you might have a code that will work even for classes not implemented yet, the only requirement is that they must derive from Person.

Note that in this section we've just introduced the notion of polymorphism. Please, read more about it in Section X.

KEY CONCEPTS YOU NEED TO GRASP FROM THE SECTION

Message *Responsibility* *Agent* *Object*
Access Modifiers *Encapsulation* *Polymorphism*
Inheritance *Object-orientation* *UML* *public* *private*
protected *Imperative programming* *Java*

TESTS

1. Which of these access modifiers are *not* accessible in other packages?

a) private b) public

c) protected d) no access modifier

2. … allows you to derive new classes from existing classes.

a) inheritance b) abstraction

c) encapsulation d) generalization

3. The basic notion of object-oriented programming is…

a) method b) field c) inheritance d) object

4. The term used to describe the internal representation of an object that is hidden from view outside the object's definition?

a) polymorphism b) composition

c) encapsulation d) inheritance

5. When an object has many forms, it is …

a) inherited b) scalable

c) encapsulated d) polymorphic

6. Objects from different classes communicate with each through…

a) inheritance b) polymorphism

c) messages d) responsibilities

7. Which Java keyword is used to specify inheritance?

a) extends b) public c) implements d) inherits

8. Which class is a root of all classes in Java?

a) Interface b) Class

c) Object d) Collection

9. How can you call a class that is derived from another class?

a) superclass b) nested

c) subclass d) cloneable

10. Which of the languages presented below are imperative?

a) Lisp b) C c) Java d) Pascal

PROBLEMS

1. Encapsulation in OOP is used to conceal data from the user and mediate changing of mutable states. Imagine a class *Person* with private field *age*. Think, how to make it impossible to set *age* less than 0 or more than 90.

2. In the section studied we claimed that OOP resembles the way we solve problems in our real life. As a proof we provided a simple illustration for how we solve an everyday problem – ordering a cake for the mom for her birthday with an inscription "Happy Birthday, Mom!". Identify objects (or agents) in this system and their responsibilities.

3. When we were talking about inheritance we demonstrated it on the example of *Mammal* and *Cat* (notice that *Mammal* itself can be a child class of *Animal*). Specify similar hierarchical relationships, but not in the animal world, but in any organization you like (KBTU can be a good choice to consider).