Object-Oriented Programming
doc. Ing. Valentino Vranić, PhD., ÚISI FIIT STU
Test – May 10, 2018

Last name:	in capital letters
Name:	

1	
2	
3	
4	
5	
6	
7	
8	
9	
10	

The test lasts 30 minutes.

Write the answers into the table. Only the answers in the table will be taken into account. In the multiple-choice questions only one choice is correct.

In case of making corrections, mark clearly which answer is valid. Each correct answer has the value as indicated in the question. An answer that is incorrect, ambiguous, or incomplete will be marked with 0 points. The work out is not considered.

A damaged paper will not be accepted.

1. (1 b) Synchronized methods in Java

(a) alternate uniformly
(b) represent critical regions of a program
(c) start and end simultaneously
(d) prevent a deadlock
(e) represent the threads of a program

2. (1 b) Throwing an exception in Java

(a) corrects the error that occurred
(b) signals an exceptional situation to the main method in the program
(c) frees the memory from unnecessary objects
(d) send a message to the lead programmer
(e) signals an exceptional situation to client code

3. (1 b) A method which is in Java preceded by the **public** keyword

(a) can be called only within the package it belongs to
(b) can be called only within the class hierarchy it belongs to
(c) unlike **private** methods, its source code is publicly available
(d) can be called from anywhere
(e) must be prescribed by an interface which is implemented by the class it belongs to

4. (1 b) In object-oriented programming, encapsulation

(a) represents a criterion for using aggregation
(b) represents a way of making hierarchy
(c) enables to connect objects
(d) enables to decrease the dependency of the client code
(e) enables to apply an object instead of the object of its supertype

5. (1 b) Which design pattern is implemented by this Java code (each class and interface in its own file)?

```
public class A {
   private B b;
   public A(B b) {
      this.b = b;
   }
   public void m(...) {
      ...
         b.op();
      ...
   }
}

public interface B {
   ... op(...);
}

public class B1 implements B {
   public ... op(...) {
      ...
   }
}

public class B2 implements B {
   public ... op(...) {
      ...
   }
}
```

(a) Observer
(b) Visitor
(c) MVC
(d) Composite
(e) Strategy

6. (1 b) A program in Java contains the `M` class. Is it possible to access the object that represents the class itself, and if so, how?

(a) yes, by applying the `M.class` expression
(b) no
(c) yes, by applying the **class**(`M`) expression
(d) yes, by applying the **new** <`M`>() expression
(e) yes, by applying the **new** `M`() expression

7. (1 b) In object-oriented program, the main functionality typically

(a) is contained in constructors
(b) emerges from inheritances
(c) emerges in the interaction of objects
(d) is provided by static methods
(e) is contained within the `main`() method

8. (1 b) An interface instance in Java

(a) is polymorfous
(b) is abstract
(c) is generic
(d) is static
(e) does not exist

9. (1 b) The following class in Java is given:

```
class C implements Serializable {
   public String id;
   public List<C> l = new ArrayList<>();

   public C(String id) {
       this.id = id;
   }
   public static void main(String[] args)
      throws ClassNotFoundException, IOException {

      C x = new C(”x”);
      C y = new C(”y”);
      C z = new C(”z”);
      C w = new C(”w”);

      y.l.add(x);
      y.l.add(w);
      z.l.add(x);
      z.l.add(w);
      w.l.add(x);

      ObjectOutputStream out =
         new ObjectOutputStream(
            new FileOutputStream(”f.out”));
      out.writeObject(y);
      out.close();
   }
}
```

Which objects out of `x`, `y`, `z`, and `w` will be recorded into the `f.out` file (write the corresponding letters; the order is not important)?

10. (2 b) What is the output of the following program in Java?

```
abstract class A {
   public void m() {
      System.out.print(”A”);
   }
}
abstract class S extends A {
   public void m() {
      super.m();
      System.out.print(”S”);
   }
}
class D extends S {
   public void m() {
      System.out.print(”D”);
   }
}
class F extends D {
   public void m() {
      super.m();
      System.out.print(”F”);
   }
}
class M {
   public static void exe(A... o) {
      for (A e : o) {
         e.m();
         System.out.print(”–”);
      }
   }
   public static void main(String[] args) {
      A e1 = (S) new D();
      F e2 = new F();
      S e3 = new D();
      D e4 = new F();

      exe(e1, (D) e2, e3, (A) e4);
   }
}
```

Objektovo-orientované programovanie
doc. Ing. Valentino Vranić, PhD., ÚISI FIIT STU
Test – May 10, 2018

11 b

1 b

2 e

3 d

4 d

5 e

6 a

7 c

8 e

9 x, y, w

10 D-DF-D-DF-