Object-Oriented Programming
doc. Ing. Valentino Vranić, PhD., ÚISI FIIT STU
Exam – June 11, 2014

Surname:

Name:

1 b	
2 b	
3 b	

1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	

The time for the exam is 70 minutes.
Write your answers to questions 1–11 into the table. In these questions, only the answers in the table will be considered (without the procedure). The answer must be unambiguous and readable, otherwise it bears 0 points.
In multiple choice questions only one choice is correct. Write only the letter by which the choice you're selecting is marked.
Write the answer to question 12 exclusively to the paper with the text of the question.

1. (1 b) The friend mechanism in C++

(a) does not exist
(b) enables to other classes to access non-public elements of a class
(c) enables to other functions to access non-public elements of a class
(d) is used to turn on or off polymorphism
(e) is used to mark the user-friendly functionality

2. (1 b) Properties in C# correspond to Java

(a) methods in general
(b) attributes with accompanying set/get methods
(c) attributes in general
(d) set/get methods
(e) objects of anonymous classes

3. (1 b) By code outside a given method, in AspectJ it is possible to

(a) just to disable or enable execution of this method, including repeated execution
(b) substitute this method with completely different code, but not to leave out its execution
(c) just to change the return value of this method
(d) substitute this method with completely different code, including leaving out its execution
(e) substitute this method with sufficiently similar code

4. (1 b) Importing a package into the Java program will

(a) enlarge this program by the code from the package only in its compiled form
(b) enlarge this program by the code from the package both in its not compiled and compiled form
(c) enlarge this program by the actually used code from the package
(d) not enlarge the program
(e) enlarge the program by the code from the package in its not compiled form

5. (1 b) Encapsulation in object-oriented programming

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

6. (2 b) The following code in Java is given:

```
interface A {
   void m(X x);
}
interface X {
   ...
}
class C implements A {
   public void m(X x) {
      x.op(this);
   }
}
```

For which design pattern is this code characteristic?

7. (2 b) According to Liskov principle of substitution

(a) when applying inheritance, the code that implements should be considered
(b) when applying inheritance, the client code should be considered
(c) inheritance should be applied as much as possible
(d) when applying inheritance, real or mathematical objects should be considered
(e) inheritance should be applied to the minimal extent

8. (2 b) Method f() of class A throws MyException. Class B is given:

```
class B {
   void m() {
      new A().f();
   }
}
```

Method m() of class B

(a) has to throw an exception of the MyException type
(b) is correct
(c) has to handle an exception of the MyException type
(d) has to declare that it throws an exception of the MyException type
(e) has to declare or throw an exception of the MyException type

9. (3 b) The following Java program is given:

```
class A {
    private static int a = 'a', b = 'b';
    public static void m() {
        if (a == 'a') {
            a = 'b';
            b = 'a';
        }
        else {
            a = 'a';
            b = 'b';
        }
        if (a == b)
            System.out.println("=");
    }
}
class B implements Runnable {
    A a;
    public B(A a) {
        this.a = a;
    }
    public void run() {
        for (int i = 1; i < 100000; i++)
            a.m();
    }
}
class M {
    public static void main(String[] args) {
        A a = new A();
        new Thread(new B(a)).start();
        new Thread(new B(a)).start();
    }
}
```

In order to never output the = character

(a) method m() must not be static
(b) method m() must be synchronized and must not be static, so as attributes a and b
(c) method m() must be synchronized
(d) method m() must be synchronized and must not be static
(e) no change is needed

10. (3 b) A game in Java is controlled by pressing the corresponding keys as follows:

```
    switch (o)
    case 'a':
        if (player1.getEnergy() < 10)
            player2.addObject(player1.takeObject());
        else
            player1.addEnergy(player2.getEnergy());
        break;
    case 'b':
    . . .
```

From the perspective of the object-oriented design, it would be adequate to

(a) take out individual cases into corresponding methods
(b) leave everything as it is
(c) implement mouse control instead of keyboard
(d) apply polymorphism
(e) apply encapsulation

11. (3 b) What is the output of the following Java program?

```
interface I {
    void m();
}
abstract class K implements I {
    public void m() {
        System.out.print("K");
    }
}
class L extends K {
    public void m() {
        super.m();
        System.out.print("L");
    }
}
class M extends L {
    public void m() {
        super.m();
        System.out.print("M");
    }
}
class C {
    public static void exe(I... a) {
        for (I e : a)
            e.m();
    }
    public static void main(String[] args) {
        I a = new K();
        K b = new L();
        I c = new M();
        I d = new L();
        M e = new M();

        exe((I)a, b, c, d, e);
    }
}
```

Surname:

Name:

12. (10 b) Apart from other things, the system stores a list of textual entries. In general, it is possible to filter entries according to various criteria the result of which is the list of records that fulfil the criterion in case. New ways of filtering of any program complexity are expected to be added.

Draw a class diagram with the most important relations, operations, and attributes that are subsumed by the above description of the inner program model (GUI is not considered). Write the corresponding Java code including an example of use employing the respective objects and their interaction. Apply the mechanisms of object-oriented programming. If it is appropriate, apply one of design patterns and explain what is gained by that.

Object-Oriented Programming
doc. Ing. Valentino Vranić, PhD., ÚISI FIIT STU
Exam – June 11, 2014

1 c

2 b

3 d

4 d

5 c

6 Visitor

7 b

8 e

9 c

10 a

11 KKLKLMKLKLM

30