Object-Oriented Programming **A**

doc. Ing. Valentino Vranić, PhD., ÚISI FIIT STU

Exam – June 9, 2017

Last name:

Name:

1 b	
2 b	
3 b	

The exam lasts 70 minutes.

1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	

Write the answers to questions 1–12 into the table. With these questions, only the answers in the table will be considered (without the work out). An answer must be unambiguous and readable, otherwise it will be marked with 0 points.

In multiple-choice questions only one choice is correct – write into the table only the letter by which the answer you choose is marked with.

Write the answer to question 13 exclusively on the paper with the question text.

A damaged paper will not be accepted.

1. (1 b) Class in the C++ language have their roots in the mechanism of

(a) union
(b) template
(c) struct
(d) define
(e) include

2. (1 b) To what is in Java implemented by attributes with accompanying set and get methods, in the C# language corresponds the mechanism of

(a) event
(b) delegate
(c) sealed
(d) var
(e) property

3. (2 b) The following program in Java is given:

```
class A {
    private int a;
    private int b;
    private int c;

    public A(int v) {
        b = v;
        c = 2 * b;
    }
    public void x() {
        c++;
        c--;
    }
    public void y() {
        a--;
        a++;
    }
    public void z() {
        c = 2 * b;
    }
    public void w() {
        synchronized(this) {
            if (c != 2 * b)
                System.out.println("!");
        }
    }
}
class B {
    public void m(A o) {
        new Thread() {
            public void run() {
                for (int i = 1; i < 1000000; i++)
                    o.w();
            }
        }.start();

        new Thread() {
            public void run() {
                for (int i = 1; i < 1000000; i++)
                    o.y();
            }
        }.start();

        new Thread() {
            public void run() {
                for (int i = 1; i < 1000000; i++)
                    o.z();
            }
        }.start();

        new Thread() {
            public void run() {
                for (int i = 1; i < 1000000; i++)
                    o.x();
            }
        }.start();
    }
    public static void main(String[] args) {
        A a = new A(9999);
        B b = new B();
        b.m(a);
    }
}
```

At which methods it is *necessary* to introduce the **synchronized** modifier in order to make sure that the exclamation mark never appears in the output?

4. (2 b) What all makes the output of the System.out.print() statements of the following program in Java (until its successful or unsuccessful ending)?

```
class E extends Exception {}

class M {
    public void m(char c) throws E {
        if (c == 'a')
            System.out.print("A");
        else
            throw new E();
    }
    public void f(char c) throws E {
        System.out.print("F");
        try {
            m(c);
        } catch (E e) {
            throw e;
        } finally {
            System.out.print("!");
        }
    }
    public static void main(String[] args) throws E {
        new M().f('b');
        new M().f('a');
    }
}
```

5. (1 b) The friend mechanism known from the C++ language in Java

(a) does not exist
(b) exist under the same name
(c) exist under the **static** name
(d) exist under the package access name
(e) exist under the generics name

6. (1 b) Java supports persistence by

(a) aggregation
(b) radialization
(c) modularization
(d) serialization
(e) synchronization

7. (1 b) A potential disadvantage of aspects in the AspectJ language is that

(a) they cannot affect the user interface
(b) in the code they affect it is not visible it is affected
(c) they cannot substitute existing methods in classes
(d) they cannot add new elements into classes
(e) there can be only a small number of them

8. (1 b) Generics in Java enables collections to

(a) be specialized for any data type during inheritance
(b) be specialized for any data type during instantiation
(c) perform faster
(d) store a greater number of objects
(e) be automatically saved to the disk

9. (2 b) Extensibility of a class with other operations (methods) should be provided, but so that the adding of operations wouldn't require changing its code. What design pattern would you use?

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

10. (3 b) What is the output of the execution of the following program in Java?

```
class A {
    public void x() {
        System.out.print("Ax");
    }
    public static void y() {
        System.out.print("Ay");
    }
}
class B extends A {
    public void x() {
        super.x();
        System.out.print("Bx");
    }
    public static void y() {
        A.y();
        System.out.print("By");
    }
}
class C extends B {
    public void x() {
        System.out.print("Cx");
    }
    public static void y() {
        System.out.print("Cy");
    }
}
class M {
    public static void main(String[] args) {
        A o1 = new B();
        C o2 = new C();
        B o3 = new C();
        B o4 = new B();
        A o5 = new B();

        o1.x();
        o1.y();
        System.out.print(" ");

        ((B) o2).x();
        ((B) o2).y();
        System.out.print(" ");

        ((C) o3).x();
        ((C) o3).y();
        System.out.print(" ");

        o4.x();
        o4.y();
        System.out.print(" ");

        ((A) o5).x();
        ((A) o5).y();
    }
}
```

11. (2 b) The following part of the code of a game in Java is given:

```
class Control {
    public static void main() {
        ...

        if (move == 'a') { // attack the enemy
            if (player.hasSword()) {
                enemy.decreaseEnergy(10);
            } else {
                player.decreaseEnergy(10);
            }
        } else if...
        ...
    }
}
```

From the perspective of the object-oriented design flexibility, the most important to do would be to

(a) create a graphical user interface
(b) instead of the **if** statement use the **switch**–**case** statement
(c) take the whole control of the game into a non-static method
(d) change the decreaseEnergy() method to be static
(e) take the code under the `a` move into a method independent of control

12. (3 b) The class that represents a special document is devised from the class that represents a general document. The method for identifying the general document signatory does not guarantee returning the name of the signatory because a general document may have no one. By this, preconditions and postconditions of these methods weaken, strengthen, or remain the same? Is Liskov substitution principle (LSP) preserved by this?

Answer in this form: *preconditions / postconditions / LSP*. Items *preconditions* and *postconditions* replace with the one of the following possibilities: *weaken*, *strengthen*, or *remain the same*. Items *LSP* replace with the one of the following possibilities: *preserved* or *not preserved*.

Last name:

Name:

13. (10 b) In a task management system, each task has its name, start of realization (it is sufficient to implement it as an integer: the day from the beginning of the project), assumed duration (it is sufficient to implement it as an integer: the number of days from the beginning of the realization), and description. A task can be a continuation of another task (one task). Apart from others, two summarizations are provided for the stored tasks, which are being automatically updated upon a change of the task data:

- a list of task names with the start of realization
- a list of pairs of tasks that are in the relationship of continuation

Design and implement in Java the corresponding object-oriented solution that takes into account the principles of object-oriented programming. In this, apply the most appropriate design pattern among Strategy, Observer, Visitor, and Composite.

Provide the basic design as a UML class diagram sketch containing the most important relationships, operations, and attributes. In this, take into account the design pattern. Visibility is not required to be provided.

In implementation, focus on the application logic – GUI is not the subject of the question. Also, the algorithms applied do not have to be optimal.

Explicitly identify the elements that model and implement the roles of the design pattern applied and explain why did you apply this design pattern. Present an example of use in which you create the corresponding objects and start off their interaction.

Object-Oriented Programming **A**
doc. Ing. Valentino Vranić, PhD., ÚISI FIIT STU
Exam – June 9, 2017
30

1 c

2 e

3 A.x() a A.z(); A.x() is also accepted as an answer (thread switching occurs rarely during the processing of the expression in method A.z())

4 F!

5 a

6 d

7 b

8 b

9 c

10 AxBxAy CxAyBy CxCy AxBxAyBy AxBxAy

11 e

12 remain the same / strengthen / preserved

The Observer pattern had to be applied in the last question. The subject being observed should have been the object of the class used to store tasks (the Subject role), and the observers should have been the objects of the classes that implement summarizations (the Observer role).

The question would be marked according to the following key:

- providing the basic functionality – 4 b
- a design with respect to the open-closed principle and appropriate use of encapsulation – 6 b

In marking of the both parts, an explanation (including the example of the use) and the corresponding class diagram would be counted in approximately with 10–20% of the points for the corresponding part.