Object-Oriented Programming □
Ing. Valentino Vranić, PhD., IISI FIIT STU
Exam — May 26, 2009

Surname:

Name:

1 b	
2 b	
3 b	

	a	b	c	d	e
1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					
15					
16					

The time for the exam is 90 minutes.
In questions 1–16, only one possibility is correct. Mark your answer by a cross in the table. Only the answers in the table will be taken into consideration.
In case you have to change your answer, clearly mark the answer that holds. Each correct answer has the value stated in the question. An incorrect answer, masking several answers, or unclear marking of the answer has the value of 0 points. The explanation of a solution in questions 1–16 is not taken into account. A damaged sheet will not be accepted.
Write the answers to questions 17 and 18 on the additional sheet. Mark this sheet also with your name.

1. (1 b) The following code is given:

```
for (Object o : l) {
   if (o.class == "TerribleOgre")
      ((TerribleOgre)o).attack();
   else if (o.class == "PlachyObor")
      ((FearfulOgre)o).escape();
   else
      ;
}
```

This code violates

(a) the polymorphism principle
(b) Liskovej substitution principle
(c) the principle of generalization and specialization
(d) the abstraction principle
(e) the open–closed (of code) principle

2. (1 b) An object in object-oriented programming represents

(a) an instance of a class or interface
(b) an instance of a class
(c) a type
(d) a class
(e) a module

3. (1 b) If a class in C++ contains purely virtual functions then

(a) it can't have instances
(b) it can't inherit from the class that doesn't include them
(c) can inherit from another class only virtually
(d) can be inherited from it only virtually
(e) doesn't support polymorphism

4. (1 b) A class atribute preceded with the **protected** keyword

(a) is protected against writing
(b) would not be written into file in the object serialization
(c) is available only in the given class
(d) is available only in the given class hierarchy
(e) is available only in one thread

5. (2 b) The type of inheritance in C++ that corresponds to the extend inheritance between classes in Java is:

(a) **private**
(b) **protected** with virtual classes
(c) **protected**
(d) **public** with virtual classes
(e) **public**

6. (2 b) In the Swing framework, a button click is monitored

(a) by calling the actionPerformed() method of the respective button in a loop
(b) by the onClick() method of the button implemented upon deriving it from the general JButton
(c) by a click event listener registered for the given button
(d) by calling the EventQueue.onClick() static method and by a subsequent check whether the click belongs to the given button
(e) automatically after adding the button to the window by its add() method

7. (2 b) In contrast to object-oriented programming, aspect-oriented programming enables

(a) tangling of crosscutting concerns
(b) faster program execution
(c) splitting code into components
(d) creating modules
(e) separation of crosscutting concerns

8. (2 b) The UML sequence diagram shows

(a) relationships between class instances in a particular moment of the program execution
(b) classes and relationships between them
(c) functionality from the user perspective
(d) a sequence of calls between objects
(e) the system structure

9. (2 b) The open–closed principle states that

(a) code has to be open to changes, but closed to extension
(b) each data stream besides the standard input and output has to be closed after being opened
(c) code has to be open to extension, but closed to changes
(d) code has to be both open and closed for modification
(e) each data stream has to be closed after being opened

10. (2 b) Static method synchronization

(a) is impossible
(b) means to lock the class object for any other synchronized access
(c) means to lock the **this** object for any other synchronized access
(d) means to lock the **this** object for any other access
(e) means to lock the class object for any other access

11. (2 b) In deciding whether to use inheritance, the key issue is to follow

(a) the possibility to employ the subtype objects in place of the supertype objects
(b) the possibility to generalize the subtype by the supertype
(c) the possibility to abstract the subtype by the supertype
(d) the number of common attributes and methods of the subtype and supertype
(e) the extent to which the objects modeled by the subtype and supertype are related in reality

12. (3 b) The implementations of the listner interfaces in Swing represent

(a) observers in the Observer pattern
(b) visitors in the Visitor pattern
(c) elements in the Visitor pattern
(d) concrete strategies in the Strategy pattern
(e) contexts in the Strategy pattern

13. (3 b) After executing the code from Fig. 1

(a) xy will be printed out and the program will throw an exception
(b) yyy will be printed out
(c) the program will throw an exception
(d) yyx will be printed out
(e) yy will be printed out and the program will throw an exception

```
class X {
   public void m() { System.out.print("x"); }
}
class Y extends X {
   public void m() { System.out.print("y"); }
}
class A {
   public Runnable r(final X x) {
      return new Runnable(){
         public void run(){
            x.m();
         }
      };
   }
   public static void main(String[] args) {
      X ox = new Y();
      Y oy = new Y();

      Runnable a = (new A()).r(ox);
      Runnable b = (new A()).r(oy);
      Runnable c = (new A()).r((X)oy);
      a.run();
      b.run();
      c.run();
   }
}
```

Obrázok 1: The code for questions 13 and 15.

14. (3 b) In one of the program classes, a computation is performed based on the coefficients supplied to it. In case of their disagreement, which can be identified by comparing them, the computation makes no sense. This should be handled

(a) directly in the method that realizes the computation by stopping the program due to an error
(b) by throwing a general exception of the Exception type and by handling it in the method that needed the computation
(c) by throwing an exception of the type created specifically for this purpose and by handling it in the method that needed the computation
(d) directly in the method that realizes the computation by an attempt to correct the result
(e) by throwing an exception of the RuntimeException type

15. (3 b) The following class is given to the code in Java in Fig. 1:

```
class M {
   static void m(Class<? extends X> T, X... o) {
      for (X e : o)
         if (T.isInstance(e))
            System.out.print("i");
   }

   public static void main(String... args) {
      m(X.class, new X[]{new X(), new Y()});
   }
}
```

By executing it

(a) ii will be printed out
(b) an exception will arise
(c) i will be printed out
(d) nothing will print
(e) iii will be printed out

16. (3 b) In the aspect-oriented implementation of the Observer pattern

(a) the logic of the pattern that belongs mostly to the Observer part is separated out
(b) the third part of the pattern emerges between Subject and Observer denoted as Mediator
(c) the logic of the pattern that belongs mostly to the Subject part is separated out
(d) ceases the need to implement Subject
(e) the application logic separates from the application interface

17. (5 b) Draw the UML class diagram for the code in Fig. 1 extended by the following class:

```
class C extends X {
   private X x;
   public void setX(X x) { this.x = x; }
   public X getX() { return x; }
}
```

18. (12 b) An application enables to encrypt different kinds of files. Assume binary and text files. In text files, take into account their encoding—let us assume ASCII and Unicode. The application provides two kinds of encryption for all the file types supported. In future, the application will support further ways of encryption, and possibly further file types.

Write the code in Java that implements the files and encryption. In this, apply object-oriented programming mechanisms to the greatest extent and explain their role. Do not introduce the encryption as such, nor file operations.

Objektovo-orientované programovanie $\boxed{\textbf{A}}$
Ing. Valentino Vranić, PhD., ÚISI FIIT STU
Skúška — 26. máj 2009

1 e

2 b

3 a

4 d
—

5 d

6 c

7 e

8 d

9 c

10 b

11 a
—

12 d

13 b

14 **??**

15 a

16 c

50