UNIVERSITÉ DU LUXEMBOURG

Lab 3: Object-oriented modelling-3

Programming Fundamentals 2

16th March 2021

Goals

- ✯ Understanding the inheritance and polymorphism
- ✯ Overview of the encapsulation, overriding and abstraction
- ✯ Reading and writing file and packages

- This laboratory can be done by group of at most 2 students (alone is also fine).
- **Deadline:** 29 March 08:00AM

Deliverables

1. The code on your Github repository generated by clicking here: `https://classroom.github.com/g/QgE3dDnE`
2. A presentation video of 5' minutes on FLIPGRID: `https://flipgrid.com/f29f102f`
   - Use your @student.uni.lu email to connect.
   - Share your screen and comment the Project work.

Exercise 1 – Inheritance

- Lab: See the example code in the **Inheritance-Lab** folder and compile it. Can you able to compile it? what do you understand?
  - Here, you will understand the single, multilevel and hierarchical inheritance basic concepts.
- Exercise: Go to the **Inheritance-Exercise** folder.
  - Go to the **SingleInheritance** folder and separate the class (animals) in a separate `.java` file. For example, Dog and Animal can be defined in the separate `.java` file. For example, `dog.java` file.
  - Likewise, do it for the rest of the inheritance concepts.
  - And add another class in each and every concept (single, multilevel and hierarchical). For example, add another animal, it can be, for example, parrot, pigeon, rabbit, etc,.
  - In the exercise (code), you can see the `STUDENT WORK` where you can complete your work.

Exercise 2 – Overriding

- Lab: See the example code in the **Overriding-Lab** folder and compile it. Are you able to compile it? what do you understand?
- Exercise: Add another class in the separate file and complete the task in the `Main.java` in the **Overriding-Exercise** folder.

Exercise 3 – Polymorphism

- Lab: See the example code in the **Polymorphism-Lab** folder and compile it. Try to understand, dynamic, static, constructor and super.
- Exercise: Go to **Polymorphism-Exercise** folder and add more animal class in **Polymorphism1**, one more shape in **Polymorphism2** and one more arithmetic computation in **Polymorphism3**.

Exercise 4 – Abstraction

- Lab: See the example code in the **Abstraction-Lab** folder and compile it. Are you able to compile it? Try to understand the abstraction class.
- Exercise: Add cube class in the **Abstraction-Exercise** folder files.

Exercise 5 – Encapsulation

- Lab: See the example code in the **Encapsulation-Lab** folder and compile it. Can you able to compile it? Try to understand the private variable.
- Exercise: Create one more inherit class (it can be another person), see the code skeleton in the folder **Encapsulation-Exercise** folder.

[Project-Compulsory]: You can have you own idea, or you can refer to `https://projectgurukul.org/java-project-ideas/` for some project ideas. The project work should address, four main fundamental concepts of OOP, that is polymorphism, inheritance, encapsulation, and abstraction.

Project option 1
Online Banking at Luxembourg, maintaining a joint account of peoples:
Welcome terminal: (for bank employee)
User Name:
Login:

People:
Man: add or save
Woman: add or save
Kid: add or save

Account detail: First Name (man)
Personal account: xyxyx dollars

Saving account: xyxyx dollars

Account detail: First Name (Woman)
Personal account: xyxyx dollars
Saving account: xyxyx dollars

Account detail: First Name (Kid)
Personal account: xyxyx dollars
Saving account: xyxyx dollars

Options:
1) Transfer (within the family, for example, within the saving to personal)
    1) man
    2) woman
    3) kid
2) Deposit
    1) man
    2) woman
    3) kid
3) Width draw
    1) man
    2) woman
    3) kid
4) Account transaction history:
    1) Man
    2) Woman
    3) Kid
5) Quit

Project option 2
An example for a hospital management project:

Hospital management:

Overview: Date, day and time

Manpower:
Doctors: how many of them are at duty today
Nurses: how many of them are at duty today
Receptionist: how many of them are at duty today
Admin: how many of them are at duty today

Wards:
Emergency Ward: Available and occupied
Normal Ward: Available and occupied

Medicine:
Medicine Normal: available and not available
Emergency Medicine: available and not available

Patience:
Patience: Presently, how many (counting emergency ward, normal ward)

Options:

1) Patience:
    1) Normal: add or remove
    2) Emergency: add or remove
2) Manpower: add or remove
    1) Doctors: add or remove
    2) Nurses: add or remove
    3) Receptionist: add or remove
    4) Admin: add or remove
3) Medicine:
    1) Normal medicine: add or remove
    2) Emergency medicine: add or remove
4) Show statistics of the entire system
5) Quit

Project option 3
In this project, where you can maintain the activities in the computer system:

Hotel management: Manpower:
Admin: how many (show maximum of 5 person)
Cleaning: how many (show maximum of 5 person)

Available rooms:
Single: how many (maybe 5)
Double: how many (maybe 5)
Large: how many (maybe 5)

Food Menu:
Breakfast: raw material (in stock)
Lunch: raw material (in stock)
Dinner: raw material (in stock)

Beverages:
Water: in stock
Juice: in stock
Tea: in stock
Coffee: in stock

Options:
1) Manpower:
    1) Admin: remove or add
    2) Cleaning: remove or add
2) Rooms:
    1) Single: add or vacate
    2) Double: add or vacate
    3) Large: add or vacate
3) Food menu:
    1) Breakfast: add or remove
    2) Lunch: add or remove
    3) Dinner : add or remove
4) Beverages:
    1) Water: add or remove
    2) Juice: add or remove
    3) Tea: add or remove
    4) Coffee: add or remove
5) Show all statistics (present)

6) Quit

[Plus Ultra] You can create a GUI application for anyone of the project ideas which is specified above. For example, the following list also shows a few of the specific examples.

- Billing system
- Management system
- Reservation system

Good luck!