF ve Ay=DT => 3
+ 097 2D Ax=SF ve Ay=KL => 2
+ 098 2D Ax=SF ve Ay=LA => 2
```

```
+ 099 2D Ax=TS ve Ay=CM => 3
+ 100 2D Ax=TS ve Ay=DT => 3
+ 101 2D Ax=TS ve Ay=KL => 2
+ 102 2D Ax=TS ve Ay=LA => 2
+ 103 2D Ax=TS ve Ay=SF => 3
```

```
;;;
;;; bölüm 3
;;;
```

```
(if (and (= DV 2) (> D2 0) (= Ax "KU")) (setq A (+ A 8)))
(if (and (= DV 2) (> D1 0) (= Ay "KU")) (setq A (+ A 8)))
```

EK E Banyo.dta Dosyası

```

;;;
;;; 20.01.1997 Çelik ingin
;;;
;;; Banyo.dta
;;;
;;; Banyo programı için eleman bilgileri veri dosyası.
;;;
;;;
;;; bu dosyada, basında * karakteri olan her satır
;;; firma ismi satırı olarak değerlendirilecektir.
;;;

```

firmalar:

```

*Toprak
*Vitra
*Serel

```

```

;;;
;;;
;;; bölüm 2 çeşitler:
;;;
;;; modeller için alan adları
;;;
;;; (list model kısa_isim blok_adi X Xn Y Yn fiyat)
;;;
;;; X : Yatay genişlik (mm)
;;; Xn : Yatay kullanım alanı (mm)
;;; Y : Düşey genişlik (mm)
;;; Yn : Düşey kullanım alanı (mm)
;;; TL : Türk lirası olarak fiyatı (noktasız olacak)
;;;
;;; vitrifiye fiyatı takım olarak veriliyor...

```

Toprak_Vitrifiye_Antik

```

(list "Antik" "LA" "AN_LA" 660 200 560 200 15190000)
(list "Antik" "KL" "AN_KL" 420 200 730 200 15190000)

```

Toprak_Vitrifiye_Fiesta

```

(list "Fiesta" "LA" "FI_LA" 615 200 490 200 14790000)
(list "Fiesta" "KL" "FI_KL" 360 200 650 200 14790000)

```

Toprak_Vitrifiye_Koral

```

(list "Koral" "LA" "KO_LA" 630 200 510 200 15190000)
(list "Koral" "KL" "KO_KL" 380 200 660 200 15190000)

```

Toprak_Vitrifiye_Bella

```

(list "Bella" "LA" "BE_LA" 700 200 540 200 22000000)
(list "Bella" "KL" "BE_KL" 440 200 690 200 22000000)

```

Toprak_Vitrifiye_Nova

```
(list "Nova" "LA" "NO_LA" 650 200 510 200 27530000)
(list "Nova" "KL" "NO_KL" 390 200 740 200 27530000)
```

```
;;; karolar için alan adlari
```

```
;;;
```

```
;;; (list model kısa_isim derz X Xn Y Yn fiat)
```

```
;;;
```

```
;;; X : Yatay geniilik (mm)
```

```
;;; Xn : derz
```

```
;;; Y : Dusey genislik (mm)
```

```
;;; Yn : derz
```

```
;;; TL : Türk lirası olarak fiati (noktasız olacak)
```

Toprak_Yer_Karosus_20x20

```
(list "20x20" "YK" "Toprak" 200 2 200 2 600000)
```

Toprak_Yer_Karosus_30x30

```
(list "30x30" "YK" "Toprak" 300 2 300 2 1200000)
```

Toprak_Yer_Karosus_30x40

```
(list "30x40" "YK" "Toprak" 300 2 400 2 1300000)
```

Toprak_Yer_Karosus_33x33

```
(list "33x33" "YK" "Toprak" 330 2 330 2 1300000)
```

Toprak_Duvar_Karosus_20x20

```
(list "20x20" "DK" "Toprak" 200 2 200 2 950000)
```

Toprak_Duvar_Karosus_20x25

```
(list "20x25" "DK" "Toprak" 200 2 250 2 975000)
```

Toprak_Duvar_Karosus_25x33

```
(list "25x33" "DK" "Toprak" 250 2 330 2 1300000)
```

Vitra_Vitrifiye_Akdeniz

```
(list "Akdeniz" "LA" "AN_LA" 660 200 560 200 17200000)
```

```
(list "Akdeniz" "KL" "AN_KL" 420 200 730 200 17200000)
```

Vitra_Vitrifiye_Terme

```
(list "Terme" "LA" "FI_LA" 615 200 490 200 15500000)
```

```
(list "Terme" "KL" "FI_KL" 360 200 650 200 15500000)
```

Vitra_Vitrifiye_Pamukkal

```
(list "Pamukkal" "LA" "KO_LA" 630 200 510 200 19750000)
```

```
(list "Pamukkal" "KL" "KO_KL" 380 200 660 200 19750000)
```

Vitra_Vitrifiye_Bergama

```
(list "Bergama" "LA" "BE_LA" 700 200 540 200 25000000)
```

```
(list "Bergama" "KL" "BE_KL" 440 200 690 200 25000000)
```

Vitra_Vitrifiye_Side

```
(list "Side" "LA" "NO_LA" 650 200 510 200 30000000)
(list "Side" "KL" "NO_KL" 390 200 740 200 30000000)
```

Vitra_Yer Karosu_10x10

```
(list "10x10" "YK" "Vitra" 100 2 100 2 725000)
```

Vitra_Yer Karosu_20x20

```
(list "20x20" "YK" "Vitra" 200 2 200 2 1000000)
```

Vitra_Yer Karosu_33x33

```
(list "33x33" "YK" "Vitra" 330 2 330 2 1350000)
```

Vitra_Duvar Karosu_10x10

```
(list "10x10" "DK" "Vitra" 100 2 100 2 750000)
```

Vitra_Duvar Karosu_15x20

```
(list "15x20" "DK" "Vitra" 150 2 200 2 950000)
```

Vitra_Duvar Karosu_20x25

```
(list "20x25" "DK" "Vitra" 200 2 250 2 1125000)
```

Vitra_Duvar Karosu_25x33

```
(list "25" "DK" "Vitra" 250 2 330 2 1200000)
```

Serel_Vitrifiye_Pinar

```
(list "Pinar" "LA" "AN_LA" 660 200 560 200 13300000)
(list "Pinar" "KL" "AN_KL" 420 200 730 200 13300000)
```

Serel_Vitrifiye_Sebnem

```
(list "Sebnem" "LA" "FI_LA" 615 200 490 200 12000000)
(list "Sebnem" "KL" "FI_KL" 360 200 650 200 12000000)
```

Serel_Vitrifiye_Nilufer

```
(list "Nilufer" "LA" "KO_LA" 630 200 510 200 15000000)
(list "Nilufer" "KL" "KO_KL" 380 200 660 200 15000000)
```

Serel_Vitrifiye_Irmak

```
(list "Irmak" "LA" "BE_LA" 700 200 540 200 20000000)
(list "Irmak" "KL" "BE_KL" 440 200 690 200 20000000)
```

Serel_Vitrifiye_Nergis

```
(list "Nergis" "LA" "NO_LA" 650 200 510 200 25000000)
(list "Nergis" "KL" "NO_KL" 390 200 740 200 25000000)
```

Serel_Yer Karosu_20x20

```
(list "20x20" "YK" "Serel" 200 2 200 2 725000)
```

Serel_Yer Karosu_30x30

```
(list "30x30" "YK" "Serel" 300 2 300 2 980000)
```

Serel_Yer Karosu_33x33

```
(list "33x33" "YK" "Serel" 330 2 330 2 1330000)
```

Serel_Yer_Karosu_40x40

(list "40x40" "YK" "Serel" 400 2 400 2 1250000)

Serel_Duvar_Karosu_15x20

(list "15x20" "DK" "Serel" 150 2 200 2 850000)

Serel_Duvar_Karosu_20x25

(list "20x25" "DK" "Serel" 200 2 250 2 900000)

Serel_Duvar_Karosu_25x33

(list "25x33" "DK" "Serel" 250 2 330 2 1100000)

;;; diger elemanlar için alan adlari

;;;

;;; (list tanımsız kısa_isim blok_adi X Xn Y Yn fiat önal)

;;;

;;; X : Yatay genişlik (mm)

;;; Xn : yatay kullanım alanı (mm)

;;; Y : Düşey genişlik (mm)

;;; Yn : düşey kullanım alanı (mm)

;;; TL : Türk lirası olarak fiati (noktasız olacak)

;;; önal : ön kullanım alanı (mm)

Çamaşır Makinası

(list "*" "CM" "CM" 600 250 600 250 50000000 450)

Düş Teknesi

(list "*" "DT" "DT" 800 500 800 250 50000000 750)

Küvet

(list "*" "KU" "KU" 700 250 1700 0 50000000 0)

Sofben

(list "*" "SF" "SF" 350 200 255 200 50000000 350)

Termosifon

(list "*" "TS" "TS" 430 200 430 200 50000000 0)

;;; Ek Bigiler

;;;

;;; 7 Tavan m2 maliyeti

;;; 6 Tavan Yüksekliği

;;; 5 Pencere Yüksekliği

;;; 4 Kapi Yüksekliği

;;; 0 alan adı

;;;

;;; ölçüler mm dir

;;;

	0	1	2	3	4	5	6	7
--	---	---	---	---	---	---	---	---

(list "Ekler" "x" "x" 0 2200 500 2700 350000)

EK F Banyo.def Dosyası

Bu dosya program çalışmaya henüz başladığında varsayılan (default) değerleri programa gönderir. Kullanıcı programı çalıştırdığında, sürekli olarak kullandığı değerleri görmek istiyorsa bu dosyadan gerekli değişiklikleri yapabilir.

```

;;;
;;; 20.01.1997 Çelik ingin
;;;
;;; Banyo.def
;;;
;;; Banyo programı için varsayılan degerler dosyasi.
;;;
;;;
;;; "-" set_tile "label" adi
;;; "*" set_tile degeri
;;;

;;Elemanlar

-DC_LA
*1
-DC_KL
*1
-DC_CM
*0
-DC_SS
*1
-DC_YK
*1

;;Yikanma

-DC_KU
*1
-DC_DT
*0

;;Sıcak Su

-DC_TS
*0
-DC_SF
*1

```

;;Duvar Sayisi

-DC_1D
*1
-DC_2D
*0

;;Tolerans

-DC_TO
*4
-DC_SLD
*4

;;Vitrifiye

-DC_MKVx
*Toprak
-DC_VTx
*Antik

;;Yer Kaplamasi

-DC_MKKx
*Vitra
-DC_KPx
*20x20

;;Duvar Kaplamasi

-DC_MKDx
*Toprak
-DC_KPy
*20x20

;;Çizim sayisi

-DC_CZ1
*1
-DC_CZ2
*0
-DC_CZ3
*0
-DC_CZ3x
*1

ÖZGEÇMİŞ

Halen İ.T.Ü. Mimarlık Fakültesinde Mimarlık Anabilim dalı Bina Bilgisi Programında Yüksek Lisans Öğrenimini sürdüren Mimar Çelik İngin 21 Ağustos 1970 yılında Sivas'ta doğmuştur. İlk ve orta öğrenimini İstanbul Fikret Yüzerli Ortaokulu'nda, lise öğrenimini ise 1987 yılında İstanbul Ataköy Lisesi'nde tamamlamıştır.

1987 yılında öğrenime başladığı İ.T.Ü. Mimarlık Fakültesi Mimarlık Bölümü'nden 1992 yazında mezun olmuştur. 1993 yazında başladığı yüksek lisans öğrenimini halen sürdürmektedir. Yabancı dillerden Almanca ve İngilizce bilmektedir.

Ortaokul yıllarında başladığı, mimarlık öğrenimi dolayısıyla ara verdiği bilgisayar çalışmalarına yüksek lisans öğrenimiyle tekrar başlamıştır. Bu dönem içinde MS Basic, Pascal, MS Visual Basic, Clipper, AutoLISP programlama dillerinde yazılım geliştirmiş, Dbase III+, MS FoxPro, AutoCAD ve 3D Studio programlarıyla çalışmıştır.

Değişik firmalarda bir süre mimarlık çalışmaları yaptıktan sonra, çeşitli firmalar için MS Visual Basic'le uygulama yazılımları geliştirmiş, 3D Studio programıyla mimari sunum, canlandırma, ve çokluortam (multimedia) çalışmaları yapmıştır.

Halen bilgisayar çalışmalarının yanı sıra aydınlatma armatürleri üreten bir firmada tasarımcı olarak görev yapmaktadır.