

# PROGRAMLAMA TEMELLERİ

Fonksiyonlar

# Fonksiyonlar

- C++ bütün altyordamlara fonksiyon gözüyle bakar.
- Ana program (main) bile bir fonksiyondur.

# C++'ta Fonksiyonlar

- Ana programımız main bir fonksiyondur.

```
#include <cstdlib>
#include <iostream>
```

Dönüş tipi

namespace

Parametre

```
int main(int argc, char *argv[])
{
    system("PAUSE");
    return EXIT_SUCCESS; // 0
}
```

Fonksiyon adı

Dönüş değeri  
İşletim sistemi bu  
değeri  
ERRORLEVEL'da  
saklar.

# C++'ta Fonksiyonlar

```
• #include <cstdlib>
• #include <iostream>
• using namespace std;
• int TekMi (int sayi)
• {
•     return (sayi % 2) ;
• }
• int main(int argc, char *argv[])
• {
•     int sayi;
•     cout << "Tek Mi Cift Mi ..." << endl;
•     cout << "Bir tamsayi: ";
•     cin >> sayi;
•
•     if (TekMi(sayi)) cout << "Sayi Tek" << endl;
•     else cout << "Sayi Cift" << endl;
•     system("PAUSE");
•     return EXIT_SUCCESS;
• }
```

Fonksiyonumuzun adı "TekMi"

Parametrelerimiz: 1 adet, "int" tipinde ve adı «sayi» "sayi" lokal değişkeninin değeri "main" ana fonksiyonu tarafından atanacak.

Fonksiyonumuzun dönüş tipi "int"

Dönüş değeri  
0 ise Çift, 1 ise Tek

# C++'ta fonksiyonlar

Parametrelerde kullanılan yerel deęişkenlerin isimleri, ana programdaki ile aynı olmak zorunda deęildir.

- Bir programda ister bir deęişkenle, istersek doğrudan veri ile altprogramı çağırabiliriz.
- Bir programda birden fazla –farklı deęişken adıyla- deęişken veya veri ile altprogramı çağırabiliriz.
- Altprogramı farklı programlarda da çağırabiliriz.

# C++'ta fonksiyonlar

- **void**
- fonksiyonun **dönüş tipi** olduğunda «fonksiyon hiçbir değer döndürmeyecek» anlamına gelir. Bu tür fonksiyonlarda **return** komutu kullanılmaz.
- fonksiyonun parametresi olduğunda «fonksiyon dışarıdan değer kabul etmez» anlamına gelir.

# Ödev (19.12.2011)

- Alan3 adlı bir fonksiyon tanımlayınız. Kendisine parametre olarak gelen 3 kenar uzunluğuna bakarak üçgenin alanını döndürsün.
- AlanKare adlı bir fonksiyon tanımlayınız. Kendisine parametre olarak gelen 1 kenar uzunluğuna bakarak karenin alanını döndürsün.
- AlanDaire adlı bir fonksiyon tanımlayınız. Kendisine parametre olarak gelen yarıçap uzunluğuna bakarak dairenin alanını döndürsün.
- Alan4 adlı bir fonksiyon tanımlayınız. Kendisine parametre olarak gelen 4 kenar uzunluğuna bakarak dışbükey dörtgenin alanını döndürsün.

```
/* Alan3 adlı bir fonksiyon tanımlayınız.  
Kendisine parametre olarak gelen 3 kenar uzunluğuna  
bakarak üçgenin alanını döndürsün. */  
#include <cstdlib>  
#include <iostream>  
#include <cmath>  
using namespace std;  
// <dönüş tipi> <fonk adı> ( <parametre listesi> ) { <komutlar> }  
double Alan3 (double a, double b, double c)  
{  
    double alan, u=(a+b+c)/2;  
    alan= sqrt(u*(u-a)*(u-b)*(u-c));  
    return alan;  
}  
int main(int argc, char *argv[])  
{  
    double kenar1, kenar2, kenar3;  
    cout<< "3genin kenar uzunluklari..."<<endl;  
    cin >> kenar1 >> kenar2 >> kenar3;  
    cout << "ALANI: " << Alan3(kenar1, kenar2, kenar3)<< endl;  
    system("PAUSE");  
    return EXIT_SUCCESS;  
}
```



```
/* AlanKare adlı bir fonksiyon tanımlayınız.
```

```
Kendisine parametre olarak gelen 1 kenar uzunluğuna bakarak  
karenin alanını döndürsün. */
```

```
#include <cstdlib>
```

```
#include <iostream>
```

```
using namespace std;
```

```
double AlanKare (double a)
```

```
{  
    return a*a;  
}
```

```
int main(int argc, char *argv[])
```

```
{  
    double kenar;  
    cout<<"Karenin 1 KENARI: ";  
    cin >> kenar;  
    cout <<"ALANI: " << AlanKare(kenar) << endl;  
    system("PAUSE");  
    return EXIT_SUCCESS;  
}
```

# Referans ve Değer

- Referans: değişkenlerin buldukları bellek adresidir.
- Değer: Değişkenin sakladığı/tuttuğu veridir. Bellekte saklanan veridir.

```
#include <cstdlib>
```

```
#include <iostream>
```

```
using namespace std;
```

```
int main(int argc, char *argv[])
```

```
{
```

```
    int a=8;
```

```
    cout << "a değişkeni..." <<endl;
```

```
    cout << "Değeri   :" << a << endl;
```

```
    cout << "Adresi   :" << &a << endl;
```

```
    system("PAUSE");
```

```
    return EXIT_SUCCESS;
```

```
}
```

# Referans

Aşağıdaki örnekte b bir referans değişkendir ve a ile aynı adresi kullanır. b değişkenine yapılan atamalar a değişkenine yansır. Aynı şekilde a değişkenine yapılan atamalar b değişkenine de yansır.

```
#include <cstdlib>
#include <iostream>
using namespace std;
int main(int argc, char *argv[])
{
    int a=8, &b=a;
    b=25;
    cout << " a: " << a << endl;
    cout << " b: " << b << endl;
    cout << "&a: " << &a << endl;
    cout << "&b: " << &b << endl;
    system("PAUSE");
    return EXIT_SUCCESS;
}
```

# Değer

- Burada a ve b alelade değişkenlerdir. Birbirlerinden etkilenmezler.

```
#include <cstdlib>
```

```
#include <iostream>
```

```
using namespace std;
```

```
int main(int argc, char *argv[])
```

```
{
```

```
    int a=8, b=a;
```

```
    b=25;
```

```
    cout << " a: " << a << endl;
```

```
    cout << " b: " << b << endl;
```

```
    cout << "&a: " << &a << endl;
```

```
    cout << "&b: " << &b << endl;
```

```
    system("PAUSE");
```

```
    return EXIT_SUCCESS;
```

```
}
```

# Fonksiyonlarda Referans ve Değer

- Referansını almak istediğiniz değişkenin adının başına & konur.

```
#include <cstdlib>
#include <iostream>
using namespace std;

void test (int a, int &b)
{
    a=5000;
    b=1500;
}
int main(int argc, char *argv[])
{
    int x=15, y=200;
    test (x, y);
    cout << "x: " << x << endl;
    cout << "y: " << y << endl;
    system("PAUSE");
    return EXIT_SUCCESS;
}
```

# Ör: Takas fonksiyonu

```
#include <cstdlib>
#include <iostream>
using namespace std;

void Takas (int &a, int &b)
{ // a:100 b:999
  int g;
  g=a; // g:100
  a=b; // a:999
  b=g; // b:100
}

int main(int argc, char *argv[])
{
  int x=100, y=999;
  cout << "Takastan once ..." <<endl;
  cout << "x: " << x << endl;
  cout << "y: " << y << endl;
  cout << endl;

  Takas(x, y);
  cout << "Takastan sonra ..." <<endl;
  cout << "x: " << x << endl;
  cout << "y: " << y << endl;

  system("PAUSE");
  return EXIT_SUCCESS;
}
```

# Ör: Diziyi büyükten küçüğe sırala

```
#include <cstdlib>
#include <iostream>
#include <iomanip>
using namespace std;

int EnBuyuk (double Dizi[], int bas, int n)
{
    double test=Dizi[bas];
    int i, yer = bas;
    for(i=bas+1; i<n; i++)
        if (Dizi[i] > test)
        {
            test = Dizi[i];
            yer = i;
        }
    return yer;
}
```

# Ör: Diziyi büyükten küçüğe sırala

```
void Takas (double &a, double &b)
{
    double g=a; a=b; b=g;
}
```

```
void DiziyiYazdir(double A[], int bas, int n)
{
    cout << "Sira Sayi" << endl;
    cout << "----" << endl;
    for(int i=bas; i<n; i++)
        cout << setw(4) << i << " " << A[i] << endl;
}
```



# Ör: Diziyi büyükten küçüğe sırala

```
int main(int argc, char *argv[])
{
    double Dizi[10];
    int i, n=10, yer;
    cout << n << " adet gercel sayi ... " << endl;
    for( i=0; i<n; i++)
    {
        cout << i << ". sayi: ";
        cin >> Dizi[i];
    }
    for(i=0; i<n-1; i++)
    {
        yer = EnBuyuk(Dizi, i, n);
        Takas(Dizi[yer], Dizi[i]);
    }

    DiziyiYazdir(Dizi, 0, n);
    system("PAUSE");
    return EXIT_SUCCESS;
}
```

```
#include <cstdlib>
```

```
#include <iostream>
```

```
using namespace std;
```

```
void Test1(int a) // Değişken veya değer kabul eder
```

```
{  
    a=50; // Ana programa yansımaz  
}
```

```
void Test2(int &a) //değişken ve adresini bekliyor
```

```
{  
    a=100; // Ana programa yansır  
}
```

```
void Test3(int *a) //a bir adrestir
```

```
{  
    *a=250; // a adresine 250 yaz, ana programa yansır  
}
```

```
int main(int argc, char *argv[])
```

```
{  
    int x=192;  
  
    cout<<"Testten ONCE   : "<< x  
    <<endl;  
    Test1(x);  
    cout<<"Test1'den SONRA : "<< x  
    <<endl;  
    Test2(x);  
    cout<<"Test2'den SONRA : "<< x  
    <<endl;  
    Test3(&x); // X'in adresini  
    cout<<"Test3'den SONRA : "<< x  
    <<endl;
```

```
    system("PAUSE");
```

```
    return EXIT_SUCCESS;
```

```
}
```