

**ΕΡΓΑΣΤΗΡΙΟ 14****Polymorphic behavior via abstract classes**

1. Τι θα εκτυπώσει το πρόγραμμα; (Συμπληρώστε στο παράθυρο Output:)

<pre>public class OOPEercises { static int a = 555; public static void main(String[] args) { A objA = new A(); B objB = new B(); System.out.println("in main(): "); System.out.println("a = "+a); a = 444; System.out.println("objB.a = "+objB.getA()); objA.setA (77777); objB.rollBackA(); System.out.println("After roll back -----"); System.out.println("a = "+a); System.out.println("objA.a = "+objA.getA()); System.out.println("objB.a = "+objB.getA()); } }</pre>	Output: in main(): a = 555 objB.a = 2222 Afer roll back ---- a = 444 objA.a = 77777 objB.a = 333
<pre>public class A { int a = 100; public A() { //System.out.println("in the constructor of class A: "); //System.out.println("a = "+a); a = 333; //System.out.println("a = "+a); } public void setA(int value) { a = value; } public int getA() { return a; } } //class A</pre>	
<pre>public class B extends A { private int a = 123; public B() { a = 2222; } public void rollBackA () { a = super.getA(); } public void setA(int value) { a = value; } public int getA() { return a; } } //class B</pre>	



2. Τι θα εκτυπώσει το πρόγραμμα; (Συμπληρώστε στο παράθυρο Output:)

<pre>public class OOPExercises { static int a = 555; public static void main(String[] args) { A objA = new A(); B objB1 = new B(); A objB2 = new B(); C objC1 = new C(); B objC2 = new C(); A objC3 = new C(); objA.display(); objB1.display(); objB2.display(); objC1.display(); objC2.display(); objC3.display(); } }</pre>	Output: a in A = 100 a in B = 123 a in B = 123 a in C = 543 a in C = 543 a in C = 543
<pre>public class A { int a = 100; public void display() { System.out.printf("a in A = %d\n", a); } } //class A</pre>	
<pre>public class B extends A { private int a = 123; public void display() { System.out.printf("a in B = %d\n", a); } } //class B</pre>	
<pre>public class C extends B { private int a = 543; public void display() { System.out.printf("a in C = %d\n", a); } } //class C</pre>	



3. Τι θα εκτυπώσει το πρόγραμμα;

```
class X {
    void calculate(int a, int b) {
        System.out.println("Class X");
    }
}
class Y extends X {
    @Override
    void calculate(int a, int b) {
        System.out.println("Class Y");
    }
}
class Z extends Y {
    @Override
    void calculate(int a, int b) {
        System.out.println("Class Z");
    }
}
public class MainClass {
    public static void main(String[] args) {
        X x = new Y(); Class Y
        x.calculate(10, 20); Class Y

        Y y = (Y) x; ClassCastException
        y.calculate(50, 100);

        Z z = (Z) y;
        z.calculate(100, 200);
    }
}
```

4. Τι θα εκτυπώσει το πρόγραμμα;

```
class A {
    static void methodOne() {
        System.out.println("AAA");
    }
}
class B extends A {
    static void methodOne() {
        System.out.println("BBB");
    }
}
public class MainClass {
    public static void main(String[] args) {
        A a = new B(); AAA

        a.methodOne(); BBB
    }
}
```



5. Βρείτε το λάθος στον παρακάτω κώδικα;

```
abstract class AbstractClass {  
    private abstract int abstractMethod();  
}
```

6. Τι θα εκτυπώσει το πρόγραμμα;

```
abstract class A {  
    abstract void firstMethod();  
  
    void secondMethod() {  
        System.out.println("SECOND");  
  
        firstMethod();  
    }  
}  
abstract class B extends A {  
    @Override  
    void firstMethod() {  
        System.out.println("FIRST");  
  
        thirdMethod();  
    }  
  
    abstract void thirdMethod();  
}  
class C extends B {  
    @Override  
    void thirdMethod() {  
        System.out.println("THIRD");  
    }  
}  
public class MainClass {  
    public static void main(String[] args) {  
        C c = new C();  
        c.firstMethod();  
        c.secondMethod();  
        c.thirdMethod();  
    }  
}
```

7. Γράψτε ένα πρόγραμμα Java το οποίο να μοντελοποιεί τα κτίρια μιας πόλης. Το πρόγραμμα να υπολογίσει την συνολική φορολογία (φόρος ακίνητης περιουσίας). Ξεκινήστε με το διάγραμμα κλάσεων, και μετά με τον κώδικα.