Metotlar II

public static int FonksiyonAdı(String a, double b) {  
 // kod  
}

public, erişim belirleyicisidir. Bununla ilgili daha fazla bilgiyi ilerleyen bölümlerde öğreneceksiniz.

static, metodu sınıfa ait olduğunu ve sınıfın bir örneği olmadığını belirtir.

int, dönüş tipidir. Olası dönüş tipleri int, double, String, boolean, void, vb. olabilir. FonksiyonAdı, fonksiyonun adıdır.

String a, double b, parametrelerdir. İstediğiniz kadar parametreniz olabilir, ancak virgülle ayrılmalıdır. Parametreler, fonksiyona iletilen değişkenlerdir. Bu değişkenleri fonksiyonda kendi içinde tanımlanmış gibi kullanabilirsiniz.

* Parametre isimleri ve argüman isimleri farklı olabilir.

public static void main(String[] args) {  
 int a = 5;  
 int b = 4;  
  
 // a, b sum fonksiyonuna argüman olarak iletiliyor  
 sum(a, b);  
}  
  
// parametre isimleri x, y  
public static int sum(int x, int y){  
 return x+y;  
}

* Java’da fonksiyonlar aşırı yüklenebilir (overloaded). Bu, aynı isme sahip ancak farklı parametrelere sahip birden çok fonksiyon olabileceği anlamına gelir.

public static void main(String[] args) {  
 int a = 5;  
 int b = 4;  
 int c = 1;  
  
 sum(a, b); // A'yı çağırır  
 sum(a, b, c); // B'yi çağırır  
}  
  
// A  
public static int sum(int x, int y){  
 return x+y;  
}  
// B  
public static int sum(int x, int y, int z){  
 return x+y+z;  
}

**Eğlenceli ekstra bilgi**: println(), System sınıfında aşırı yüklenen bir metoddur. Aşağıdaki gibi yazabilirsiniz:

System.out.println("aynı"); //String  
System.out.println(24); //Integer  
System.out.println(24.6); //Double  
System.out.println(a==b); //Boolean

* Java’da, ilkel tipler (int, char, double, float vb.) değer ile geçirilir. Nesneler ise referans ile geçirilir. (referans ile geçirme hakkında daha sonra öğreneceksiniz)

public static void main(String[] args) {  
 int a = 5;  
 int b = 4;  
  
 // a, b değişkenleri yok edilemez  
 destroyer(a, b);  
  
 System.out.println(a); // 5  
 System.out.println(b); // 4  
}  
  
public static void destroyer(int a, int b){  
 // a, b'nin yerel kopyalarını değiştiriyoruz  
 a = 0;  
 b = 0;

# Egzersizler

1. Aşağıdaki döngüde hangi ifadeyi kullanmalıyız ki girdisinin tersini döndürecek şekilde fonksiyon tamamlansın?

public class punchcard {  
 public static String tersCevir(String s) {  
 String res = "";  
 for(int i=0; i<s.length(); i++) {  
 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
 }  
 return res;  
 }  
  
 public static void main(String[] args) {  
 String s = "Merhaba Dünya!";  
 System.out.println(tersCevir(s));  
 }  
}

1. res += s.charAt(i);
2. res = s.charAt(i) + res;
3. s.charAt(i) = s.charAt(s.length()-i);
4. s.charAt(i) = s.charAt(s.length()-i-1);

*cevap*: b - res = s.charAt(i) + res;

1. Bu programı çalıştırdığımızda ekrana ne yazdırılır?

public class punchcard {  
 public static void swap(int a, int b) {  
 int temp = a;  
 a = b;  
 b = temp;  
 }  
 public static void main(String[] args) {  
 int a = 5;  
 int b = 2;  
 swap(a, b);  
 System.out.println(a);  
 }  
}

1. 0
2. 2
3. 5
4. Null
5. hata

*cevap*: c - 5

1. Aşağıdaki fonksiyonda ne yanlış?

public class punchcard {  
 public static String birlestir(String s1, String s2) {  
 // iki string'i birleştirir  
 System.out.println(s1 + s2);  
 }  
 public static void main(String[] args) {  
 // ana metod  
 birlestir("Merhaba", "Dünya");  
 }  
}

1. Yorumlar fonksiyonlar içinde olamaz.
2. static fonksiyonlar parametre alamaz.
3. Dönüş tipi String, ancak fonksiyon hiçbir şey döndürmüyor.
4. + operatörü stringleri birleştiremez.

*cevap*: c - Dönüş tipi String, ancak fonksiyon hiçbir şey döndürmüyor.

1. Aşağıdaki programdaki ne yanlış?

public class punchcard {  
 public static int min(int a, int b) {  
 if(a < b)  
 return a;  
 else  
 return b;  
 }  
 public static void main(String[] args) {  
 int a = 3;  
 int b = 2;  
 int c = 5;  
 a = min(a,b,c);  
 }  
}

1. If-else yapısı süslü parantez içermiyor.
2. min() fonksiyonunun parametre leri main içindeki a, b değişkenleriyle çakışıyor.
3. a değişkeni atama operatörünün her iki tarafında da bulunuyor, bu da tanımsız davranışa neden oluyor.
4. min fonksiyonuna iki değer yerine üç değer verilmeli.
5. Aşağıdaki programdaki ne yanlış?

public class punchcard {  
 public static innocentFunction() {  
 System.out.println("Ben masumum.");  
 }  
 public static void main(String[] args) {  
 innocentFunction();  
 }  
}

1. Aynı kapsam içinde iki static fonksiyon olamaz.
2. Bir fonksiyonun dönüş tipi olmalıdır.
3. Bir fonksiyonun içinde birden fazla ifade olmalıdır.
4. Bir fonksiyon bir şey döndürmelidir.

*cevap*: b - Bir fonksiyonun dönüş tipi olmalıdır.

1. Java’da bir metot ismi aşağıdakilerden hangisiyle başlayamaz?
2. Sayı ile
3. #
4. - (tire)
5. Yukarıdakilerin hepsi

*cevap*: d - Yukarıdakilerin hepsi

1. Aşağıdaki kod ne yapar?

public class punchcard {  
 public static int gizem(int a, int b) {  
 if(b==0)  
 return a;  
 else  
 return gizem(a, b-1) + 1;  
 }  
 public static void main(String[] args) {  
 System.out.println(gizem(2, 25));  
 }  
}

1. Yinelemeli çıkarma ile iki sayıyı çıkarır.
2. İki sayıyı bir dizi işlemiyle çarpar.
3. İki sayıyı çok verimsiz bir şekilde toplar.
4. İki sayıyı iteratif bölme ile böler.

*cevap*: Çok verimsiz bir şekilde iki sayıyı toplar.

1. Fonksiyonumun pozitif tamsayıların çarpımını hesaplamasını istiyorum. Boşluğa ne koymalıyım?

public class punchcard {  
 public static int carp(int a, int b) {  
 if(a == 1)  
 \_\_\_\_\_\_\_\_\_  
 else  
 return carp(a-1, b) + b;  
 }  
  
 public static void main(String[] args) {  
 System.out.println(carp(3, 5));  
 }  
}

1. return 1;
2. return 0;
3. return a;
4. return b;

*cevap*: d - return b;

1. Aşağıdaki programı çalıştırdığımızda ne olur?

public class punchcard {  
 public static void a() {  
 b();  
 }  
 public static void b() {  
 a();  
 }  
  
 public static void main(String[] args) {  
 a();  
 }  
}

1. Program, herhangi bir çıktı vermeden sonsuz bir döngüye girer. (StackOverflowError)
2. Program başarılı bir şekilde çalışır ve bir çıktı verdikten sonra sonlanır.
3. Özyinelemeli metod çağrıları nedeniyle program derleme hatası ile karşılaşır.
4. Program herhangi bir hata ile karşılaşmaz, ancak çalışmaz.

*cevap*: a - Program, herhangi bir çıktı vermeden sonsuz bir döngüye girer. (StackOverflowError)

1. Bir(n) pozitif olmayan n değeriyle çağrıldığında ne olur?

public class punchcard {  
 public static void a(int n) {  
 if(n==0)  
 System.out.println("a() içinde biter");  
 else  
 b(n-1);  
 }  
 public static void b(int n) {  
 if(n==0)  
 System.out.println("b() içinde biter");  
 else  
 a(n-1);  
 }  
  
 public static void main(String[] Args) {  
 a(5);  
 }  
}

1. a() içinde biter
2. b() içinde biter
3. derleme hatası
4. çalışma zamanı hatası

*cevap*: b - b() içinde biter

Eğer n çift ise a() içinde biter, tek ise b() içinde biter. Nedenini görebiliyor musunuz?

Şöyle varsayalım n=5  
  
a(5) --> b(4) --> a(3) --> b(2) --> a(1) --> b(0) "b() içinde biter" yazdırır

1. Aşağıdaki kodun çıktısı nedir?

public class punchcard {  
 public static void Fonksiyon(int a) {  
 System.out.print("Girdi integer " + a);  
 }  
 public static void Fonksiyon(double a) {  
 System.out.print("Girdi double " + a);  
 }  
  
 public static void main(String[] Args) {  
 Fonksiyon(2);  
 }  
}

1. Girdi integer 2
2. Girdi double 2.0
3. Girdi integer 2 Girdi double 2.0
4. hata

*cevap*: a - Girdi integer 2

1. Son kodda ilk fonksiyonu kaldırırsak ne olur? Çıktı ne olur?

public class punchcard {  
 public static void Fonksiyon(double a) {  
 System.out.println("Girdi double " + a);  
 }  
  
 public static void main(String[] Args) {  
 Fonksiyon(2);  
 }  
}

1. Girdi integer 2
2. Girdi double 2.0
3. Girdi integer 2 Girdi double 2.0
4. hata

*cevap*: b - Girdi double 2.0