Object Oriented Programming

Kookmin University

Department of Computer Science

Announcements

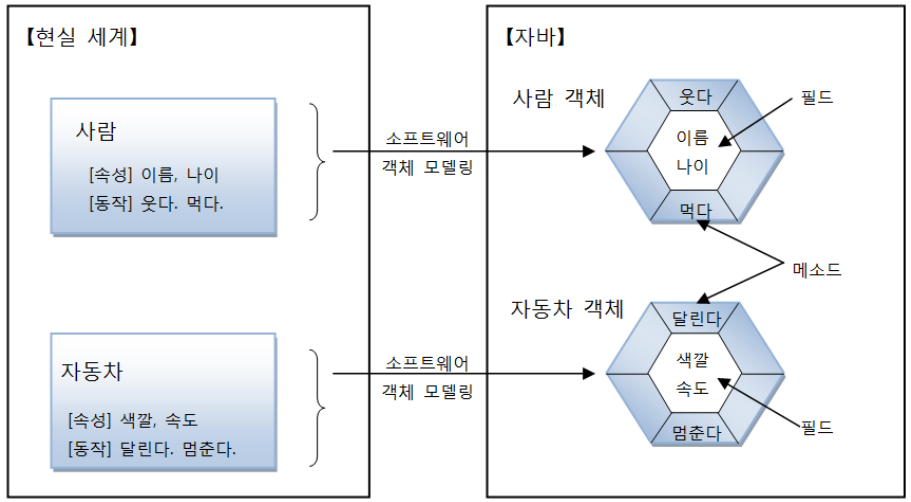
- Midterm exam
  - Oct. 18th (**Wednesday**)
  - Location: 7호관 629호
  - Until multi-dimensional array
- Homework 3 explanation

Object Oriented Programming

- Comparing to a process of making a car
  - Makes components first – engine, tire, door, …
  - Later, assemble it
  - Some cars shares components – economic and luxurious version
- What is **Object**
  - Things that physically exist – car, book, person
  - Abstract stuffs that have own attributes
    - KMU CS department - # of students, classes, faculty, …
    - KMU CS department class can be another object – title, timetable, syllabus, …
- Object Oriented Programming models programming using objects

Object in Java

- Java allows express an object using
  - Field – attributes that express characteristics of an object
    - Person's name, Class hour
  - Method – an action that an object can take
    - Person: walk, class: taking an exam



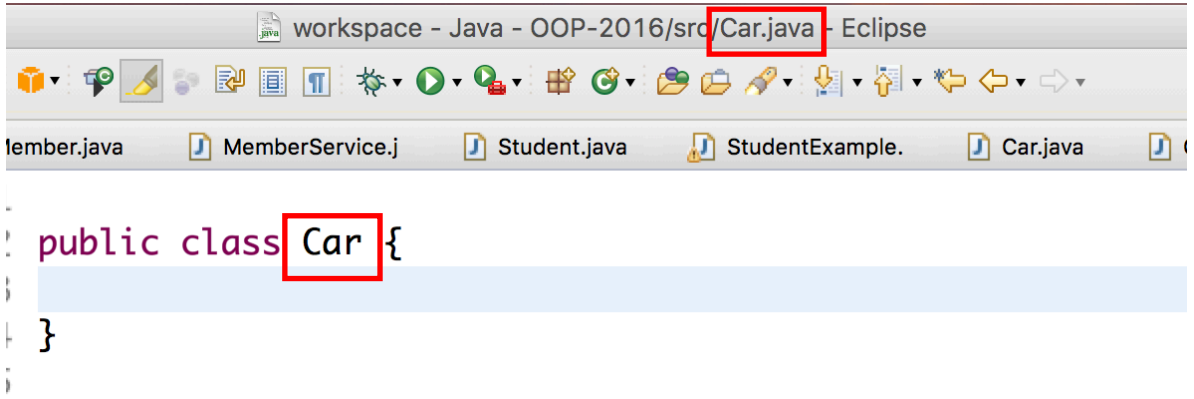
이것이자바다 P. 186

Object and Class

- In Java, an object is created from a class
- A class defines how an object will be shaped
  - Field – defines attributes
  - Method – defines functions
- Instance
  - An object created from a class
- Multiple instances can be created from a class
  - Student class can general many instances per each students

Implementing Class

- Decide class name
  - Encouraged to use English name
    - No number as the first character of class name
    - No special character except $ and _
    - No java keyword (int, for, while, …)
  - Conventional: the first character of a word is capital (WebBrowser)
- Class definition and compile
  - One class in one file
  - Access modifier class class_name
    - public class Car {…}
  - Code file name: class_name.java
  - Car.java → public class Car {…}



Creating Object from Class

- If an object is created from a class definition, it is stored in the heap region, and the stack region stores the address to the instance (reference type variable)
- new keyword is used to create an instance
  - Car myCar = new Car();

Class Composition

- A class is composed of
  - Constructor – called once in the initialization (setting age to 0)
  - Field – stores data of a class (name, kmu-id, …)
  - Method – defines an action for a class (eat, walk, …)

```
public class Car {
    Car() {

    }

    String modelName;

    void drive() {

    }
}
```

Class Field

- A field in a class stores data or the status of an object
- EX: Car object
  - General data – manufactured company, model, color
  - Status – current speed, gas level
  - Components – Car body, engine, tire
- Field needs to defined within {} of a class – not in method and constructor scope
- Declaration format is very similar to variable declaration
  - Syntax: type name = initialization
  - EX: String company = “Hyundai Motor”;

Accessing Object Field

- An object field can be accessed only when an object is created from a class definition
- The object field is accessed by using a variable name followed by dot and field name
  - Car myCar = new Car();
  - myCar.name

Class Method

- Actions that can be executed using class fields
- Method consists of
  - Return type
    - Decides a type of outcome from execution of a method (code body)
  - Method name
    - A name of method where a programmer can reference the method
  - Arguments
    - Passes data to be executed in a method code
  - Code body
    - A set of codes that is executed when a method is called
- Method signature
  - Return type + Method name + Arguments

Class Method Example

- Return type
  - int (the output is an age of a car)
- Method name
  - GetAge : calculate age of a car
- Argument
  - int currentYear: Using current year, we can calculate the age of a car
- Code body
  - Calculate the age of a car

```
int modelYear = 2006;

int GetAge(int currentYear) {
    int age = currentYear - modelYear;
    return age;
}
```

Method Name

- Method name should be reasonable to understand what the method does
- It should not start with number
- No special characters except $ and _
- By convention
  - The method name should be lowercase letters
  - For consecutive words, use capital for a new word (Camel case)

Method Arguments

- Arguments allow to pass data from the method caller to the method itself
- Arguments can be omitted (input data is not necessary)
- Each argument should contain its type and name that can be accessed within a method body
- Multiple arguments are separated by commas
- Array can be passed as an argument
  - If the length of an array is not known at the time of declaration, we can use "…" : int sum(int … inputs)
- Calculator exercise

Calling Method

- Within Object
  - Use the method name directly with input arguments
- Outside of an Object
  - Create an object instance and access the method using "." (dot) separator
    - calculator.average();

```
Calculator calculator = new Calculator();
int[] numbers = { 10, 9, 8, 7, 6 };
System.out.println(calculator.average(numbers));
```

```
double divide(int x, int y) {
    double result = (double) x / (double) y;
    return result;
}

int sumArray(int... inputs) {
    int sum = 0;
    for (int i : inputs) {
        sum += i;
    }
    return sum;
}

double average(int... inputs) {
    int sum = sumArray(inputs);
    return divide(sum, inputs.length);
}
```

References

- 이것이자바다 – 한빛미디어 2015