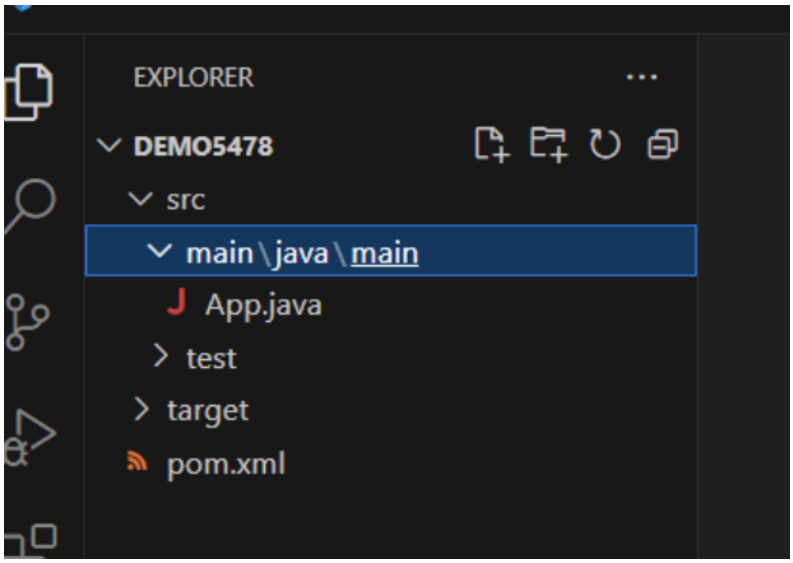
Object-Oriented Programming – Exercise Week 2

Write a program where you create a hedgehog object, and interact with it via command line using a simple menu structure.

Submission

You need a similar project structure as we had in the hello world in the first week. See detailed instructions for creating the Java project there. In short, you must have an App.java file that contains the main-method. In addition you will need a java class file called Hedgehog.java, which defines the hedgehog class.

Submit the program via the submission link to CodeGrade as a zip file:

- **Important! The main-method must be located in a App.java file with specific directory structure, because CodeGrade is very specific about it:**
- 

Hedgehog program

You can score one or points from this exercise, depending on the amount of features you implement.

To interact with this program, you need to implement menu-based controls that take user input from the command line. The idea of menu programs has been already learned in Fundamentals of Programming, so everyone is expected to know the theory of how to read input and create menu-based command line programs.

1 point implementation

Feature 1: Your hedgehog class must have a default constructor, where you can give the name and age as parameters. These parameters are String name and int age. By default, these values should be “Pikseli” and “5”.

Feature 2: The hedgehog should have the ability to speak. This will be handled via the speak() method. The speak method takes a String as an input parameter, and it will print to the console the "<hedgehog name>: <user input>". For example:

Pikseli: Hello World!

Feature 3: Add a possibility to create a new hedgehog with a new name and age. This can be done with the same constructor as in Feature 1, by giving the constructor new name and age parameters. If you enter undefined values as menu options, such as negative numbers, letters or numbers out of range, print out "Wrong input value" to the console.

Task: In the main-method create a default hedgehog instance, with the name Pikseli and age 5, that is usable immediately.

2 point implementation

Feature 4: The hedgehog can also run, so create a run() method for the Hedgehog class. The run method will be given an integer as a parameter (int). The method will print out "<hedgehog name> runs really fast!" as many times as the integer parameter indicated. Ask the user for "How many laps?" and pass the answer integer value to the run method.

Note that <hedgehog name> should be replaced with the actual hedgehog name, for example:

Pikseli runs really fast!

Feature 5: If you enter empty values as input for the hedgehog to say, it will tell it's name and ask for a new input:

"I am <hedgehog name> and my age is <hedgehog age>, but could you still give me input values?".

Example console output and input for the program

```
1) Make hedgehog talk, 2) Create new hedgehog, 3) Make hedgehog run, 0) Quit
1
What does hedgehog say?

I am Pikseli and my age is 5, but could you still give me input values?
1) Make hedgehog talk, 2) Create new hedgehog, 3) Make hedgehog run, 0) Quit
2
What is the name of the hedgehog:
Sonic
What is the age of the hedgehog:
asgoinags
Wrong input value
1) Make hedgehog talk, 2) Create new hedgehog, 3) Make hedgehog run, 0) Quit
2
What is the name of the hedgehog:
Sonic
What is the age of the hedgehog:
33
1) Make hedgehog talk, 2) Create new hedgehog, 3) Make hedgehog run, 0) Quit
3
How many laps?

Wrong input value
1) Make hedgehog talk, 2) Create new hedgehog, 3) Make hedgehog run, 0) Quit
3
How many laps?
4
Sonic runs really fast!
Sonic runs really fast!
Sonic runs really fast!
Sonic runs really fast!
1) Make hedgehog talk, 2) Create new hedgehog, 3) Make hedgehog run, 0) Quit
0
Thank you for using the program.
```