Static Variable in Character Class

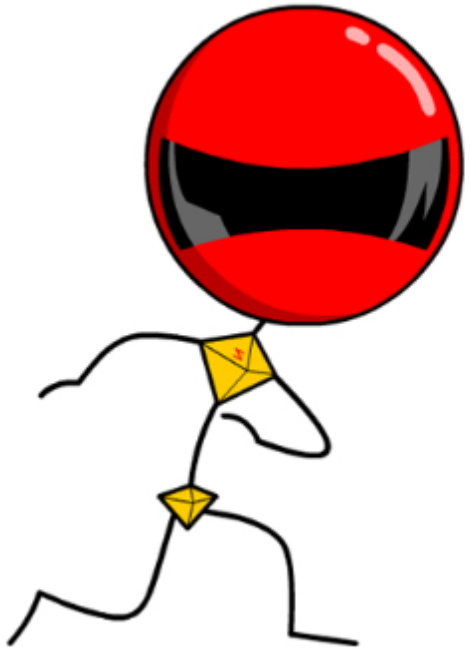
Sungchul Lee

Learning Object

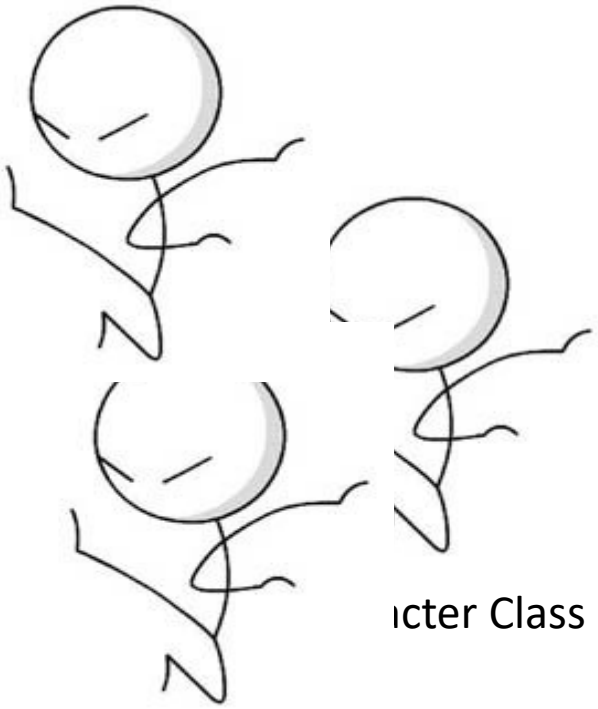
static String server_name

Main Character

Enemy



V.S.



Character Class

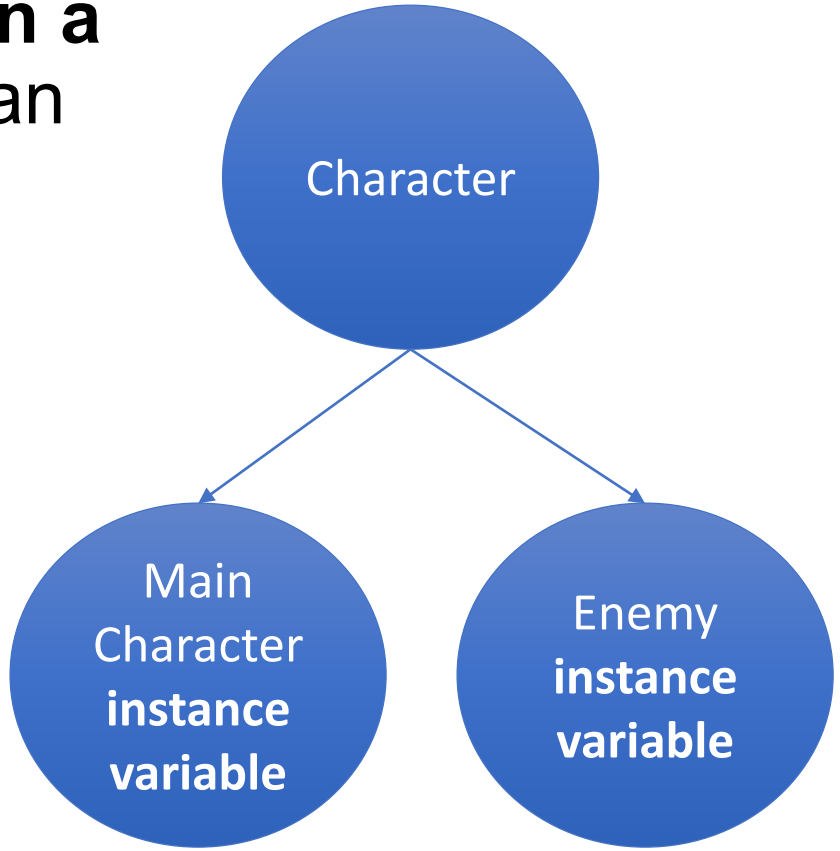
Array of Object

ıcter Class

Instance

- A variable of method that is **dependent on a specific instance** of the class should be an **instance** variable or method.
  - Each variable has own memory area

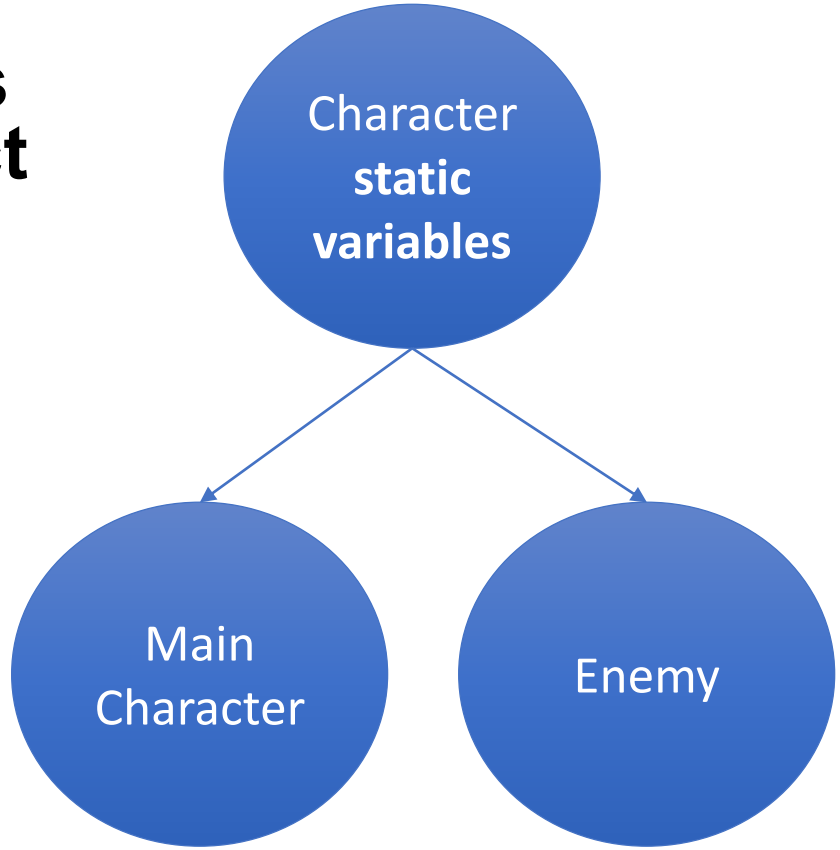
- Instance variable in Character Class
  - Hp,mp, power, defense, name



Static

- **A static variable belongs to a class as a whole and not to an individual object**
  - **Static variables** store values for the variables in a common memory location
  - Reduce usage memory and code

- Static variables in Character class
  - Game Server name
    - All character in same



Practice

1. Open project (Starting from preivous)
   - ❑Project name: Game
2. Create a new Class File
   - ❑Class name: Main
   - ❑Class name: Character
3. Coding:
   - ❑import java.util.Random; // for Character class
   - ❑import java.util.Scanner; // for Main class

Practice – code (Character)

Character.java

```java
import java.util.Random;
public class Character {
    private static String server_name = "CS172";
    private int hp;
    private int power;
    private String name;
    private int defense;
    private int mp;
    Random rnd =new Random();

    public Character(String name){
        this.hp = rnd.nextInt(100);
        this.power = rnd.nextInt(100);
        this.name = name;
        this.defense = rnd.nextInt(10);
        this.mp=rnd.nextInt(100);
    }
    public Character(int hp, int power,String name){
        this.hp = hp;
        this.power = power;
        this.name=name;
        this.defense = rnd.nextInt(10);
        this.mp=rnd.nextInt(100);

    }
```

Add static variable

Instance Variable and Constructor

```java
    public static String getServer_name() {
        return server_name;
    }
    public static void setServer_name(String server_name) {
        Character.server_name = server_name;
    }
    public int getHp() {
        return hp;
    }
    public void setHp(int hp) {
        this.hp = hp;
    }
    public int getPower() {
        return power;
    }
    public void setPower(int power) {
        this.power = power;
    }
    public String getName() {
        return name;
    }
    //no setter for name
    public int getDefense() {
        return defense;
    }
    public void setDefense(int defense) {
        this.defense = defense;
    }
    public int getMp() {
        return mp;
    }
    public void setMp(int mp) {
        this.mp = mp;
    }
```

Add Getter and Setter

Damage method

```java
    public int damage(int enemy_power){
        int damage = enemy_power-this.defense;
        if(damage<0){ // avoid healing by damage
            damage =0;
        }
        this.hp=this.hp - damage;
        if(this.hp<0){ // avoid minus hp
            this.hp =0;
        }
        return damage;
    }
}//End Class
```

Practice – code (Main)

Main.java

```
1  import java.util.Scanner;
2  public class Main {
3      public static void main(String[] args) {
4          // TODO Auto-generated method stub
5          Scanner scanner = new Scanner(System.in);
6          System.out.println("Input Main Character Name:");
7          String main_name=scanner.nextLine();
8          System.out.println("Input Main Character Power:");
9          int main_power=scanner.nextInt();
10         System.out.println("Input Main Character HP:");
11         int main_hp=scanner.nextInt();
12
13         //Generate Main Character Object (Declare and initialize)
14         Character main_ch = new Character(main_hp,main_power,main_name);
15         show_status(main_ch);
16
17         Character enemies[] = new Character[5];//Generate objects
18         for(int i =0 ; i<enemies.length;i++){
19             enemies[i]=new Character("enemy " + i); //initialize
20             show_status(enemies[i]);
21         }
```

Generate Characters

Practice – code (Main) – cont.

Give damage using damage method in Character class

Change server name

```
23          show_status(enemies[0]);
24          enemies[0].damage(main_ch.getPower());
25          show_status(enemies[0]);

            System.out.println("Character Server:");
            System.out.println("Main:"+ main_ch.getServer_name());
29          System.out.println("Enemies:"+enemies[0].getServer_name());
30
31          main_ch.setServer_name("New Server");
            System.out.println("After changing Character Server:");
            System.out.println("Main:"+ main_ch.getServer_name());
34          System.out.println("Enemies:"+enemies[0].getServer_name());
35      }// end main method
```

Practice – code (Main)

Static method for checking character status

```
37      public static void show_status(Character character){
38          System.out.println("==============Character Status================");
39          System.out.println("Character Name:"+character.getName());
40          System.out.println("Character hp:"+character.getHp());
41          System.out.println("Character Name:"+character.getPower());
42          System.out.println("Character Name:"+character.getDefense());
43          System.out.println("Character hp:"+character.getMp());
44      }
45  } //End Class
```

Practice –Result

- Damage Methods in Character Class
  - Give damage
- Static Server name
  - All character's server name changed
  - Common memory for Character Class

Result

```
==============Character Status===============
Character Name:enemy 0
Character hp:43
Character Name:85
Character Name:3
Character hp:61
==============Character Status===============
Character Name:enemy 0
Character hp:0
Character Name:85
Character Name:3
Character hp:61
Character Server:
Main:CS172
Enemies:CS172
After changing Character Server:
Main:New Server
Enemies:New Server
```

Before damaging

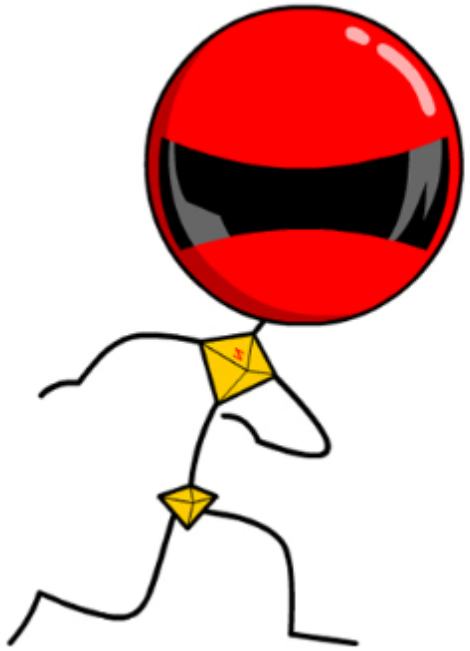
After damaging

Summary

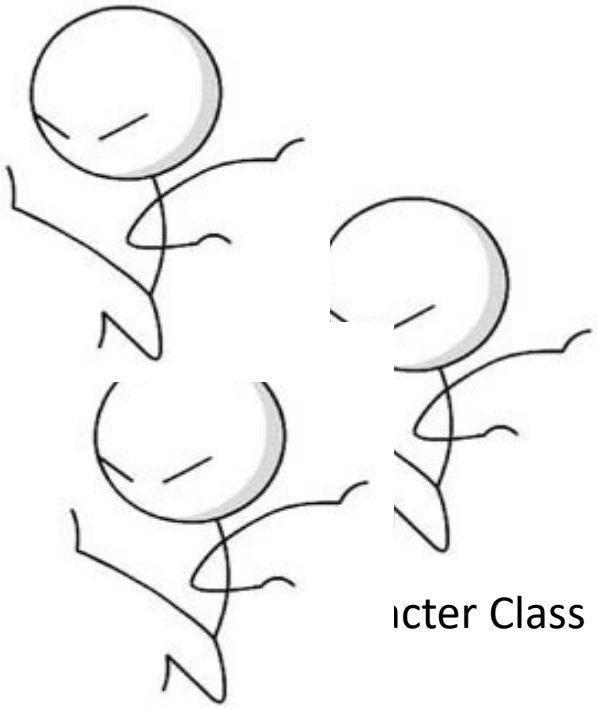
static String server_name

Main Character

Enemy



V.S.



ıcter Class

Character Class

Array of Object