

Surname:

Name:

1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	
13	

The exam lasts 60 minutes.
In questions 1-13 there is always only one correct option. Write the answers into the table. Only answers in the table will be considered. In case of making corrections, mark clearly which answer is valid. Each correct answer has its value indicated in the question. An answer that is incorrect, ambiguous, or incomplete will be marked with 0 points. The work out is not considered. Only undamaged papers will be accepted. Write the answer to question 14 exclusively on the free area of the paper.

1. (3p) Given is the following code in Java:

```
public class A {
    protected void f() {
    }
}

public class B {
    public static void main(String[] args) {
        A pc = new A();
        pc.f();
        B f = new B();
        f.f();
        A pcf = new B();
        pcf.f();
        B fpc = new A();
        fpc.f();
    }
}
```

Fair will be the calling of f's object method:

- (a) pc
- (b) f
- (c) pcf
- (d) fpc
- (e) none of them

2. (1p) The fact that a subtype without affecting the inherited behavior:

- (a) Is not interesting
- (b) Is required
- (c) Is impossible
- (d) Is possible
- (e) Is appropriate
- (f) Is desired

3. (1p) In created hierarchy of classes we can access hidden attributes and overridden methods by:

- (a) Keyword new
- (b) Keyword final
- (c) Keyword import
- (d) Keyword super
- (e) Keyword extend

4. (2p) For polymorphism except static and final methods is significant:

- (a) Two methods of the same class can have the same name if they differ in the parameter list
- (b) Promotion of primitive types
- (c) In case of overloaded methods the one is selected for which type of the parameters do not fit
- (d) Static attribute initialization takes place when loading classes, otherwise when you create the object
- (f) Selection of the method takes place during the runtime
- (g) The return value cannot be used to distinguish between overloaded methods

5. (2p) Polymorphism means:

- (a) Mechanism which allows different objects to respond the calling different methods in the same way
- (b) Mechanism which allows methods of different objects respond to a calling the same class in a different way
- (c) Mechanism which allows different objects of different types respond to a calling of the same method in a different way
- (d) Mechanism which allows classes of different objects respond to a calling the same method in the same way
- (e) Mechanism which allows slasses of different types respond to a calling the same method in the same way

6. (1p) Access modifier protected

- (a) Makes accessible attributes of a supertype which have to be accessible in its subtype and throughout the hierarchy also in a package
- (b) Makes accessible attributes of a subtype, which have to be accessible in its supertype and throughout the hierarchy also in a package
- (c) Makes accessible attributes of a supertype which have to be accessible in its subtype and throughout the hierarchy except in a package
- (d) Makes accessible attributes of a supertype which have to be accessible in its subtype and throughout the hierarchy also out of a package
- (e) Makes accessible attributes of a supertype which must not be accessible in its subtype and throughout the hierarchy nor in a package

7. (3p) Given is the following code in Java:

```
public interface Skladovatelny {
    void skladuj(Sklad s);
}

public abstract class Potrava implements Skladovatelny{
    void nakrm(Zviera z) {
        System.out.println("Som potrava a krmim
zviera.");
    }
}

public class Kost extends Potrava {
    void nakrm(Zviera z) {
        System.out.println("Som Kost a krmim
zviera.");
    }
}

public abstract class Zviera {
    void zjedz(Potrava p) {
        System.out.println("Zviera");
    }
}

public class Sklad {
    void pridajDoSkladu(Potrava p) {
    }
}
```

The following method

```
public void skladuj(Sklad s) {
    s.pridajDoSkladu(this);
}
```

called skladuj:

- (a) Cannot be implemented in class Kost
- (b) Must be implemented in class Kost
- (c) Must be implemented in class Potrava
- (d) Can be implemented in class Kost
- (e) Cannot be implemented in class Potrava
- (f) None of the above

8. (2p) The key word *this* in command `s.pridajDoSkladu(this);` means:

- (a) Reference to an interface (interface)
- (b) Reference to an instance of an abstract class
- (c) Reference to an implicit constructor
- (d) Reference to an explicit constructor
- (e) Reference to an instance of a method
- (f) Reference to an instance of a class

9. (1p) Which of the following statements about inheritance in Java is wrong:

- (a) It represents the principle of re-using of a code
- (b) It represents an extension of more than one class
- (c) It allows to change something from the supertype which does not fit
- (d) Position in a hierarchy implies a level of abstraction
- (e) It represents concretization/generalization
- (f) It defines the relationship between supertype and subtype

10. (2p) Given is the following code in Java:

```
public class Obyvatel extends Clovek {
    Auto sukromeAuto;
    int vek;
    int silaZraku;
    int rozpocet;

    void zaplatDanZNehnutelnosti(int dan) {
        rozpocet -= dan;
    }
}

public static void main(String[] args) {
    Obyvatel o = new Obyvatel();
    System.out.println(o);
    o.zaplatDanZNehnutelnosti(100);
    Clovek c;
}
```

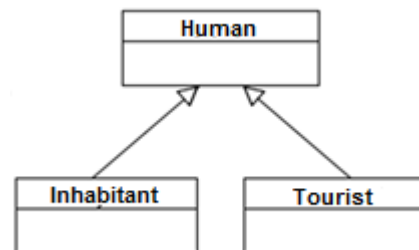
We achieve upcasting if:

- (a) We create an instance of the class Clovek `c=new Clovek();`
- (b) We do not achieve
- (c) Reference c we assign by reference o
- (d) Reference o we assign by instance c
- (e) Reference o we assign by reference c

11. (1p) When there is an implicit constructor in the supertype or there is a constructor without a parameters

- (a) Constructor in the subtype must not be implicit
- (b) Constructor in the subtype must have parameters
- (c) Constructor in the subtype must not have any parameters
- (d) Constructor in the subtype can be implicit
- (e) Constructor in the subtype must have no parameters

12. (2p) Given is the following relationship of classes and the code in Java:



```
Inhabitant i = new Inhabitant();
```

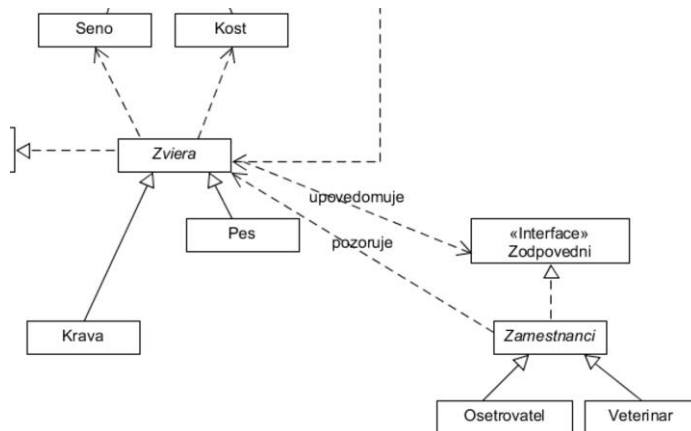
Which of the following statements is **not** correct?

- (a) Object referenced by i can be casted to an object Human
- (b) Object referenced by i cannot be casted to an object Tourist
- (c) Java compiler works with object referenced by i as an instance of the Object class
- (d) Reference i is an instance of the Inhabitant class
- (e) Object referenced by i is a type of Inhabitant; it has all of the methods from Human and Inhabitant class

Surname:

Name:

13. (1p) The relationship *upovedomuje* is realized:



- By calling the method prescribed by the *Zodpovedni* interface with a *Seno* or *Kost* type reference
- By calling the method prescribed by the *Zodpovedni* interface by a reference of type *Krava* or *Pes*
- By calling the method prescribed by the *Zodpovedni* interface by a reference of type *Zodpovedni*
- By calling the method prescribed by the *Zodpovedni* interface by a reference exclusively of type *Osetrovatel* or *Veterinar*
- By calling the method prescribed by the interface *Zodpovedni* by reference exclusively of type *Zamestnanci*
- By calling a method prescribed by the abstract class *Zviera* by a reference of type *Zodpovedni*
- By calling a method prescribed by the abstract class *Zodpovedni* by a reference of type *Zviera*
- None of the above

14. (10p) In our country simulator we have the opportunity to calculate and charge fees for the use of a set of elements of transport infrastructure (toll sections - roads, bridges, tunnels, etc.) and selected category of vehicles (car, bus, tractor, etc.). The fees will vary for different categories of vehicles and time of use of toll sections (weak days, strong days and holidays, etc.). Fees will define in the tariff. The system will allow the identification of the vehicle, calculation and collection of fees without having to stop it. Write the corresponding Java code. Map the real entity of the virtual world and apply appropriate mechanisms of object-oriented programming. Especially use polymorphism. Finally, draw a diagram of the identified classes indicating the relationships between classes. Applied OOP mechanisms comment directly in the code.

Total 25 points

Solution:

	points	
1	3	a
2	1	a
3	1	d
4	2	e
5	2	c
6	1	a
7	3	b
8	2	f
9	1	b
10	2	c
11	1	d
12	2	d
13	4	c