

Kontrol Yapıları (2)

Giriş
Sayaç Kontrollü Tekrar Yapısı
for Tekrar Yapısı
for Yapısı örnekleri
switch Çoklu Seçim Yapısı
do/while Tekrar Yapısı
break ve continue deyimleri
Mantık ve Şart Operatörleri
Özet

Sayaç Kontrollü Tekrar Yapısı

- **Sayaç Kontrollü Tekrar**
 - Kontrol değişkeni (Control variable)
 - Değişken döngünün devam edip etmeyeceğini belirlemek için kullanılır
 - Kontrol değişkeninin başlangıç değeri atanır
 - Değişken değeri artırılır veya azaltılır
 - Döngünün devamı için şart
 - Şart doğruysa döngü devam eder

```
1 // Fig. 5.1: WhileCounter.cs
2 // Counter-controlled repetition.
3
4 using System;
5
6 class WhileCounter
7 {
8     static void Main( string[] a
9     {
10        int counter = 1; // in
11        while ( counter <= 5 ) // repe
12        {
13            Console.WriteLine( counter );
14            counter++; // in
15        } // end while
16    } // end method Main
17 } // end class WhileCounter
```

Sayaç deęişkeni 1 deęeriyle başlatılır.

Döngü sayaç 5 deęerinden büyük oluncaya kadar devam eder (6 deęeri olduğunda biter)

Sayaç her döngüde 1 artırılır.

```
1
2
3
4
5
```

for Tekrar Yapısı

- **for** tekrar yapısı
 - Syntax: **for** (Expression1, Expression2, Expression3)
 - Expression1 = Kontrol deęişkeninin adı
 - Birden fazla deęişken içerebilir
 - Expression2 = Döngü devam şartı
 - Expression3 = Artırma/Azaltma
 - Expression1 birden fazla deęişkene sahipse Expression3'te buna baęlı olarak birden fazla deęişken içerir
 - Variable scope
 - Expression1 içinde tanımlanan deęişkenler sadece **for** döngüsü içinde kullanılabilir.
 - Döngü bitince deęişkenler geçersiz olur.

for Tekrar Yapısı

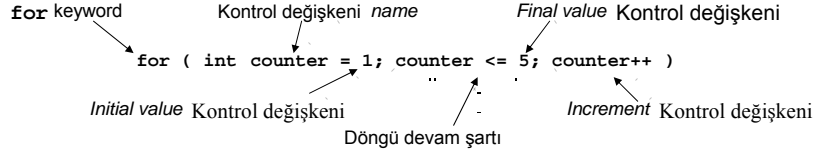


Fig. 5.3 for başlığının elemanları

for Tekrar Yapısı

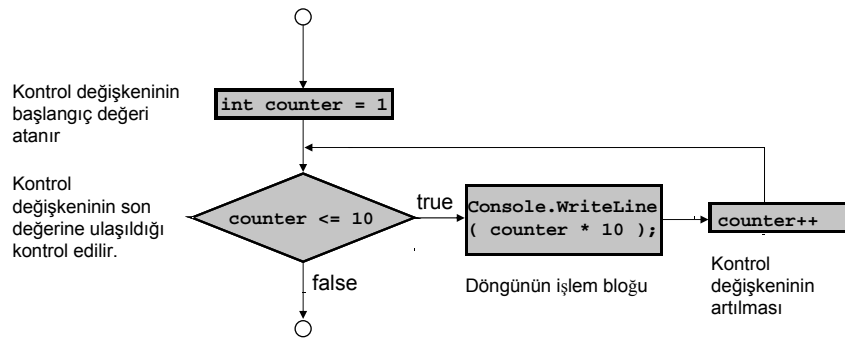


Fig. 5.4 for tekrar yapısının akış şeması.

```
1 {
2   Kontrol deęiřkeni 1 deęeriyle
3   bařlatıldı
4   using System;
5
6   class ForCounter
7   {
8     static void Main(
9     {
10      // initialization, repetition condition and incrementing
11      // are all included in the for structure
12      for ( int counter = 1; counter <= 5; counter++ )
13        Console.WriteLine( counter );
14    }
15  }
```

with the for structure.

Sayaç her döngüde 1 artırılır

Kontrol deęiřkeni 6'dan küçük olduęu sürece döngü devam eder

```
1
2
3
4
5
```

for Yapısı Örnekleri

- Artırma ve Azaltma
 - Artırmada
 - Çoęunlukla < veya <= kullanılır
 - Azaltmada
 - Çoęunlukla > veya >= kullanılır
- Mesaj kutuları (Message boxes)
 - Butonlar
 - OK
 - OKCancel
 - YesNo
 - AbortRetryIgnore
 - YesNoCancel
 - RetryCancel

for Yapısı Örnekleri

- Mesaj kutuları
 - Icon'lar
 - **Exclamation**
 - **Question**
 - **Error**
 - **Information**
- Format
 - (variable : format)
 - Format kodları Fig. 5.9'da verilmiştir.

```
1 // Fig. 5.5: Sum.cs
2 // Summation
3
4 using System;
5 using System.Windows.Forms;
6
7 class Sum
8 {
9     static void Main( string[] args )
10    {
11        int sum = 0;
12
13        for ( int number = 2; number <= 100; number += 2 )
14            sum += number;
15
16        MessageBox.Show( "The sum is " + sum,
17                        "Sum Even Integers from 2 to 100",
18                        MessageBoxButtons.OK,
19                        MessageBoxIcon.Information );
20
21    } // end method Main
22
23 } // end class Sum
```

Sayaç 2 değeriyle başlatıldı

Sayaç 100'den büyükse döngü biter

Her döngüde sayaç 2 artar

Mesaj metni

Mesaj kutusunun başlığı

Ekranda OK butonunu gösterir

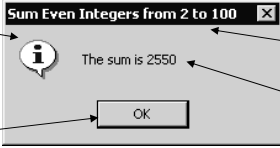
information icon ekranda gösterilir

Parametre 4: **MessageBox** Icon (Opsiyonel)

Parametre 2: Başlık çubuğu metni (Opsiyonel)

Parametre 3: **OK** butonu (Opsiyonel)

Parametre 1: Ekranda gösterilecek mesaj



for Yapısı Örnekleri





MessageBox Icon'ları	Icon	Tanım
<code>MessageBoxIcon.Exclamation</code>		Ekrana ünlem işaretini çıkarır ve kullanıcıyı olası bir probleme karşı uyarır.
<code>MessageBoxIcon.Information</code>		Information işaretini ekrana çıkarır ve kullanıcıyı bir konuda bilgilendirmek için kullanılır.
<code>MessageBoxIcon.Question</code>		Ekrana soru işaretini çıkarır ve kullanıcıyı bir seçim için soru yöneltmek için kullanılır.
<code>MessageBoxIcon.Error</code>		Ekrana X işaretini çıkarır ve önemli bir problem için kullanıcı uyarılır.

Fig. 5.6 Mesaj kutuları için icon'lar

for Yapısı Örnekleri

MessageBox Butonları	Tanımlama
<code>MessageBoxButton.OK</code>	Diyalog kutusunun OK butonu içerdiğini bildirir.
<code>MessageBoxButton.OKCancel</code>	Diyalog kutusunun OK ve Cancel butonlarını birlikte içerdiğini bildirir. Kullanıcı işlemi onaylayabilir veya iptal edebilir.
<code>MessageBoxButton.YesNo</code>	Diyalog kutusunun Yes ve No butonlarını birlikte içerdiğini bildirir. Kullanıcıdan bir soya cevap almak için kullanılır.
<code>MessageBoxButton.YesNoCancel</code>	Diyalog kutusunun Yes, No ve Cancel butonlarını birlikte içerdiğini bildirir. Kullanıcı işlemi kabul edebilir, kabul etmeyebilir veya iptal edebilir.
<code>MessageBoxButton.RetryCancel</code>	Diyalog kutusunun Retry ve Cancel butonlarını birlikte içerdiğini bildirir. Kullanıcı bir hatadan dolayı uyarılır. Kullanıcı işlemi tekrar ettirebilir veya iptal edebilir.
<code>MessageBoxButton.AbortRetryIgnore</code>	Diyalog kutusunun Abort, Retry ve Ignore butonlarını birlikte içerdiğini bildirir. Kullanıcı bir hatadan dolayı uyarılır. Kullanıcı işlemi kesebilir, tekrar edebilir veya hatayı gözardı edebilir.

Fig. 5.7 Mesaj kutuları için butonlar

```

1 // Fig. 5.8: Interest.cs
2 // Calculating compound interest.
3
4 using System;
5 using System.Windows.Forms;
6
7 class Interest
8 {
9     static void Main( string[] args )
10    {
11        decimal amount, principal = ( decimal ) 1000.00;
12        double rate = .05;
13        string output;
14
15        output = "Year\tAmount on deposit\n";
16
17        for ( int year = 1; year <= 10; year++ )
18        {
19            amount = principal *
20                ( decimal ) Math.Pow( 1.0 + rate, year );
21
22            output += year + "\t" +
23                String.Format( "{0:C}", amount ) + "\n";
24        }
25
26        MessageBox.Show( output, "Compound Interest",
27                        MessageBoxButtons.OK, MessageBoxIcon.Information );
28
29    } // end method Main
30
31 } // end class Interest

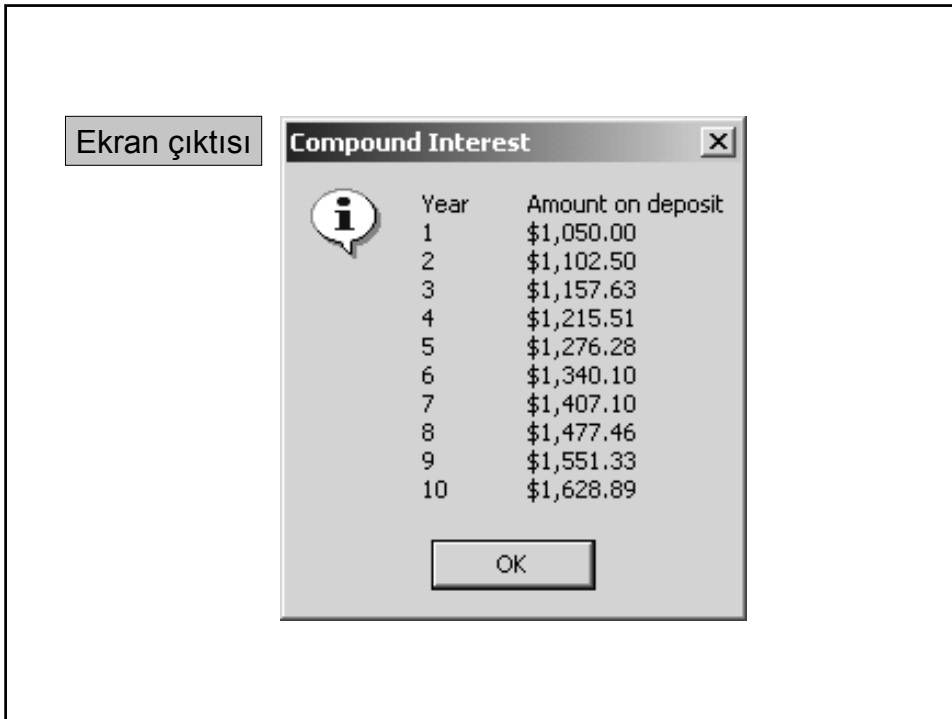
```

Döngü 10 kez tekrarlanır. Her seferde 1 artırılır ve 11 olunca çıkılır

Para birimine sahip format oluşturur

Bir Tab ekle

"Compound Interest" başlığına sahip ve bir OK butonu ile bir Information icon'ına sahip bir mesaj kutusu ekrana getirir.



for Yapısı Örnekleri

Format Kod	Tanımlama
C or c	Parabirimi formatı. İlgili para birimi ülkeye göre değişir. Bilgisayarın bölgesel ayarlarına bağlı olarak çalışır.
D or d	String türündeki bilgileri integer olarak ekrana yazdırır.
N or n	Virgülden sonraki iki basamak olarak yazar..
E or e	Bilimsel gösterimdir. Virgülden sonra 6 basamak alır.
F or f	Sabit basamak sayısına göre sayıyı yazar. İstenen basamak sayısı F'den hemen sonra girilir.
G or g	Genel formattır.
X or x	Hexadecimal formattır.

Fig. 5.9 Format kodları.

switch Çoklu Seçim Yapısı

- **switch** deyimi
 - Sabit deyimler
 - Durumlar (Cases)
 - Case 'x' :
 - Sabit durum değişkenleri kullanılır
 - Boş durumlar (Empty Cases)
 - Varsayılan durum (default case)
 - **break** deyimi
 - **switch** deyiminden çıkış


```

1 // Fig. 5.10: SwitchTest.cs
2 // Counting letter grades.
3
4 using System;
5
6 class SwitchTest
7 {
8     static void Main( string[] args )
9     {
10         char grade; // one grade
11         int aCount = 0, // number of As
12             bCount = 0, // number of Bs
13             cCount = 0, // number of Cs
14             dCount = 0, // number of Ds
15             fCount = 0; // number of Fs
16
17         for ( int i = 1; i <= 10; i++ )
18         {
19             Console.Write( "Enter a letter grade: " );
20             grade = Char.Parse( Console.ReadLine() );
21
22             switch ( grade )
23             {
24                 case 'A': // grade is uppercase A
25                 case 'a': // or lowercase a
26                     ++aCount;
27                     break;
28
29                 case 'B': // grade is uppercase B
30                 case 'b': // or lowercase b
31                     ++bCount;
32                     break;
33

```

```

34                 case 'C': // grade is uppercase C
35                 case 'c': // or lowercase c
36                     ++cCount;
37                     break;
38
39                 case 'D': // grade is uppercase D
40                 case 'd': // or lowercase d
41                     ++dCount;
42                     break;
43
44                 case 'F': // grade is uppercase F
45                 case 'f': // or lowercase f
46                     ++fCount;
47                     break;
48
49                 default: // processes all other characters
50                     Console.WriteLine(
51                         "Incorrect letter grade entered." +
52                         "\nGrade not added to totals." );
53                     break;
54             } // end switch
55         } // end for
56
57     } // end method Main
58
59     Console.WriteLine(
60         "\nTotals for each letter grade are:\nA: {0}" +
61         "\nB: {1}\nC: {2}\nD: {3}\nF: {4}", aCount, bCount,
62         cCount, dCount, fCount );
63
64 } // end class SwitchTest
65
66

```

```
Enter a letter grade: a
Enter a letter grade: A
Enter a letter grade: c
Enter a letter grade: F
Enter a letter grade: z
Incorrect letter grade entered.
Grade not added to totals.
Enter a letter grade: D
Enter a letter grade: d
Enter a letter grade: B
Enter a letter grade: a
Enter a letter grade: C

Totals for each letter grade are:
A: 3
B: 1
C: 2
D: 2
F: 1
```

switch Çoklu Seçim Yapısı

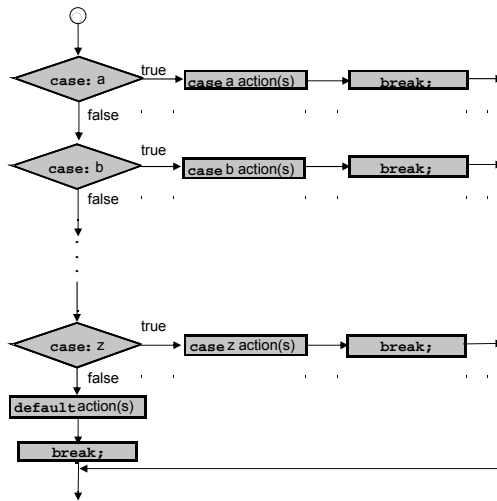


Fig. 5.11 switch çoklu seçimin akış şeması.

do/while Tekrar Yapısı

- **while** döngüsü ile **do/while** döngüsü karşılaştırması
 - **while** döngüsü
 - Döngü şartı test edilir
 - İşlemler gerçekleştirilir
 - Döngü tekrarlanır
 - **do/while**
 - İşlemler gerçekleştirilir
 - Döngü şartı test edilir
 - Döngünün tamamı en az bir kez çalıştırılır

```
1 // Fig. 5.12: DoWhileLoop.cs
2 // The do/while repetition structure.
3
4 using System;
5
6 class DoWhileLoop
7 {
8     static void Main( string[] args )
9     {
10         int counter = 1;
11
12         do
13         {
14             Console.WriteLine( counter );
15             counter++;
16         } while ( counter <= 5 );
17
18     } // end method Main
19
20 } // end class DoWhileLoop
```

```
1
2
3
4
5
```

do/while Tekrar Yapısı

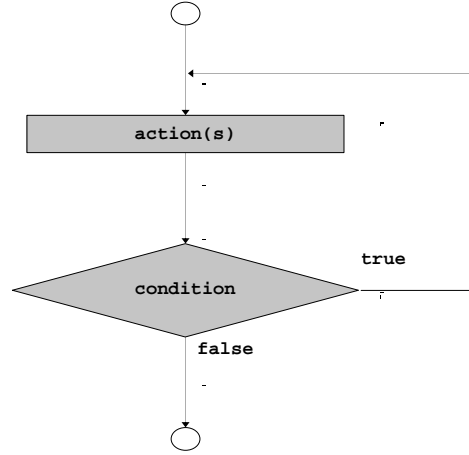
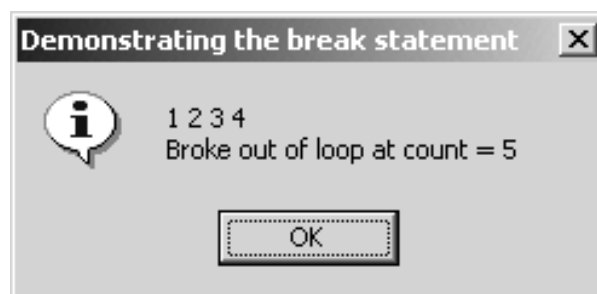


Fig. 5.13 do/while tekrar yapısı akış şeması.

break ve continue deyimleri

- Kullanım
 - Kontrol akışını değiştirmek için kullanılır
 - **break** deyimi
 - Döngüden erken çıkmak için kullanılır
 - **continue** deyimi
 - Döngünün kalan işlemleri atlanır ve döngünün başına dönülür

```
1 // Fig. 5.14: BreakTest.cs
2 // Using the break statement in a for structure.
3
4 using System;
5 using System.Windows.Forms;
6
7 class BreakTest
8 {
9     static void Main( string[] args )
10    {
11        string output = "";
12        int count;
13
14        for ( count = 1; count <= 10; count++ )
15        {
16            if ( count == 5 )
17                break;           // skip remaining code in loop
18                                // if count == 5
19
20            output += count + " ";
21
22        } // end for loop
23
24        output += "\nBroke out of loop at count = " + count;
25
26        MessageBox.Show( output, "Demonstrating the break statement",
27                        MessageBoxButtons.OK, MessageBoxIcon.Information );
28
29    } // end method Main
30
31 } // end class BreakTest
```



```

1 // Fig. 5.15: ContinueTest.cs
2 // Using the continue statement in a for structure.
3
4 using System;
5 using System.Windows.Forms;
6
7 class ContinueTest
8 {
9     static void Main( string[] args )
10    {
11        string output = "";
12
13        for ( int count = 1; count <= 10; count++ )
14        {
15            if ( count == 5 )
16                continue; // skip remaining code in loop
17                          // only if count == 5
18
19            output += count + " ";
20        }
21
22        output += "\nUsed continue to skip printing 5";
23
24        MessageBox.Show( output, "Using the continue statement",
25                        MessageBoxButtons.OK, MessageBoxIcon.Information );
26
27    } // end method Main
28
29 } // end class ContinueTest

```



Mantık ve Şart Operatörleri

- Operatörler
 - Mantıksal AND (&)
 - Şart AND (&&)
 - Mantıksal OR (|)
 - Şart OR (||)
 - Mantıksal exclusive OR or XOR (^)
 - Mantıksal NOT (!)
- Birden fazla şart deyimlerinde kullanılır

Mantık ve Şart Operatörleri

expression1	expression2	expression1 && expression2
false	false	false
false	true	false
true	false	false
true	true	true

Fig. 5.16 Mantıksal AND (&&) için doğruluk tablosu

expression1	expression2	expression1 expression2
false	false	false
false	true	true
true	false	true
true	true	true

Fig. 5.17 Mantıksal OR (||) için doğruluk tablosu

Mantık ve Şart Operatörleri

expression1	expression2	expression1 ^ expression2
false	false	false
false	true	true
true	false	true
true	true	false

Fig. 5.18 Mantıksal exclusive OR (^) için doğruluk tablosu

expression	!expression
false	true
True	false

Fig. 5.19 Mantıksal NOT (!) için doğruluk tablosu

```
1 // Fig. 5.20: LogicalOperators.cs
2 // Demonstrating the logical operators.
3 using System;
4
5 class LogicalOperators
6 {
7     // main entry point for application
8     static void Main( string[] args )
9     {
10         // testing the conditional AND operator (&&)
11         Console.WriteLine( "Conditional AND (&&)" +
12             "\nfalse && false: " + ( false && false ) +
13             "\nfalse && true: " + ( false && true ) +
14             "\ntrue && false: " + ( true && false ) +
15             "\ntrue && true: " + ( true && true ) );
16
17         // testing the conditional OR operator (||)
18         Console.WriteLine( "\n\nConditional OR (||)" +
19             "\nfalse || false: " + ( false || false ) +
20             "\nfalse || true: " + ( false || true ) +
21             "\ntrue || false: " + ( true || false ) +
22             "\ntrue || true: " + ( true || true ) );
23
24         // testing the logical AND operator (&)
25         Console.WriteLine( "\n\nLogical AND (&)" +
26             "\nfalse & false: " + ( false & false ) +
27             "\nfalse & true: " + ( false & true ) +
28             "\ntrue & false: " + ( true & false ) +
29             "\ntrue & true: " + ( true & true ) );
30     }
31 }
```



```

31 // testing the logical OR operator (|)
32 Console.WriteLine( "\n\nLogical OR (|)" +
33     "\nfalse | false: " + ( false | false ) +
34     "\nfalse | true: " + ( false | true ) +
35     "\ntrue | false: " + ( true | false ) +
36     "\ntrue | true: " + ( true | true ) );
37
38 // testing the logical exclusive OR operator (^)
39 Console.WriteLine( "\n\nLogical exclusive OR (^)" +
40     "\nfalse ^ false: " + ( false ^ false ) +
41     "\nfalse ^ true: " + ( false ^ true ) +
42     "\ntrue ^ false: " + ( true ^ false ) +
43     "\ntrue ^ true: " + ( true ^ true ) );
44
45 // testing the logical NOT operator (!)
46 Console.WriteLine( "\n\nLogical NOT (!)" +
47     "\n!false: " + ( !false ) +
48     "\n!true: " + ( !true ) );
49 }
50 }

```

```

Conditional AND (&)
false && false: False
false && true: False
true && false: False
true && true: True

```

```

Conditional OR (||)
false || false: False
false || true: True
true || false: True
true || true: True

```

```

Logical AND (&)
false & false: False
false & true: False
true & false: False
true & true: True

Logical OR (|)
false | false: False
false | true: True
true | false: True
true | true: True

Logical exclusive OR (^)
false ^ false: False
false ^ true: True
true ^ false: True
true ^ true: False

Logical NOT (!)
!false: True
!true: False

```

Haftalık Ödev

Bir otoparkta otomobil için 2YTL, kamyon için 5YTL ve otobüs için 8YTL ücret alınmaktadır. Kar oranları otomobil için %25, kamyon için %30 ve otobüs için %35 olarak belirlenmiştir. Kullanıcı gelen araçları sürekli bilgisayara girmektedir. Araç türü bilgisi olarak otomobil için "O", kamyon için "K" ve Otobüs için "B" girilmektedir. Programdan çıkmak için "C" harfi girilmektedir. Her araç girildiğinde yeni araç türü girilmesi istenmekte ve çıkış için "C" girilmesi gerektiği kullanıcıya bildirilmektedir. "C" ile çıkıldığında ekrana aşağıdaki bilgileri başlıkları ile birlikte (Örn.: TOPLAM KAR MİKTARI = 75 YTL) alt alta yazan bir program yazınız.

- Toplam otomobil, kamyon ve otobüs sayılarını
- Toplam otomobil, kamyon ve otobüs cirolarını
- Otomobil, kamyon ve otobüs cirolarını yüzdelik dağılımlarını
- Toplam otomobil, kamyon ve otobüs kar miktarlarını
- Otomobil, kamyon ve otobüs kar miktarlarını yüzdelik dağılımlarını
- Toplam araç sayısını
- Toplam gelir miktarını
- Toplam kar miktarını

Not:

Ödev programın kod çıktısı şeklinde teslim edilecek. Kapak sayfasında sadece öğrenci bilgileri (Numara, Ad Soyad) ile ödev adı olarak "OTOPARK OTOMASYON PROGRAMI" başlığı yazılacaktır. Haftalık ödev rapor kapak formatı <http://w3.gazi.edu.tr/web/akcayol> web sayfasından downloads bölümünden elde edilebilir.