

This document contains information about laboratory exercises.

Object programming - classes 9

Subjects - Polimorphism, Constructors, *final*, interface, JavaDoc

1. Variable, attributes, method access modifiers, polimorphism, abstract class

Encapsulation:

1. Create new project *Zajecia10*.
2. Create new package *paket*.
3. Create in package *paket* class A with *public*, *protected*, *private* attributes and without any access modifier attribute. Add to class A method

```
public void wypisz()
```

that will write on screen *Metoda klasy A*.
4. For given class A create subclass B (B inherits from A). Check to what attributes you have access. Override in class B method

```
public void wypisz()
```

from class A, make `wypisz()`, write on screen *Metoda klasy B*.
5. Create class D in the same package as A and B. Add to class attribute *objectA* of type A. Check to what attributes in object *objectA* you have access.
6. For given class A create in another package subclass C. Check to what attributes you have access.

Polimorphism, abstract class

1. Add to project in default package class *GlownaKlasa* that will contains method *main()*. Define inside three objects:

```
A jeden = new A();  
B dwa = new B();  
A trzy = new B();
```

Write to console value of attribute *a* and invoke method *wypisz()* for each object.
2. Change class A to abstract class (add a word *abstract* before method *wypisz()*), is it sufficient ?

2. Constructors, initialization, final, interface

1. Add to project file *Konstruktory.java*. Run and check how it works.
2. Create second object of type *Pszczola*, what is the order of objects initialization now ?

3. Change constructor of class *Insekt* to parametric constructors with one parameter of type *int*. Correct errors.
4. Add to class *Pszczola* second constructor and invoke inside it the existing one.
5. Add to method *nazwa()* and attributes inside class *Insekt* word *final*. What happened ?
6. What will the effect when we add word *final* before name of class *Insekt*?
7. Change class *Insekt* to (*interface*), correct errors.

3. Do it yourself tasks

1. Write a program that will contains three classes. One abstract class and two others inherits from it. Inside program you have easy access to variable that will count number of object of two non-abstract classes.