

BLM111 Programlama Dilleri I

Hafta 12

Fonksiyonlar

Mehmet Zahid YILDIRIM

Fonksiyonlar

► Fonksiyonlar

- C 'de modüller
- Programlar kullanıcı tanımlı fonksiyonları ve kütüphane fonksiyonlarını birlikte kullanırlar.
- C standart kütüphanesi zengin bir fonksiyon çeşitliliğine sahiptir.

Fonksiyonların Faydaları

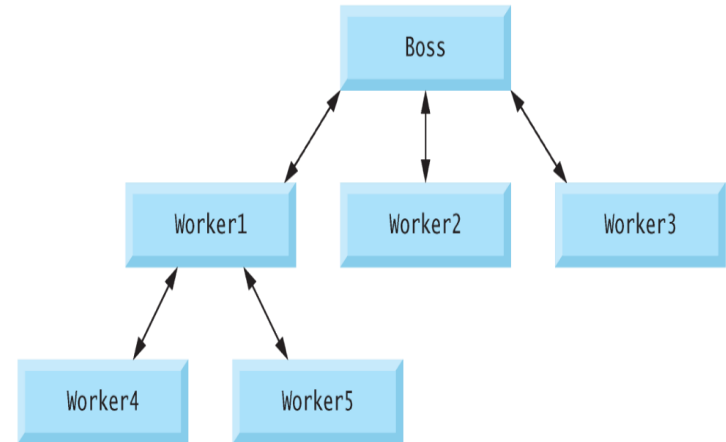
- Fonksiyonları faydaları
 - Böl ve yönet
 - Küçük parçalar veya bileşenlerden program oluştur.
 - Bu küçük parçalara modül denir.
 - Fonksiyonlar bir programı modüler hale getirir.
 - Her bir modül programın kendisinden daha yönetilebilirdir.
 - Yazılımın tekrar kullanılabilirliği
 - Mevcut fonksiyonlarınızı yeni bir program yapımında kullanabilirsiniz
 - Soyutlama – iç detaylar gizlenir(kütüphane fonksiyonları)
 - Kod tekrarı önlenir.

Fonksiyonlar

- Fonksiyonlar içinde tanımlanan tüm değişkenler yerel değişkenlerdir
 - Sadece tanımlandıkları fonksiyon içinde geçerlidirler
- Parametreler
 - Fonksiyonlar ile haberleşmede kullanılan bilgi
 - Yerel değişkendirler
- Fonksiyon çağırılması
 - Fonksiyon ismini ve argümanlarını (veri) belirt.
 - Fonksiyonlar birtakım işlemler gerçekleştirir.
 - Fonksiyonlar sonuç döndürürler

Fonksiyonlar

- Fonksiyon çağırma analogisi
 - Patron işçiden bir işi yapmasını ister
 - İşçi bilgi toplar, işi yapar ve sonucu patrona bildirir.
 - Bilgi gizleme: patron işin detaylarını bilmez.



Fonksiyon Tanımlama

- Fonksiyon tanımlama formatı:

```
geri_dönüş_değer_tipi fonksiyon_adı ( parameter_listesi )  
{  
    tanımlamalar_ve_ifadeler  
}
```

- fonksiyon_adı* herhangi bir geçerli tanımlayıcı olabilir.
- geri_dönüş_değer_tipi* fonksiyonu çağırana döndürülen sonucun veri tipidir.
- geri_dönüş_değer_tipi* void ise fonksiyon herhangi bir değer döndürmez.
- geri_dönüş_değer_tipi*, *fonksiyon_adı* ve *parametre_listesi* üçlüsü fonksiyon başlığı (header) olarak adlandırılır.

Fonksiyon Tanımlama

- *parametre_listesi* fonksiyonun çağırılırken aldığı parametreleri tanımlayan, virgülle birbirinden ayrılmış bir listedir.
- Eğer fonksiyon herhangi bir parametre almıyorsa, parametre listesi **void** olur.
- Her bir parametrenin tipi belirtilmelidir.

Fonksiyon Tanımlama

- Kırılmaç parantezler arasındaki ***tanımlamalar_ve_ifadeler*** fonksiyon gövdesini oluşturur.
- Fonksiyon gövdesinin diğerk bir adı **blok**'tur.
- Değişkenler herhangi bir blok içerisinde tanımlanabilir ve bloklar iç içe olabilir.
- **Bir fonksiyon diğerk bir fonksiyonun içerisinde tanımlanamaz.**

Fonksiyon Tanımlama

- Çağırılan bir fonksiyondan, fonksiyonun çağırıldığı noktaya kontrolün iade edilmesinin üç yöntemi vardır.
- Eğer fonksiyon herhangi bir sonuç döndürmüyorsa,
 - Sağ kurlangıç parantez ile kontrol iade edilmiş olunur.
 - Ya da basitçe `return`; ifadesi çalıştırılır.
- Eğer fonksiyon bir sonuç döndürüyorsa,
`return ifade;`
İfadenin değerini fonksiyonu çağırana döndürür.

Fonksiyon Prototipi

- Bir fonksiyonun künyesidir.
- Eğer fonksiyon tanımlaması çağırımdan sonra ise prototip tanımlanmalıdır.
- Aşağıdaki prototipe sahip fonksiyon
 - `int maximum(int x, int y, int z);`
 - 3 tamsayı parametre alır.
 - Geriye tamsayı döndürür.

Fonksiyon Prototipi

- Bir fonksiyon çağırımı prototipi ile uyuşmuyorsa derleme hatası oluşur.
- Eğer fonksiyon prototipi ile fonksiyon tanımlaması uyuşmuyorsa da hata oluşur.
- Fonksiyon prototiplerinin diğer bir önemli özelliği ise argümanların uygun bir veri tipine zorlanmasıdır.
- Örneğin, matematik fonksiyonu `sqrt` <math.h> içerisinde yer alan prototipinde `double` belirtilmiş olsa da `integer` argümanla da çağırılabilir. Fonksiyon yine de doğru çalışacaktır
 - `printf("%.3f\n", sqrt(4));`
 - İfadesindeki `sqrt(4)` doğru bir şekilde değerlendirilir ve 2.000 değeri yazdırılır.

Fonksiyon Tanımlama

```
/* finding maximum of three integers */
#include <stdio.h>

int max(int x, int y, int z); //function prototyp

int main()
{
    int a, b, c;
    printf("plesase enter three numbers");
    scanf("%d%d%d",&a,&b,&c);
    printf("the maximum number is %d",max(a,b,c));
    return 0;
}

int max(int x, int y, int z)
{
    int maximum;
    if(x>y)
    {
        if(x>z)
            maximum=x;
        else
            maximum=z;
    }
    else if(y>z)
        maximum=y;
    else
        maximum=z;

    return maximum;
}
```

Başlık Header Dosyaları

- Kütüphane fonksiyonlarının prototiplerini barındırırlar.
- `<stdlib.h>` , `<math.h>` , vs
- `#include <dosya_adı>` ile yüklenir.
 - `#include <math.h>`
- Özel başlık dosyaları
 - Fonksiyonlar içeren bir dosya oluştur.
 - `dosya_adı.h` şeklinde isim ile kaydet.
 - Başka dosyalar içerisinde `#include “dosya_adı.h”` olarak yükle.
 - Fonksiyonları tekrar kullan.

Başlık Header Dosyaları

- **math.h** → Matematik kütüphanesi
- **ctype.h** → Karakter özellikleri, küçük büyük harfe çevirme vs.
- **stdio.h** → Standart giriş / çıkış fonksiyonları
- **stdlib.h** → Sayıyı metne metni sayıya dönüştürme, hafıza yönetimi, rasgele sayılar ve bazı diğer faydalı fonksiyonlar.
- **string.h** → String işlemleri
- **time.h** → Zaman ve tarih fonksiyonları

Matematik Fonksiyonları

- Matematik kütüphane fonksiyonları
 - Temel matematik hesaplamaları yapar.
 - `#include <math.h>`
- Fonksiyonları çağırmak için kullanılacak format
 - `FonksiyonAdı (argüman1);`
- Eğer birden fazla argüman varsa, aralarında virgül kullan
- Tüm matematik fonksiyonlar `double` veri tipi döndürür
- Argümanlar sabit, değişken veya ifade olabilir

Matematik Fonksiyonları

Function	Description	Example
<code>sqrt(x)</code>	square root of x	<code>sqrt(900.0)</code> IS 30.0 <code>sqrt(9.0)</code> IS 3.0
<code>exp(x)</code>	exponential function e^x	<code>exp(1.0)</code> IS 2.718282 <code>exp(2.0)</code> IS 7.389056
<code>log(x)</code>	natural logarithm of x (base e)	<code>log(2.718282)</code> IS 1.0 <code>log(7.389056)</code> IS 2.0
<code>log10(x)</code>	logarithm of x (base 10)	<code>log10(1.0)</code> IS 0.0 <code>log10(10.0)</code> IS 1.0 <code>log10(100.0)</code> IS 2.0
<code>fabs(x)</code>	absolute value of x	<code>fabs(13.5)</code> IS 13.5 <code>fabs(0.0)</code> IS 0.0 <code>fabs(-13.5)</code> IS 13.5
<code>ceil(x)</code>	rounds x to the smallest integer not less than x	<code>ceil(9.2)</code> IS 10.0 <code>ceil(-9.8)</code> IS -9.0
<code>floor(x)</code>	rounds x to the largest integer not greater than x	<code>floor(9.2)</code> IS 9.0 <code>floor(-9.8)</code> IS -10.0

Matematik Fonksiyonları

Function	Description	Example
<code>pow(x, y)</code>	x raised to power y (x^y)	<code>pow(2, 7)</code> is 128.0 <code>pow(9, .5)</code> is 3.0
<code>fmod(x, y)</code>	remainder of x/y as a floating-point number	<code>fmod(13.657, 2.333)</code> is 1.992
<code>sin(x)</code>	trigonometric sine of x (x in radians)	<code>sin(0.0)</code> is 0.0
<code>cos(x)</code>	trigonometric cosine of x (x in radians)	<code>cos(0.0)</code> is 1.0
<code>tan(x)</code>	trigonometric tangent of x (x in radians)	<code>tan(0.0)</code> is 0.0

Örnek: Kare alan fonksiyon

```
#include <stdio.h>
float kareAl(float);

void main()
{
    int sayac;
    for(sayac = 1; sayac<=10; sayac++)
    {
        printf("Sayi:%d  Karesi:%d\n", sayac, kareAl(sayac));
    }

    printf("\n%.2f", kareAl(4.5));
}

float kareAl(float a)
{
    return a*a;
}
```

Örnek: Dört İşlem

```
#include <stdio.h>
int topla(int, int);
int cikar(int, int);
int carp(int, int);
float bol(int, int);

void main()
{
    int secim,s1,s2;
    while(1)
    {
        printf("1-Topla\n2-Cikar\n3-Carp\n4-Bol\n5-Cikis\n");
        scanf("%d", &secim);
        printf("Sayilari gir:");
        scanf("%d %d", &s1, &s2);

        if(secim == 1)
            printf("Sonuc = %d", topla(s1,s2));
        else if(secim == 2)
            printf("Sonuc = %d", cikar(s1,s2));
        else if(secim == 3)
            printf("Sonuc = %d", carp(s1,s2));
        else if(secim == 4)
            printf("Sonuc = %.2f", bol(s1,s2));
        else if(secim == 5)
            exit(0);
        else printf("Yanlis giris");
    }
}
```

```
int topla(int a, int b)
{
    return a+b;
}

int cikar(int a, int b)
{
    return a-b;
}

int carp(int a, int b)
{
    return a*b;
}

float bol(int a, int b)
{
    return (float)a/b;
}
```

Örnek: Üs alma

```
#include <stdio.h>
double usAl(double, double);

void main()
{
    double a,b;
    printf("Taban ve us degeri gir:");
    scanf("%lf %lf", &a, &b);
    printf("%.2f", usAl(a,b));
}

double usAl(double x, double y)
{
    int sayac;
    double sonuc=1.0;
    for(sayac=0;sayac<y;sayac++)
    {
        sonuc *= x;
    }
    return sonuc;
}
```

Kaynaklar

- ▶ Doç. Dr. Fahri Vatansever, “Algoritma Geliştirme ve Programlamaya Giriş”, Seçkin Yayıncılık, 12. Baskı, 2015.
- ▶ J. G. Brookshear, “Computer Science: An Overview 10th Ed.”, Addison Wisley, 2009.
- ▶ Kaan Aslan, “A’dan Z’ye C Klavuzu 8. Basım”, Pusula Yayıncılık, 2002.
- ▶ Paul J. Deitel, “C How to Program”, Harvey Deitel.
- ▶ Bayram AKGÜL, C Programlama Ders notları