

Μεθοδολογία Προγραμματισμού

Εισαγωγή στη Java - Java κλάσεις

1. (1 Μονάδες) Η λέξη κλειδί **this** χρησιμοποιείται στη Java για να αναφερθούμε στο αντικείμενο που είμαστε. Τι πιστεύετε ότι θα τυπώσει ο παρακάτω κώδικας αν φτιάξουμε ένα αντικείμενο ως εξής; `ClassOne classOne = new ClassOne(100);`.

```
public class ClassOne {  
  
    int x = 3;  
  
    public ClassOne(int x)  
    {  
        System.out.println("Parameter x is: " + x);  
        System.out.println("Field x is: " + this.x);  
    }  
}
```

- A'. Parameter x is:100
Field x is:3
- B'. Parameter x is:100
Field x is:100
- Γ'. Parameter x is:3
Field x is:3
- Δ'. Parameter x is:3
Field x is:100
- Ε'. Καμία απάντηση δεν είναι σωστή

2. (1 Μονάδες) Δείτε τον ακόλουθο κώδικα:

```
public class aClass {
    static int x;
    public void setValue(final int value)
    {
        this.x = value;
    }
}
```

- Α'. Ο κώδικας είναι σωστός αλλά θα υπάρχει προειδοποίηση για τη χρήση του **this**
- Β'. Ο κώδικας δε θα γίνει καν compile.
- Γ'. Ο κώδικας θα γίνει compile αλλά δε θα τρέξει.
- Δ'. Ο κώδικας θα ήταν σωστός αν η παράμετρος value δεν ήταν **final**
- Ε'. Ο κώδικας θα ήταν σωστός αν και η μέθοδος setValue() ήταν **static**

3. (1 Μονάδες) Δείτε τον ακόλουθο κώδικα:

```
public class aClass {
    static int x;
    public static void setValue(final int value)
    {
        this.x = value;
    }
}
```

- Α'. Ο κώδικας είναι σωστός αλλά θα υπάρχει προειδοποίηση για τη χρήση του **this**
- Β'. Ο κώδικας δε θα γίνει καν compile.
- Γ'. Ο κώδικας θα γίνει compile αλλά δε θα τρέξει.
- Δ'. Ο κώδικας θα ήταν σωστός αν η παράμετρος value δεν ήταν **final**
- Ε'. Ο κώδικας θα ήταν σωστός αν και η μέθοδος setValue() δεν ήταν **static**

4. (1 Μονάδες) Δείτε τον ακόλουθο κώδικα:

```
public class ExampleClass {
    static int instance = 0;
    int foo;
    public int getThree() {
        return 3;
    }
    public void setInstance(int value)
    {
        instance = value;
    }
    public int getInstance() {
        return instance;
    }
    public void setValue(int loo) {
        foo = loo;
    }
}
```

Ποιές από τις μεθόδους θα μπορούσαν να χαρακτηριστούν ως **static**;

- Α. Όλες
- Β. Όλες εκτός από τη setValue() και την setInstance().
- Γ. Όλες εκτός από τη setValue().
- Δ. Μόνο η getThree()
- Ε. Καμία

5. (1 Μονάδες) Δείτε τον ακόλουθο κώδικα όπου έχουμε ορίσει έναν μόνο δομητή και ο οποίος είναι **private**:

```
public class ExampleClass {
    private ExampleClass() {
    }
}
```

- Α. Κάτι τέτοιο δεν επιτρέπεται και θα μας βγάλει λάθος ο compiler
- Β. Κάτι τέτοιο δεν επιτρέπεται. Αν προσπαθήσουμε να φτιάξουμε ένα αντικείμενο τύπου ExampleClass και προσπαθήσουμε να τρέξουμε το πρόγραμμα θα πάρουμε κάποιο Exception.
- Γ. Κάτι τέτοιο επιτρέπεται, αλλά έχουμε φτιάξει μια κλάση για την οποία δε μπορούμε να δημιουργήσουμε νέα αντικείμενα.
- Δ. Κάτι τέτοιο επιτρέπεται, απλώς ο compiler αγνοεί την οδηγία **private**
- Ε. Τίποτα από τα παραπάνω

6. (1 Μονάδες) Δείτε το ακόλουθο κώδικα

```
public class ClassFoo {
    private int foo;
    public int getFoo() {
        return foo;
    }
    public void setFoo(int foo) {
        this.foo = foo;
    }
}
public class Swapper {
    public static void swap(int x, int y) {
        int temp = x;
        x = y;
        y = temp;
    }
    public static void swap(ClassFoo classFoo, ClassFoo classTwo)
    {
        ClassFoo temp = classFoo;
        classFoo = classTwo;
        classTwo = temp;
    }
    public static void swap2(ClassFoo classFoo, ClassFoo classTwo)
    {
        int temp = classFoo.getFoo();
        classFoo.setFoo(classTwo.getFoo());
        classTwo.setFoo(temp);
    }
}
public class Main {
    public static void main(String[] args) {
        ClassFoo class1 = new ClassFoo();
        class1.setFoo(100);
        ClassFoo class2 = new ClassFoo();
        class2.setFoo(50);
        System.out.println("Before swap class1.getFoo()="+class1.getFoo() + " "
            class2.getFoo()="+class2.getFoo());
        Swapper.swap(class1.getFoo(), class2.getFoo());
        System.out.println("After swap class1.getFoo()="+class1.getFoo() + " "
            class2.getFoo()="+class2.getFoo());

        System.out.println("Before swap class1.getFoo()="+class1.getFoo() + " "
            class2.getFoo()="+class2.getFoo());
        Swapper.swap(class1, class2);
        System.out.println("After swap class1.getFoo()="+class1.getFoo() + " "
            class2.getFoo()="+class2.getFoo());

        System.out.println("Before swap class1.getFoo()="+class1.getFoo() + " "
            class2.getFoo()="+class2.getFoo());
        Swapper.swap2(class1, class2);
        System.out.println("After swap class1.getFoo()="+class1.getFoo() + " "
            class2.getFoo()="+class2.getFoo());
    }
}
```

Τι πιστεύετε ότι θα τυπώσει;

A'. Before swap class1.getFoo()=100 class2.getFoo()=50
After swap class1.getFoo()=50 class2.getFoo()=100

```
Before swap class1.getFoo()=100 class2.getFoo()=50
After swap class1.getFoo()=100 class2.getFoo()=50
Before swap class1.getFoo()=100 class2.getFoo()=50
After swap class1.getFoo()=50 class2.getFoo()=100
```

Β'. Before swap class1.getFoo()=100 class2.getFoo()=50
After swap class1.getFoo()=100 class2.getFoo()=50
Before swap class1.getFoo()=100 class2.getFoo()=50
After swap class1.getFoo()=100 class2.getFoo()=50
Before swap class1.getFoo()=100 class2.getFoo()=50
After swap class1.getFoo()=50 class2.getFoo()=100

Γ'. Before swap class1.getFoo()=100 class2.getFoo()=50
After swap class1.getFoo()=50 class2.getFoo()=100
Before swap class1.getFoo()=100 class2.getFoo()=50
After swap class1.getFoo()=50 class2.getFoo()=100
Before swap class1.getFoo()=100 class2.getFoo()=50
After swap class1.getFoo()=50 class2.getFoo()=100

Δ'. Before swap class1.getFoo()=100 class2.getFoo()=50
After swap class1.getFoo()=100 class2.getFoo()=50
Before swap class1.getFoo()=100 class2.getFoo()=50
After swap class1.getFoo()=50 class2.getFoo()=100
Before swap class1.getFoo()=100 class2.getFoo()=50
After swap class1.getFoo()=50 class2.getFoo()=100

7. (1 Μονάδες) Τι θα γίνει αν τρέξουμε το πιο κάτω πρόγραμμα;

```
public class Main {
    public static void main(String[] args) {
        ClassFoo class1;
        class1.setFoo(100);
        System.out.println("class1.getFoo()="+class1.getFoo());
    }
}
```

Α'. Θα τυπώσει

```
class1.getFoo()=100.
```

Β'. Θα βγάλει NullPointerException

Γ'. Δε θα κάνει compile καν.

Δ'. Τίποτα από τα παραπάνω.

8. (1 Μονάδες) Δείτε τον ακόλουθο κώδικα.

```
public class Foo {
    private int foo;
    public class Loo {
        private int loo;
        public int getLoo() {
            return loo;
        }
        public void setLoo(int loo) {
            this.loo = loo + foo;
        }
    }
    public int getFoo() {
        return foo;
    }
    public void setFoo(int foo) {
        this.foo = foo;
    }
}
```

Θέλουμε να φτιάξουμε ένα νέο αντικείμενο τύπου Loo σε μια τρίτη κλάση. Πως θα το κάνουμε; (Παρεπιπτόντως προσέξτε πως η εσωτερική κλάση χρησιμοποιεί την ιδιότητα foo της εξωτερικής κλάσης).

Α'. Loo loo = **new** Loo();

Β'. Loo loo = **new** Foo.Loo();

Γ'. Foo.Loo loo = **new** Foo.Loo();

Δ'. Foo foo = **new** Foo(); Loo loo = foo.**new** Loo();

9. (1 Μονάδες) Αν η κλάση Loo ήταν ορισμένη ως **private** θα μπορούσαμε να κά-
νουμε ένα νέο αντικείμενο loo από μια τρίτη κλάση;

Α'. Όχι

Β'. Ναι

10. (1 Μονάδες) Έχετε αυτήν την κλάση

```
public class Foo {
    private int foo;
    private Loo loo;

    public Foo() {
        //Create new Loo object
    }
    public class Loo {
        private int loo;
        public int getLoo() {
            return loo;
        }
        public void setLoo(int loo) {
            this.loo = loo + foo;
        }
    }
    public int getFoo() {

        return foo;
    }
    public void setFoo(int foo) {
        this.foo = foo;
    }
}
```

A'. Loo loo = **new** Loo();

B'. Loo loo = **new** Foo.Loo();

Γ'. Foo.Loo loo = **new** Foo.Loo();

Δ'. Foo foo = **new** Foo(); Loo loo = foo.**new** Loo();

11. (1 Μονάδες) Έχετε αυτό το πρόγραμμα. Τι θα τυπώσει;

```
public class ShadowTest {
    public int x = 0;
    class FirstLevel {
        public int x = 1;
        void methodInFirstLevel(int x) {
            System.out.println("x=" + x);
            System.out.println("this.x=" + this.x);
            System.out.println("ShadowTest.this.x=" + ShadowTest.this.x);
        }
    }

    public static void main(String... args) {
        ShadowTest st = new ShadowTest();
        ShadowTest.FirstLevel fl = st.new FirstLevel();
        fl.methodInFirstLevel(23);
    }
}
```

- A'. x = 23
this.x = 0
ShadowTest.this.x = 0
- B'. Δε θα κάνει compile
- Γ'. x = 1
this.x = 1
ShadowTest.this.x = 1
- Δ'. x = 23
this.x = 0
ShadowTest.this.x = 1
- Ε'. x = 23
this.x = 1
ShadowTest.this.x = 0