Yöntemler

Bir yöntem, belirli bir görevi yerine getiren ve sonucu çağıran bir dizi ifadeden oluşur. Bir yöntem, bir şey döndürmeden belirli bir görevi yerine getirebilir. Yöntemler, kodu yeniden yazmadan kodu yeniden kullanmamıza izin verir. Java’da, her yöntem C, C++ ve Python gibi dillerden farklı olarak bir sınıfın bir parçası olmalıdır. Yöntemler zaman kazandırıcıdır ve kodu yeniden yazmadan kodu yeniden kullanmamıza yardımcı olur. Java’da, her yöntem C, C++ ve Python gibi dillerden farklı olarak bir sınıfın bir parçası olmalıdır.

public class MethodExample {  
 public static void main(String[] args) {  
 int a = 10;  
 int b = 20;  
 int c = toplam(a, b);  
 System.out.println("a ve b'nin toplamı " + c);  
 }  
  
 public static int toplam(int sayi1, int sayi2) {  
 int sonuc;  
 sonuc = sayi1 + sayi2;  
 return sonuc;  
 }  
}

Bir yöntem hiçbir şey döndürebilir. Bu durumda, yöntemin dönüş türü void olmalıdır.

public class MethodExample {  
 public static void main(String[] args) {  
 int a = 10;  
 int b = 20;  
 toplam(a, b);  
 }  
  
 public static void toplam(int sayi1, int sayi2) {  
 int sonuc;  
 sonuc = sayi1 + sayi2;  
 System.out.println("a ve b'nin toplamı " + sonuc);  
 }  
}

**Ekstra ama ÖNEMLİ**: Herhangi bir Java projesinde yalnızca bir tane ve yalnızca bir tane public sınıf olmalıdır. public sınıfın adı dosyanın adıyla aynı olmalıdır. Örneğin, dosyanın adı MethodExample.java ise public sınıfın adı MethodExample olmalıdır.

tldr: dosya adı = sınıf adı

public static void main(String[] args) olan sınıf için

1. Çıktıda zaman kazandırıcı fonksiyonlar’ı yazdırmak için boşluğu doldurun.

public class Punchcard {  
 public static void printMessage() {  
 System.out.println("zaman kazandırıcı fonksiyonlar");  
 }  
 public static void main(String[] args) {  
 \_\_\_\_\_\_\_\_\_\_\_\_\_  
 }  
}

1. void(printMessage);
2. printVoid();
3. void();
4. printMessage;
5. printMessage();

*cevap*: E

1. Aşağıdaki kodun neresinde hata var?

public class Punchcard {  
 public static void main(String[] args) {  
 int n = 5;  
 System.out.println(n + " karesi " + ikiye(n));  
 System.out.println(n + " küpü " + uce(n));  
 }  
}  
  
class power {  
 public static int ikiye(int n) {  
 return n \* n;  
 }  
 public static int uce(int n) {  
 return n \* n \* n;  
 }  
}

1. Sınıf adı power olmalıdır
2. Başka bir sınıftan bir yöntemi çağırmak için, sınıf adını kullanmalısınız
3. n değişkeni ikiye ve uce yöntemlerinde bildirilmelidir
4. ikiye ve uce yöntemleri void olarak bildirilmelidir
5. Aynı java dosyasında iki sınıf bulunamaz

*cevap*: B

1. Çıktıda 2 ejderha hayal eti yazdırmak için boşluğu doldurun.

public class Punchcard {  
 public static \_\_\_\_\_\_ printMessage(\_\_\_\_\_) {  
 return x + " ejderha hayal et";  
 }  
 public static void main(String[] args) {  
 System.out.println(printMessage(2));  
 }  
}

1. void ve int
2. int ve int x
3. int ve String x
4. String x ve int
5. String ve int x

*cevap*: E

1. Aşağıdaki kodun çıktısı nedir?

public class Punchcard {  
 public static void main(String[] args) {  
 System.out.println(ortanca(1, 2, 3));  
 System.out.println(ortanca(1, 3, 2));  
 }  
 public static int ortanca(int a, int b, int c) {  
 if (a > b) {  
 if (b > c) {  
 return b;  
 } else if (a > c) {  
 return c;  
 } else {  
 return a;  
 }  
 } else {  
 if (a > c) {  
 return a;  
 } else if (b > c) {  
 return c;  
 } else {  
 return b;  
 }  
 }  
 }  
}

1. 2 ve 2
2. 2 ve 3
3. 3 ve 2
4. Hata, Aynı programda bir yöntemi iki kez çağırmak mümkün değil
5. Hata, ortanca yöntemi void olarak bildirilmelidir

*cevap*: A

1. Aşağıdaki kod neden doğru şekilde çalışmaz?

**Ekstra**: Başka bir sınıftan bir yöntemi çağırmak için, yöntemin static olarak bildirilmesi gerekmektedir

public class Punchcard {  
 public static void main(String[] args) {  
 cat.meow;  
 }  
}  
class cat {  
 public static void eat() {  
 System.out.println("kedi yemek yiyor");  
 }  
 public static void meow() {  
 System.out.println("miyav");  
 }  
}

1. Başka bir sınıftan bir yöntemi çağırmak mümkün değildir
2. meow yöntemi String olarak bildirilmelidir
3. parametre durumu eksik, yöntem adından sonra () olmalıdır (cat.meow())
4. main yönteminden önce cat sınıfı yazılmalıdır
5. Kullanılmayan eat yöntemi kaldırılmalıdır

*cevap*: C

1. Aşağıdaki kodu iki tamsayının toplamını yazdıracak şekilde hangi değişiklik düzeltebilir?

public class punchcard {  
 public static void main(String[] args) {  
 int a = 5;  
 int b = 10;  
 System.out.println("Toplam: " + ekle(a, b));  
 }  
  
 public static int ekle(int x, int y) {  
 return x - y;  
 }  
}

1. return x - y; ifadesini return x + y; olarak değiştirin
2. System.out.println("Toplam: " + ekle(a, b)); ifadesini System.out.println("Toplam: " + cikar(a, b)); olarak değiştirin
3. ekle yöntemini kaldırın
4. int a = 5; ifadesini int a = 15; olarak değiştirin
5. Yeni bir yöntem ekle public static int cikar(int x, int y) { return x + y; }

*cevap*: A

1. Aşağıdaki değişiklik hangi kodu “Merhaba, Java!” şeklinde doğru yazdıracaktır?

public class punchcard {  
 public static void main(String[] args) {  
 greet();  
 }  
  
 public static void greet(String message) {  
 System.out.println(message);  
 }  
}

1. String message = "Merhaba, Java!"; ifadesini greet(); ifadesinden önce ekleyin
2. greet(); ifadesini greet("Merhaba, Java!"); ifadesine değiştirin
3. main yönteminden String[] args ifadesini kaldırın
4. public static void greet(String message) ifadesini public static void greet() olarak değiştirin
5. greet(String message) ifadesini greet() olarak yeniden adlandırın

*cevap*: B

1. Aşağıdaki değişiklik kodu düzgün derleyip çalıştıracaktır?

public class punchcard {  
 public static void main(String[] args) {  
 System.out.println(alinti("Java eğlencelidir."));  
 }  
}  
  
public class Text {  
 public static String alinti(String metin) {  
 return "Alıntı: " + metin;  
 }  
}

1. System.out.println(alinti("Java eğlencelidir.")); ifadesini System.out.println(Text.alinti("Java eğlencelidir.")); olarak değiştirin
2. main yönteminden String[] args ifadesini kaldırın
3. System.out.println(alinti); ifadesinden önce String alinti = alinti("Java eğlencelidir."); ifadesini ekleyin
4. alinti(String metin) yöntemini kaldırın
5. return "Alıntı: " + metin; ifadesini System.out.println("Alıntı: " + metin); olarak değiştirin

*cevap*: A

1. Aşağıdaki değişiklik kodu “5” olarak düzeltecektir?

public class punchcard {  
 public static void main(String[] args) {  
 int sonuc = kare(2);  
 System.out.println(sonuc);  
 }  
  
 public static void kare(int num) {  
 return num \* num + 1;  
 }  
}

1. public static void kare(int num) ifadesini public static int kare(int num) olarak değiştirin
2. main yönteminden String[] args ifadesini kaldırın
3. kare yöntemi içerisinde num \* num; ifadesinin önüne return ekleyin
4. int sonuc = kare(2); ifadesini int sonuc = kare(5); olarak değiştirin
5. int num = 5; ve int sonuc = 0; ifadelerini int sonuc = kare(2); ifadesinden önce ekleyin

*cevap*: A

1. Aşağıdaki değişiklik kodu “Merhaba, Dünya!” olarak düzeltecektir?

public class HelloWorld {  
 public static void main(String[] args) {  
 System.out.println(greet());  
 }  
  
 public static void greet() {  
 return "Merhaba, Dünya!";  
 }  
}

1. System.out.println(greet()); ifadesini System.out.println(HelloWorld.greet()); olarak değiştirin
2. main yönteminden String[] args ifadesini kaldırın
3. public static void greet() ifadesini public static String greet() olarak değiştirin
4. public static void greet() ifadesini public static String greet() { return "Merhaba, Dünya!"; } olarak değiştirin
5. greet() yöntemini kaldırın

*cevap*: C

1. Aşağıdaki değişiklik kodu “true” olarak düzeltecektir?

public class Logic {  
 public static void main(String[] args) {  
 System.out.println(isEven(4));  
 }  
  
 public static boolean isEven(int number) {  
 if (number % 2 == 0) {  
 return "true";  
 } else {  
 return "false";  
 }  
 }  
}

1. public static boolean isEven(int number) ifadesini public static String isEven(int number) olarak değiştirin
2. main yönteminden String[] args ifadesini kaldırın
3. return "true"; ve return "false"; ifadelerindeki tırnakları (") kaldırın
4. System.out.println(isEven(4)); ifadesini System.out.println(Logic.isEven(4)); olarak değiştirin
5. if (number % 2 == 0) ifadesini if (number % 2 != 0) olarak değiştirin

*cevap*: C

1. Aşağıdaki değişiklik kodu “5” olarak düzeltecektir?

public class Numbers {  
 public static void main(String[] args) {  
 int result = add(2, 3);  
 System.out.println(result);  
 }  
  
 public static int add(int a, int b) {  
 a + b;  
 }  
}

1. a + b; ifadesini return a + b; olarak değiştirin
2. main yönteminden String[] args ifadesini kaldırın
3. add yöntemi içindeki a + b; ifadesinin önüne return ekleyin
4. int result = add(2, 3); ifadesini int result = add(5, 5); olarak değiştirin
5. int a = 5; ve int b = 0; ifadelerini int result = add(2, 3); ifadesinden önce ekleyin

*cevap*: C

1. Aşağıdaki değişiklik kodu “10” olarak düzeltecektir?

public class Numbers {  
 public static void main(String[] args) {  
 int result = multiply(2, 5);  
 System.out.println(result);  
 }  
  
 public static int multiply(int x, int y) {  
 return x + y;  
 }  
}

1. return x + y; ifadesini return x \* y; olarak değiştirin
2. main yönteminden String[] args ifadesini kaldırın
3. int result = multiply(2, 5); ifadesini int result = multiply(5, 2); olarak değiştirin
4. int x = 5; ve int y = 5; ifadelerini int result = multiply(2, 5); ifadesinden önce ekleyin
5. public static int multiply(int x, int y) ifadesini public static void multiply(int x, int y) olarak değiştirin

*cevap*: A

1. Boşluğu doldurarak tilki konușsun.

public class Punchcard {  
 public static void main(String[] args) {  
 System.out.println("Tilki ne diyor?");  
 fox.talk();  
  
 }  
}  
class \_\_\_ {  
 public static \_\_\_\_ talk() {  
 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_"Screech"\_;  
 }  
}

Elbette! İşte kodu tamamlamak için beş seçenek:

1. fox, void, return
2. fox, String, System.out.println()
3. fox, void, System.out.println()
4. Animal, void, return
5. Fox, void, System.out.print()

*cevap*: C