CS 131 - Week 4

TA : Tanmay Sardesai

How to find these slides

Piazza -> CS 131 -> Resources -> Discussion 1B

Announcements

- Email tanmays@cs.ucla.edu
- Office Hours Tuesday 9:00 am - 11:00 am. Bolter Hall 3256S-A
- HW3 will be due Monday 2/11 11:55 pm
- Midterm is on 02/06

Topics covered today

- Java introduction
- Object Oriented introduction and concepts
- Sample Midterm

Java Intro

- We will be using Java 11 for in this class.
- Recommended that you use an IDE like Eclipse, Netbeans, IntelliJ.
  - This will give you autocomplete, debugging, syntax highlighting and other features to make your life easier
  - Other option is to use a text editor + terminal
- To run a program on terminal run javac first to compile the files and then java to execute.
  - Ex: for file main.java. Use **javac main.java** and then **java Main**

Java intro

- Simple
- Object oriented
- Distributed
- Multithreaded
- Dynamic

- Architecture neutral
- Portable
- High performance
- Robust
- Secure

https://docs.oracle.com/javase/tutorial/getStarted/intro/definition.html

Java intro

- In the Java programming language, all source code is first written in plain text files ending with the `.java` extension.
- Those source files are then compiled into `.class` files by the `javac` compiler. A `.class` file does not contain code that is native to your processor; it instead contains *bytecodes* — the machine language of the Java Virtual Machine (Java VM).
- The `java` launcher tool then runs your application with an instance of the Java Virtual Machine.

Pure Object Oriented Languages

Five rules (Source: Alan Kay)

- Everything is an object
- A program is a set of objects telling each other what to do by sending messages
- Each object has its own memory (made up by other objects)
- Every object has a type
- All objects of a specific type can receive the same messages

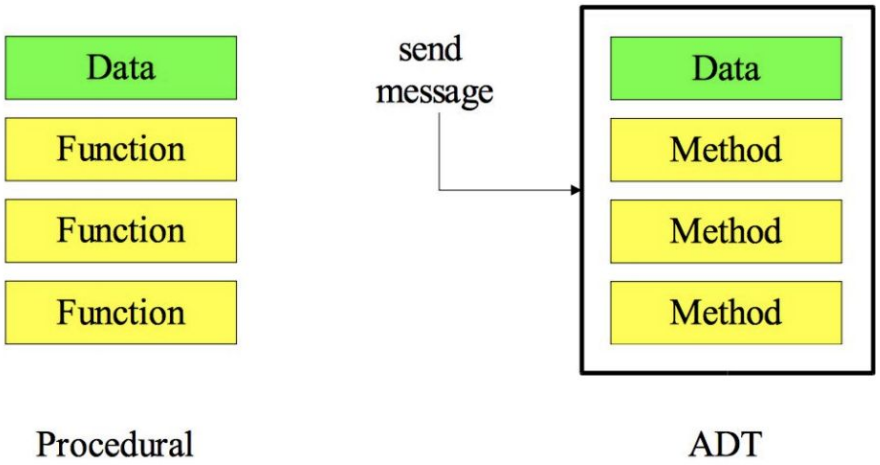
Java breaks some of these rules

What is an Object

- An object is an encapsulation of data.
- An object has:
  - State
  - Behaviours
- An object is an instance of an abstract data type
- An abstract data type is implemented as a class

Encapsulation

- Hide data from the outside world. Data can only be accessed by methods of the object



Class vs Object

Class

- A description of a common set of objects. Think of this as a blueprint
- Think, examples:
  - Person
  - BoardGame
  - Animal

Object

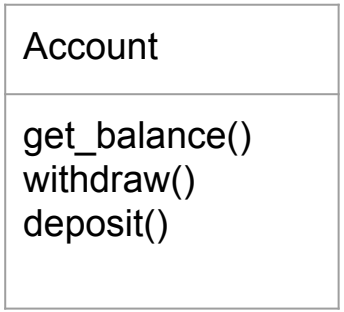
- An instance of a class. It is a representation of a single instance.
- Examples:
  - Tanmay Sardesai, Paul Eggert
  - Monopoly, Risk
  - Dog, Cat

Type and Interface

An object has a type and an interface

Type

Interface



To create an object just type `Account my_account = new Account()`

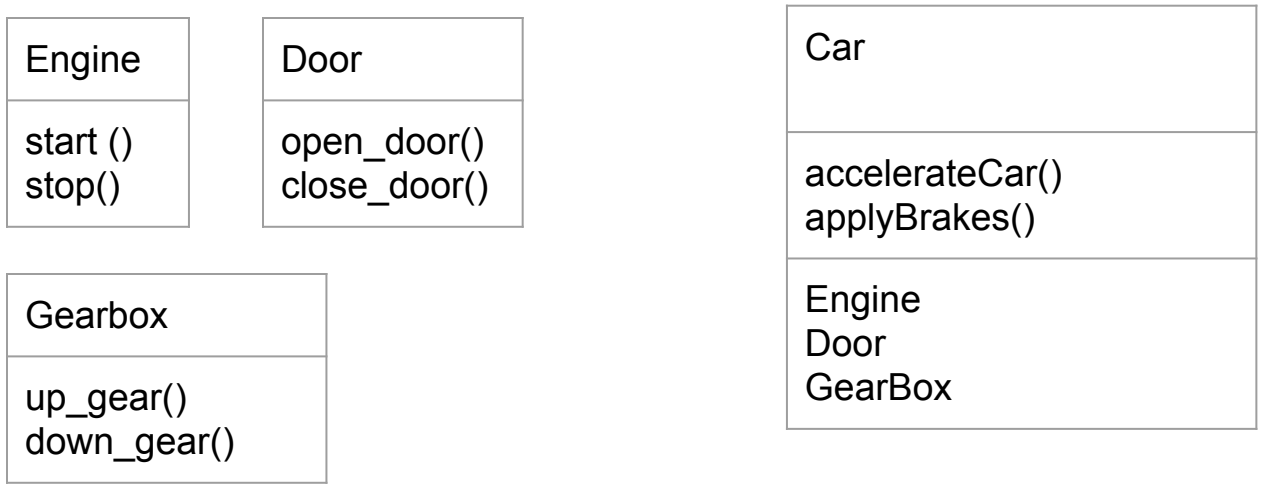
To send some message use an interface. `my_account.withdraw()`

Interactions between objects

- An object 1 interacts with another object 2 by calling a method on O2, must be part of the interface.
- The call of a method corresponds to a function (or procedure) call in a non-object-oriented language such as C.
    - <object-variable>.<method-call>

Aggregation and Decomposition

- Make new classes by reusing existing. Reuse reduces effort!



Next week

- Discuss the homework
- Inheritance
- Interfaces
- Multi-threading
- ...

Sample midterm

Questions

References

https://docs.oracle.com/javase/tutorial/tutorialLearningPaths.html

https://docs.oracle.com/javase/tutorial/getStarted/index.html

https://docs.oracle.com/javase/tutorial/java/index.html

https://docs.oracle.com/javase/tutorial/essential/index.html