LAB4 - Object Oriented Programming

COMP-2021: Object-Oriented Programming

Objective

- To define and access static members
- To abstract common operations into methods
- To apply information hiding
- To observe the initialization of **static members**, **local variables** and **instance variables**
- To organize Java classes into packages and specify the classpath correctly when using Java tools;

Example

- Refactoring Hero

Class Hero

- Existing members
  - Fields: `name`, `age`, `gender`, `superPower` and `greetingCount`
    - `superPower` is determined by `age` and `gender`
  - Constructors:
    - A three-argument (parameterized) constructor
    - A no-arg constructor
  - Methods:
    - Method `void greeting (Hero another)` accepts another `Hero` and prints a greeting message.

Class Hero

- New Members
  - Add a field `id` so that the i-th created Hero object in the program has i as its `id` .
    - Add the necessary code to initialize the field
    - Make `id` immutable, i.e., once initialized, cannot be changed.
  - Add a method `void greeting (Hero[] heroes)`
    - let `xx` be the number of non-null heroes in `heroes`
    - prints "Hello to all you xx!", if `xx` >1
    - greets the only non-null hero in `heroes` , if `xx` =1

Refactor Class Hero

- Don't repeating yourself (DRY)
  - Abstract / extract common code into a method
- Control the access to class implementation details
  - Read and write private fields via getters and setters
- Avoid magic numbers / literal values.
  - Use constants or enums if necessary

Compare Initialization of Variables

- Local variable vs. instance variable (a.k.a. instance field, non-static field)
- Instance variable vs. class variable (a.k.a. static field)

Organize Java classes into packages

- Create a package `lab.example` in the source folder `src`
  - right click the source folder, select `new` | `Package` and input the package name
- Move `Hero.java` into package `lab.example`
  - Option1: drag and drop the file to the package
  - Option2: right click the source file | `Refactor` | `Move Class ...`
  - Afterwards, the package statement `package lab.example;` should be the first line of `Hero.java`.
- Move `Main.java` into package `lab.example`

Execute Program from Command Line

- Provide search path of java classes via the `-cp` or `--classpath` option of the Java command.
  Command usage:

  `java -cp <directories and zip/jar files> class [args...]`

  - Specify the search path of java classes at `<directories and zip/jar files>`
    - On Windows, separate paths with `;` (e.g., `java -cp c:\jdk\lib;d:\mylib Main`)
    - On MacOs/Linux, separate paths with `:` (e.g., `java -cp /home/jdk/lib:/home/mylib Main`)

- Start Program from Command Line

```
java -cp out/production/lab4/:out/test/lab4/lab.example.Main
```

  - This command adds the project compilation output path to the classpath with `-cp out/production/lab4/:out/test/lab4/`
  - Then start the program from the entry `lab.example.Main`