

Yöntemler

Bir yöntem, belirli bir görevi yerine getiren ve sonucu çağırılan 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) {  
        -----  
    }  
}
```

- A) void(printMessage);
- B) printVoid();
- C) void();
- D) printMessage;
- E) printMessage();

cevap: E

2. 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;  
    }  
}
```

- A) Sınıf adı power olmalıdır
- B) Başka bir sınıftan bir yöntemi çağırmak için, sınıf adını kullanmalısınız
- C) n değişkeni ikiye ve uce yöntemlerinde bildirilmelidir
- D) ikiye ve uce yöntemleri void olarak bildirilmelidir
- E) Aynı java dosyasında iki sınıf bulunamaz

cevap: B

3. 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));  
    }  
}
```

- A) void ve int
- B) int ve int x
- C) int ve String x

- D) String x ve int
- E) String ve int x

cevap: E

4. 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;
            }
        }
    }
}
```

- A) 2 ve 2
- B) 2 ve 3
- C) 3 ve 2
- D) Hata, Aynı programda bir yöntemi iki kez çağırmak mümkün değil
- E) Hata, ortanca yöntemi void olarak bildirilmelidir

cevap: A

5. 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");
    }
}
```

- A) Başka bir sınıftan bir yöntemi çağırmak mümkün değildir

- B) meow yöntemi `String` olarak bildirilmelidir
- C) parametre durumu eksik, yöntem adından sonra `()` olmalıdır (`cat.meow()`)
- D) main yönteminden önce `cat` sınıfı yazılmalıdır
- E) Kullanılmayan `eat` yöntemi kaldırılmalıdır

cevap: C

6. 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;
    }
}
```

- A) `return x - y;` ifadesini `return x + y;` olarak değiştirin
- B) `System.out.println("Toplam: " + ekle(a, b));` ifadesini `System.out.println("Toplam: " + cikar(a, b));` olarak değiştirin
- C) `ekle` yöntemini kaldırın
- D) `int a = 5;` ifadesini `int a = 15;` olarak değiştirin
- E) Yeni bir yöntem ekle `public static int cikar(int x, int y) { return x + y; }`

cevap: A

7. 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);
    }
}
```

- A) `String message = "Merhaba, Java!";` ifadesini `greet();` ifadesinden önce ekleyin
- B) `greet();` ifadesini `greet("Merhaba, Java!");` ifadesine değiştirin
- C) `main` yönteminden `String[] args` ifadesini kaldırın
- D) `public static void greet(String message)` ifadesini `public static void greet()` olarak değiştirin
- E) `greet(String message)` ifadesini `greet()` olarak yeniden adlandırın

cevap: B

8. 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;  
}  
}
```

- A) `System.out.println(alinti("Java eğlencelidir."));` ifadesini `System.out.println(Text.alinti("Java eğlencelidir."))` olarak değiştirin
B) `main` yönteminden `String[] args` ifadesini kaldırın
C) `System.out.println(alinti);` ifadesinden önce `String alinti = alinti("Java eğlencelidir.");` ifadesini ekleyin
D) `alinti(String metin)` yöntemini kaldırın
E) `return "Alıntı: " + metin;` ifadesini `System.out.println("Alıntı: " + metin);` olarak değiştirin

cevap: A

9. 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;  
    }  
}
```

- A) `public static void kare(int num)` ifadesini `public static int kare(int num)` olarak değiştirin
B) `main` yönteminden `String[] args` ifadesini kaldırın
C) `kare` yöntemi içerisinde `num * num;` ifadesinin önüne `return` ekleyin
D) `int sonuc = kare(2);` ifadesini `int sonuc = kare(5);` olarak değiştirin
E) `int num = 5;` ve `int sonuc = 0;` ifadelerini `int sonuc = kare(2);` ifadesinden önce ekleyin

cevap: A

10. 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!";  
    }  
}
```

- A) `System.out.println(greet());` ifadesini `System.out.println(HelloWorld.greet());` olarak değiştirin
B) `main` yönteminden `String[] args` ifadesini kaldırın
C) `public static void greet()` ifadesini `public static String greet()` olarak değiştirin
D) `public static void greet()` ifadesini `public static String greet() { return "Merhaba, Dünya!"; }` olarak değiştirin
E) `greet()` yöntemini kaldırın

cevap: C

11. 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";
        }
    }
}
```

- A) `public static boolean isEven(int number)` ifadesini `public static String isEven(int number)` olarak değiştirin
 B) `main` yönteminden `String[] args` ifadesini kaldırın
 C) `return "true";` ve `return "false";` ifadelerindeki tırnakları (") kaldırın
 D) `System.out.println(isEven(4));` ifadesini `System.out.println(Logic.isEven(4));` olarak değiştirin
 E) `if (number % 2 == 0)` ifadesini `if (number % 2 != 0)` olarak değiştirin

cevap: C

12. Aşağıdaki değişiklik kodu “5” olarak düzelterektir?

```
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;
    }
}
```

- A) `a + b;` ifadesini `return a + b;` olarak değiştirin
 B) `main` yönteminden `String[] args` ifadesini kaldırın
 C) `add` yöntemi içindeki `a + b;` ifadesinin önüne `return` ekleyin
 D) `int result = add(2, 3);` ifadesini `int result = add(5, 5);` olarak değiştirin
 E) `int a = 5;` ve `int b = 0;` ifadelerini `int result = add(2, 3);` ifadesinden önce ekleyin

cevap: C

13. Aşağıdaki değişiklik kodu “10” olarak düzelterektir?

```
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;
    }
}
```

- A) `return x + y;` ifadesini `return x * y;` olarak değiştirin
 B) `main` yönteminden `String[] args` ifadesini kaldırın
 C) `int result = multiply(2, 5);` ifadesini `int result = multiply(5, 2);` olarak değiştirin

- D) `int x = 5;` ve `int y = 5;` ifadelerini `int result = multiply(2, 5);` ifadesinden önce ekleyin
E) `public static int multiply(int x, int y)` ifadesini `public static void multiply(int x, int y)` olarak değiştirin

cevap: A

14. Boşluğu doldurarak tilki konuşsun.

```
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:

- A) fox, void, return
B) fox, String, System.out.println()
C) fox, void, System.out.println()
D) Animal, void, return
E) Fox, void, System.out.print()

cevap: C