

Last name:

Name:

1 b	
2 b	
3 b	

The exam lasts 70 minutes.

Write the answers to questions 1–11 into the table. With these questions, only 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 questions with offered answers only one possibility is correct – write into the table only the letter by which the answer you choose is marked with.

Write the answer to question 12 exclusively on the paper with the question text.

A damaged paper will not be accepted.

1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	

1. (1 b) In C++, a new object can be created

- (a) exclusively by the **new** operator as in Java
- (b) by the **new** operator as in Java or by defining a variable of the given class type
- (c) not from a class, but from a template
- (d) only from classes that contain no virtual functions
- (e) exclusively by defining a variable of the given class type

2. (3 b) The class that represents a special game is devised from the class that represents the general game. The method for adding a player is overridden in the special game so that it allows for adding a player with any number of points, while in the general game the number of points has to be more than or equal to zero (a negative score is possible, too). 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*.

3. (1 b) In C#

- (a) elements from different files can be a part of one namespace if they are placed in a common folder
- (b) several namespace can be defined in one file, but elements from different files can't be a part of one namespace
- (c) elements from different files can be a part of one namespace, but it's not possible to define several namespaces in one file
- (d) it's possible to define several namespaces in one file and elements from different files can be a part of one namespace
- (e) only one namespace can be defined in one folder

4. (1 b) An advantage of using aspects in AspectJ is that

- (a) accelerate program execution
- (b) better focus the code by concerns

- (c) automatically correct the errors in a program
- (d) improve security
- (e) enable to generate a better documentation than Javadoc

5. (2 b) The following program in Java is given:

```
class C extends Thread {
    private static int x = 'a';
    private static int y = 'b';
    public void ab() {
        x = 'a';
        y = 'b';
    }
    public void ba() {
        x = 'b';
        y = 'a';
    }
    public void t() {
        if (x == y)
            e();
    }
    public void e() {
        System.out.println("=");
    }
}
```

```
class M {
    public static void main(String[] args) {
        C o = new C();

        new Thread() {
            public void run(){
                for (int i = 1; i < 100000; i++)
                    o.ab();
            }
        }.start();

        new Thread() {
            public void run(){
                for (int i = 1; i < 100000; i++)
                    o.ba();
            }
        }.start();

        new Thread() {
            public void run(){
                for (int i = 1; i < 100000; i++)
                    o.t();
            }
        }.start();
    }
}
```

Which methods must be synchronized for the program never to output the = symbol?

6. (1 b) The most similar type of the inheritance in C++ to the inheritance between classes in Java is

- (a) protected
- (b) private
- (c) virtual
- (d) public
- (e) friend

7. (2 b) What is the output of the execution of the following program in Java?

```
class X extends Exception {}

class A {
    public void m(int i) throws X {
        if (i > 0)
            System.out.print("M");
        else
            throw new X();
    }
    public void op(int i) throws X {
        try {
            m(i);
        } catch (X e) {
            throw e;
        }
    }
    public static void main(String[] args) {
        try {
            new A().op(-1);
            new A().op(1);
        } catch (X e) {
            System.out.print("X");
        } finally {
            System.out.print("F");
        }
        System.out.print("A");
    }
}
```

8. (3 b) What is the output of the execution of the following program in Java?

```
interface Y {
    void a();
}
abstract class J implements Y {
    public void a() {
        System.out.print("J");
    }
}
class K extends J {
    public void a(int i) {
        System.out.print("K");
    }
}
class L extends K {
    public void a() {
        super.a();
        System.out.print("L");
    }
}
class A {
    static void s() {
        Y o1 = new L();
        L o2 = new L();
        K o3 = new K();
        J o4 = new K();
        J o5 = new L();

        o1.a();
        o2.a();
        ((J)o3).a();
        o4.a();
        ((L)o5).a();
    }
}
class B extends A {
    static void s() {
        Y o1 = new L();
        K o2 = new K();
        Y o3 = new K();
    }
}
```

```
J o4 = new K();
L o5 = new L();
```

```
o1.a();
((J)o2).a();
o3.a();
((K)o4).a();
o5.a();
}
}
class M {
    public static void main(String[] args) {
        A o = new B();
        o.s();
    }
}
```

9. (2 b) A part of the solution is a particular functional part that can differ in different cases of its application. What design pattern would you use?

- Visitor
- Composite
- Strategy
- Observer
- MVC

10. (3 b) A part of the graphical user interface of a computer game is also the t button (an object of the JButton type), with respect to which the following Java code appears in the game:

```
class GameWindow extends JFrame {
    Player player;
    ...
    class DecreaseEnergy implements ActionListener() {
        public void actionPerformed(ActionEvent e) {
            if (player.hasShield())
                player.setEnergy(player.getEnergy() - 1);
            else
                player.setEnergy(player.getEnergy() - 2);
        }
    }
    ...
    t.addActionListener(new DecreaseEnergy());
    ...
}
```

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

- apply the **private** access modifier to the player attribute
- derive the corresponding classes from the Player class and use polymorphism instead of the **if** statement
- take the code in the actionPerformed() method into the game logic implementation
- change the DecreaseEnergy class into an anonymous one or use a lambda expression for assigning the actionPerformed() method to the t button
- name the t button more appropriately

11. (1 b) In AspectJ, for to be possible to execute some code before a method

- no intervention in it is necessary
- before its name the before keyword has to be applied
- it has to be moved into an aspect
- its name must be preceded by the before prefix
- it has to be marked with the @before annotation

(vypláte tlačným písmom)

Last name:

Name:

12. (10 b) In a university information system, among other things course records are maintained. There are three types of courses: lectured courses with exercises, lectured courses with no exercises, and project courses. Each course type is implemented with its own class. For each course its name, annotation, and number of credits are recorded, but some of the data are specific to each course type. Lectured courses have their lecturer and guaranteeing representative, while project courses have no lecturer, but a guaranteeing representative only. In lectured courses, the number of lecture hours is recorded. In lectured courses with exercises, the number of exercise hours and the list of assistants is recorded in addition.

The system provides a possibility to write out the summary information on courses in a complete and abridged form. In the complete form, all the information is written out, whereby the items irrelevant with respect to the course type are not being written out. In the abridged form, only the name of the course and lecturer and/or guaranteeing representative is written out. In further versions new output forms may be necessary.

Design the corresponding object-oriented solution and implement it in Java. Present the basic design in the form of a class diagram featuring the most important relationships, operations, and attributes.

Apply the appropriate mechanisms of object-oriented programming. If that is appropriate, apply one of the design patterns. Identify explicitly by what are the roles of the applied pattern modeled and implemented and explain what is achieved with this pattern.

Focus on the application logic – GUI is not the subject of the question. Present an example in which you create the corresponding objects and start off their interaction.

1 b**2** weaken / remain the same / preserved**3** d**4** b**5** C.ab(), C.ba() a C.t()**6** d**7** XFA**8** JLJLJJJL**9** c**10** c**11** a

In the last question, the Visitor pattern should have been applied. Course types would feature the Element role, while the types of output would feature the Visitor 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.