Bölüm 3: Örnek Fonksiyonlar

JAVA ile Nesne Yönelimli Programlama

Verilen Bir Sayı Tek Mi Çift Mi Bulma

- Tek Sayılar: 1, 3, 5, 7, 9... gibi sayılar, 2 ile bölündüğünde kalan 1'dir.
- Çift Sayılar: 2, 4, 6, 8, 10... gibi sayılar, 2 ile tam bölünebilir.
- Tek mi Çift mi Bulma Algoritması
  - Kullanıcıdan bir sayı alınır.
  - Sayı 2 ile bölündüğünde kalan 0 ise, sayı çifttir.
  - Kalan 1 ise, sayı tektir.
  - Koşul: sayi % 2 == 0 ise çift, değilse tek.

Verilen Bir Sayı Tek Mi Çift Mi Bulma

```
// verilen sayının tek mi yoksa çift mi olduğunu kontrol eder
void tekMiCiftMi(int sayi) {
  // Eğer sayının 2'ye bölümünden kalan 0 ise sayı çifttir
  if(sayi % 2 == 0) {
    System.out.println("sayı çift"); // Sayının çift olduğu yazdırılır
  }
  // Aksi halde sayı tek olacaktır
  else {
    System.out.println("sayı tek"); // Sayının tek olduğu yazdırılır
  }
}
```

Basamak Sayısı Bulma

- Bir sayının basamak sayısı, sayının içindeki toplam rakam sayısıdır.
- Örneğin:
  - 123 sayısı 3 basamaklıdır (1, 2, 3).
  - 45 sayısı 2 basamaklıdır (4, 5).
- Basamak Sayısını Bulma Algoritması
  - Kullanıcıdan bir sayı alınır.
  - Sayı 0'dan büyük olduğu sürece, sayının her bir basamağı silinir.
  - Her bölme işleminde basamak sayısı bir artırılır.
  - Koşul: while (sayi > 0).

Basamak Sayısı Bulma

```
int basamakSayisiBul(int sayi) {
  int basamakSayisi = 0; // Basamak sayısını tutacak değişken
  // Sayı 0'dan büyük olduğu sürece döngü devam eder
  while(sayi > 0) {
    // Sayı her döngüde 10'a bölünerek bir basamak eksiltilir
    sayi = sayi / 10;
    // Her bölme işleminde bir basamak eksildiği için sayacı arttır
    basamakSayisi++;
  }
  // Basamak sayısını döndürür
  return basamakSayisi;
}
```

Verilen Bir Sayının Rakamlar Toplamı

- Bir sayının rakamları toplandığında elde edilen sonuç.
- Örneğin:
  - 123 sayısının rakamları toplamı: 1 + 2 + 3 = 6
  - 456 sayısının rakamları toplamı: 4 + 5 + 6 = 15
- Rakamlar Toplamını Bulma Algoritması
  - Kullanıcıdan bir sayı alınır.
  - Sayı 0'dan büyük olduğu sürece,
    - sayının son basamağını alarak toplamına eklenir.
  - Sayının son basamağını silmek için sayı 10'a bölünür.
  - Koşul: while (sayi > 0).

Verilen Bir Sayının Rakamlar Toplamı

```
int rakamlarToplaminiBul(int sayi) {
  int rakamlarToplami = 0; // Rakamların toplamını tutacak değişken
  // Sayı 0'dan büyük olduğu sürece döngü devam eder
  while(sayi > 0) {
    // Sayının son rakamını almak için 10'a göre mod işlemi yapılır
    int rakam = sayi % 10;
    // Rakamlar toplamına bu rakam eklenir
    rakamlarToplami += rakam;
    // Sayı her seferinde 10'a bölünerek bir basamağı eksiltilir
    sayi = sayi / 10;
  }
  // Tüm rakamlar toplandıktan sonra sonucu döndürür
  return rakamlarToplami;
}
```

Verilen Bir Sayının Üssünü Alma

- Bir sayının kendisiyle belirli sayıda çarpılması işlemine denir.
- Örneğin:
  - 2^3 = 2 × 2 × 2 = 8
  - 5^2 = 5 × 5 = 25
- Üstü Alma Algoritması
  - Kullanıcıdan bir taban ve üs alınır.
  - Sonuç değişkeni başlangıçta 1 olarak ayarlanır.
  - Üs kadar döngü oluşturulur.
  - Her döngüde sonuç değişkeni ile taban çarpılır.

Verilen Bir Sayının Üssünü Alma

```
int usAl(int taban, int us) {
  int sonuc = 1; // Sonuc değişkeni 1 ile başlar.
  // Döngü, üs sayısı kadar döner ve her adımda sonucu taban ile çarpar
  for(int i = 0; i < us; i++) {
    sonuc *= taban; // Sonuc her adımda taban ile çarpılır
  }
  // Hesaplanan sonuç döndürülür
  return sonuc;
}
```

Faktöriyel Hesaplama

- Pozitif bir tam sayının, kendisinden küçük pozitif tam sayıların çarpımıdır.
- Matematiksel notasyon: n! (n faktöriyel)
- Örneğin:
  - 5! = 5 × 4 × 3 × 2 × 1 = 120
  - 3! = 3 × 2 × 1 = 6
- Faktöriyel Hesaplama Algoritması
  - Kullanıcıdan bir sayı alınır.
  - Faktöriyel başlangıçta 1 olarak ayarlanır.
  - Sayı 0'dan büyük olduğu sürece döngü çalışır,
  - Her döngüde faktöriyel değişkeni ile sayı çarpılır, ardından sayı azaltılır.

Faktöriyel Hesaplama

```
int faktoriyelAl(int sayi) {
  int sonuc = 1; // Faktöriyel sonucunu tutacak değişken
  // Sayı 0'dan büyük olduğu sürece döngü devam eder
  while(sayi > 0) {
    // Sonuç değişkeni her adımda sayi ile çarpılır
    sonuc *= sayi;
    // Sayı her seferinde 1 azaltılır
    sayi--;
  }
  // Faktöriyel hesaplandıktan sonra sonuç döndürülür
  return sonuc;
}
```

Asal Sayı Bulma

- 1 ve kendisi dışında pozitif tam böleni olmayan sayılara asal sayı denir.
- Örnekler: 2, 3, 5, 7, 11, 13...
- Not: 1 asal sayı değildir, 2 ise tek asal sayıdır.
- Asal Sayı Bulma Algoritması
  - Kullanıcıdan bir sayı alınır.
  - 2'den başlayarak, verilen sayıya kadar olan her sayıya tam bölünüp bölünmediği kontrol edilir.
  - Eğer sayı herhangi bir bölenle tam bölünmüyorsa, asal sayıdır.

Asal Sayı Bulma

```
boolean asalMi(int sayi) {
  // 2'den başlayarak sayının kendisinden bir küçük sayıya kadar tam bölünebilir olup olmadığını kontrol et.
  for(int i = 2; i < sayi; i++) {
    // Eğer sayı herhangi 'i' değerine tam bölünüyorsa, asal değildir.
    if(sayi % i == 0) {
      return false; // Sayı asal değil, false döner
    }
  }
  // Eğer döngü sonunda hiç bölünen olmadıysa, sayı asaldır.
  return true;
}
```

Öğrenci Numarasını Çözme

- Öğrenci numarası, her öğrenciyi tanımlamak için kullanılan tekil kimliktir.
- Genellikle bir dizi rakamdan oluşur ve öğrencinin kayıt bilgilerini içerir.
- Öğrenci numarası, öğrencinin kayıt bilgilerini hızlı bir şekilde çözümler.
- Öğrenci Numarasının Yapısı
  - İlk iki hane: Kayıt yılı (örneğin, 24, 23 vb.)
  - Beşinci hane: Bölüm kodu (0-9 arası, belirli bölümleri temsil eder)
  - Son üç hane: Kayıt sırası (her bölümdeki kayıt sırası numarası)

Öğrenci Numarasını Çözme

```
int kayitYili(int sayi) {
  //  Öğrenci  numarasının  ilk  iki  hanesini  (numaranın  başındaki  iki
rakamı) alır ve 2000 yılına ekler
  return (sayi / 10000000) + 2000;
}


int kayitSirasi(int sayi) {
  // Numaranın son üç rakamını almak için 1000'e göre mod işlemi yapılır
  return sayi % 1000;
}
```

Öğrenci Numarasını Çözme

```
String bolumu(int sayi) {
  // Numaranın sondan dördüncü hanesini al
  int bolum = (sayi / 1000) % 10;
  // Bölüm numarasına göre hangi bölümde okuduğunu kontrol et
  if(bolum == 7) {
    return "Elektrik Elektronik";
  } else if(bolum == 6) {
    return "Bilgisayar";
  } else if(bolum == 5) {
    return "Makina";
  } else {
    return "Bilinmiyor"; // Uymayan değer için bilinmiyor sonucu döner
  }
}
```

Fibonacci Serisi Yazdırma

- Fibonacci serisi, her sayının kendisinden önceki iki sayının toplamı olduğu bir sayı dizisidir.
- Seri genellikle 1 ve 1 ile başlar: 1, 1, 2, 3, 5, 8, 13, 21, ...
- Matematiksel olarak tanımı:
  - F(0) = 1
  - F(1) = 1
  - F(n) = F(n-1) + F(n-2) (n ≥ 2)
- Doğada pek çok yerde karşımıza çıkar (bitki yapraklarının düzeni).
- Altın oran ile bağlantılıdır; dizinin ilerleyen sayıların oranı 1.618'e yaklaşır.

Fibonacci Serisi Yazdırma

```
void fibonacci(int sayi) {
  int sonuc = 0; // Her adımda bulunan Fibonacci sayısını tutar
  int sayi1 = 1; // Fibonacci serisindeki ilk sayı
  int sayi2 = 1; // Fibonacci serisindeki ikinci sayı
  // Döngü, 2'den başlar, kullanıcının girdiği sayıya kadar
  for(int i = 2; i < sayi; i++) {
    // Her adımda önceki iki sayının toplamı yeni sonuç olur
    sonuc = sayi1 + sayi2;
    // İkinci sayı, birinci sayının yerine geçer
    sayi1 = sayi2;
    // Hesaplanan sonuç ise ikinci sayının yerine geçer
    sayi2 = sonuc;
  }
  System.out.print(sonuc + " ");
}
```

SON