

# PACKAGES AND INTERFACES

## Exercise 8

# TASK 1.

Please continue working on the program from task 1 from the previous exercises (including the Person and Staff classes). This time we add an interface called Salary to this program. Our class Staff implements this interface.

In this interface we create three methods:

- "normal" interface method (so without body) with name salaryToPay, that takes two parameters (hours and rate) and compute salary,
- default method, with name salaryForOvertime, that also takes two parameters and compute salary for overtime,
- static method that check if employee should receive bonus (using parameter given yearsOfExperience),

Which method we have to override? Which we can not? And which we can but don't have to?

How to call static method?

What behavior of our default method?

**Important: all methods and variables must have your surname and student ID number.**

# TASK 2.

Please use a program with multi-level class hierarchy that we created on previous labs (those one with Animal abstract class).

This time, we have to add to this program **three** interfaces:

**First: AnimalBehavior** with one „normal” interface method called sleep().

This interface should implement classes in lower place in our hierarchy, so Dog, Pigeon and Blowfish – so in these classes we create some implementation of method sleep(). Then we create an array of AnimalBehavior reference and create objects from our three subclasses. And then we call method sleep() on individual indexes.

**Second: AnimalMove** with one default method called move(), this method should print general info with the name of class it was called and info about Animal move. This interface will be implemented by all classes, but in classes Dog, Pigeon and Blowfish we change default implementation to more specific f.g. on console we should see such message: Dog run or Pigeon fly.

To print messages we create foreach loop on Animal array that we created on previous labs, on call on each element of array method move().

**Third: AnimalName** with one static method called name(), that in parameters receives: String name. This method we call on our interface in foreach loop with parameters getName() method on objects from array Animal (similar like in previous).

In the main class, we create an array of AnimalBehavior references and create all objects from classes that implemented this interface. And then we call selected methods on individual indexes.

# THANK YOU

Więcej na:

[www.vistula.edu.pl](http://www.vistula.edu.pl)



**Akademia Finansów i Biznesu Vistula**  
ul. Stokłosy 3  
02-787 Warszawa  
(obok stacji metro Stokłosy)