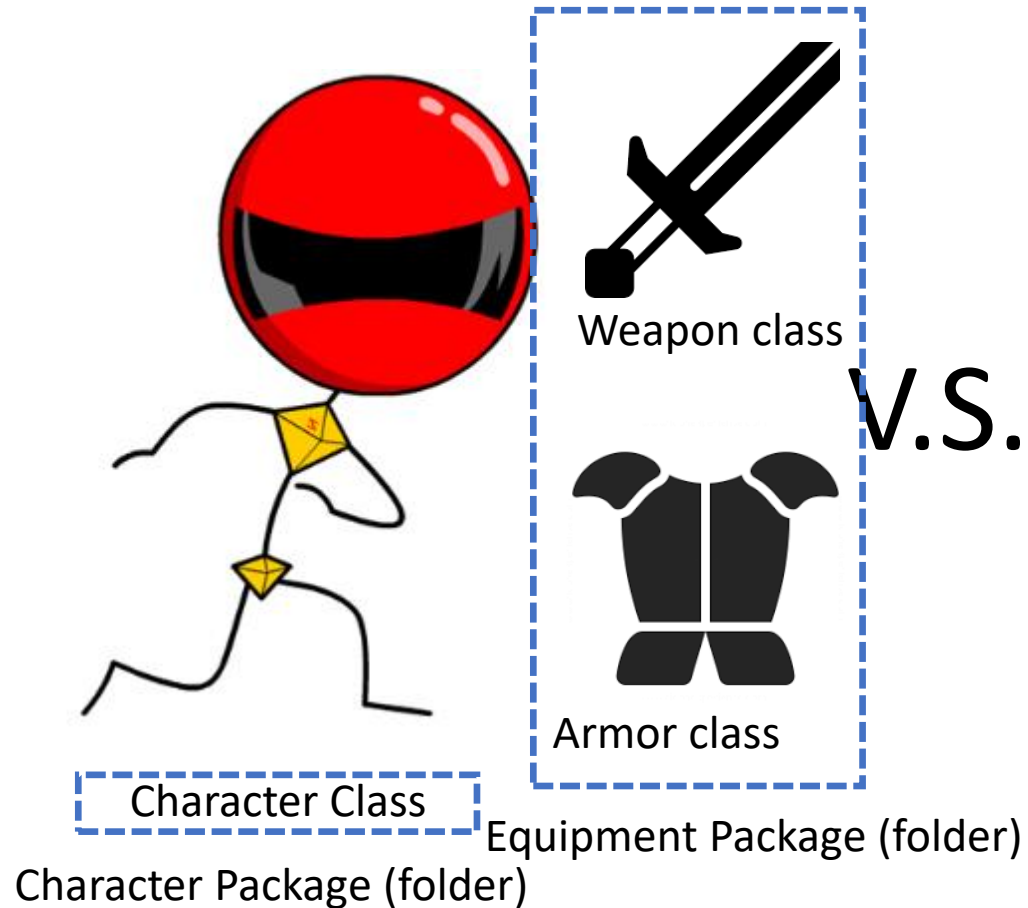


Packages and Weapon/Armor Class

Sungchul Lee

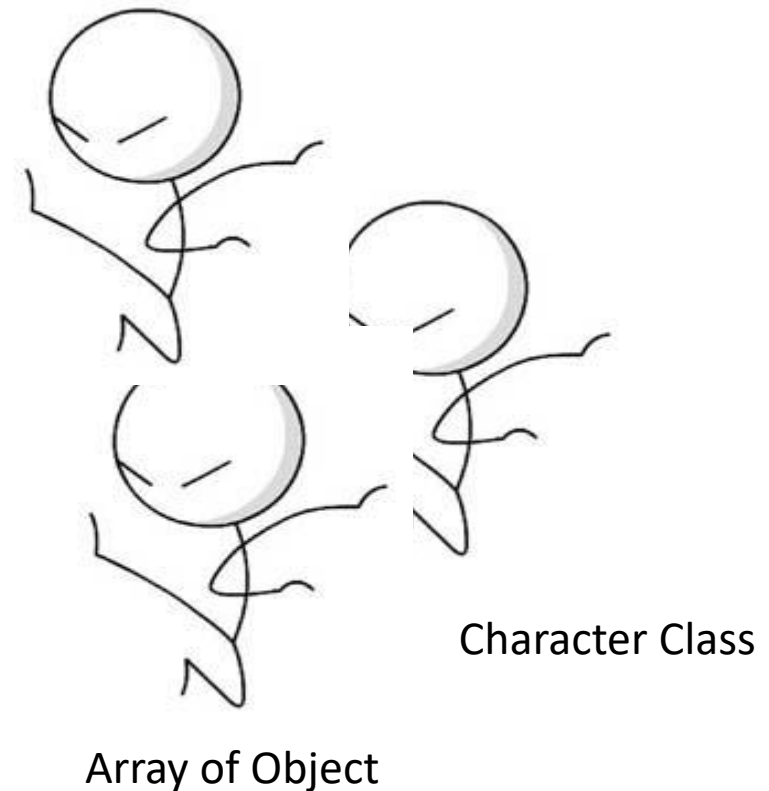
Learning Object

Main Character



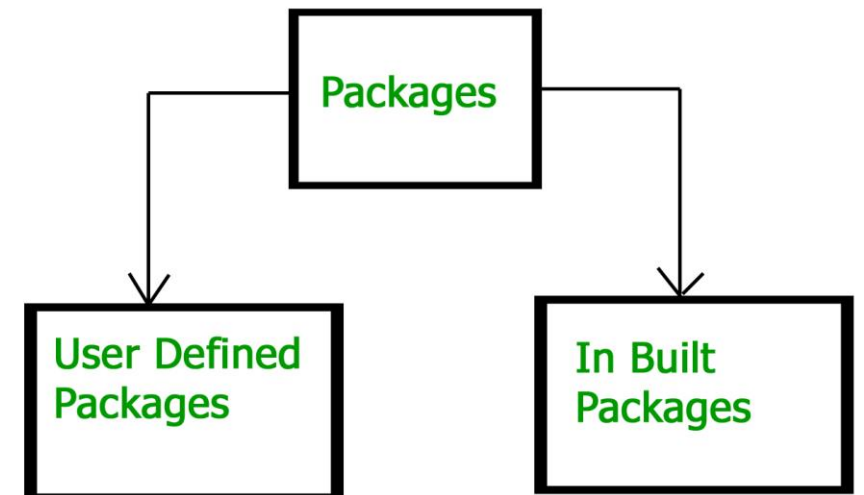
V.S.

Enemy



Packages

- Easy searching/locating and usage of classes
- Built-in Packages (Java API)
 - ❑ java.lang
 - ❖ primitive data types, math operations
 - ❑ java.io
 - ❖ input/output operations
 - ❑ java.util
 - ❖ Date/time
 - ❖ Random
- User-defined packages
 - ❑ e.g.) Character and Equipment



Weapon Class (Equipment Package)

- Increase Character's power
 - ❑ `Character.power + Weapon.power`
- ***Instance Variable***
 - ❑ Name, power
- ***Methods***
 - ❑ Constructor for initializing Weapon
 - ❑ Getter and setter for instance variable
- Built-in Package
 - ❑ Random

Armor Class (Equipment Package)

- Increase Character's defense
 - ❑ Character. defense + Armor. defense
- ***Instance Variable***
 - ❑ Name, defense
- ***Methods***
 - ❑ Constructor for initializing Armor
 - ❑ Getter and setter for instance variable
- Built-in Package
 - ❑ Random

Character Class (Character Package)

- New method to get charter's hit point
 - ❑ Need new hit point for weapon's power
 - ❑ `int hit_point = Character.power + Weapon.power`
 - ❑ Return the hit_point for damage (return type int)
- Modify damage method
 - ❑ Reducing damage by armor's defense
 - ❑ `int damage = hit_point – Charaster.defense – Armor.defense`
 - ❑ Return the damage (Return type int)

상속(Inheritance)

- 상속은 기존 클래스의 속성과 메소드를 새로운 클래스가 물려받아 사용할 수 있게 하는 메커니즘
 - 코드 재사용성을 높이고 클래스 간의 계층적 관계를 구성
 - 부모 클래스(Superclass): 상속을 해주는 기존 클래스
 - 자식 클래스(Subclass): 상속을 받는 새로운 클래스
 - extends 키워드: 상속 관계를 설정하는데 사용
- 자바에서는 다음과 같은 상속 유형을 지원함
 - 단일 상속: 하나의 자식 클래스가 하나의 부모 클래스를 상속
 - 다중 레벨 상속: 상속이 여러 단계로 이어지는 구조
 - 계층적 상속: 여러 자식 클래스가 하나의 부모 클래스를 상속

Abstract (Equipment.java)

➤ Superclass for Weapon and Armor

➤ Abstract

❑ abstract keyword

❑ 직접적인 객체 생성 불가능

❑ 반드시 하위 클래스에서 상속받아 사용

❑ 하위 클래스는 모든 추상 메소드를 구현

추상화의 장점

1. 템플릿 제공

- 공통 기능을 추상 클래스에 구현
 - 하위 클래스에서 재사용 가능한 템플릿 역할

2. 코드 재사용성

- 중복 코드 감소
- 유지보수 용이성 향상

```
package Equipment;

public abstract class Equipment {
    protected String name;

    public Equipment(String name) {
        this.name = name;
    }

    public String getName() {
        return name;
    }

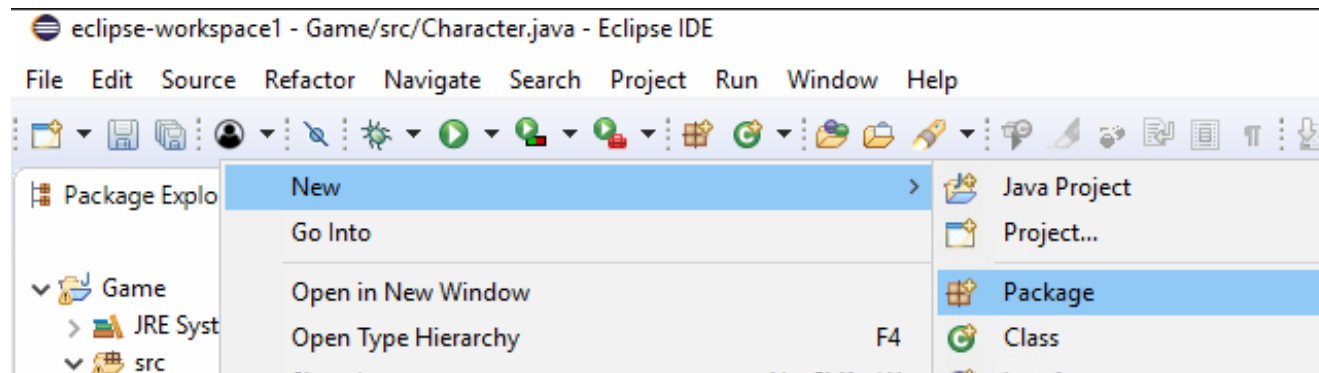
    // 추상 메소드 선언
    public abstract int getItemValue();
}
```


Practice

1. Open project (Starting from previous)
 - ❑ Project name: Game
2. Create new two packages
 - ❑ Character and Equipment
3. Create two new Class Files in Equipment package
 - ❑ Class name: Weapon
 - ❑ Class name: Armor
4. Coding:
 - ❑ `import java.util.Random;` // for Character, Weapon and Armor
 - ❑ `import java.util.Scanner;` // for Main class

How to make User Define Package

- Click mouse' right button on Project



- Make Character package and Equipment package
 - ❑ Move character class to Character Package
 - ❖ Drag character file to Character Package
 - ❖ Remove Character class in default package (if two character class in the project)

```
// Equipment.java
package Equipment;

public abstract class Equipment {
    protected String name;

    public Equipment(String name) {
        this.name = name;
    }

    public String getName() {
        return name;
    }

    // 추상 메소드 선언
    public abstract int getItemValue();
}
```

```
// Weapon.java
package Equipment;
import java.util.Random;

public class Weapon extends Equipment {
    private int power;
    private Random rnd = new Random();

    public Weapon() {
        super("Weapon" + new Random().nextInt(100));
        this.power = rnd.nextInt(100) + 1;
    }

    public Weapon(String name) {
        super(name);
        this.power = rnd.nextInt(100) + 1;
    }

    public int getPower() {
        return power;
    }

    public void setPower(int power) {
        this.power = power;
    }

    @Override
    public int getItemValue() {
        return power * 10;
    }
}
```

Super
class 호출

abstract

```
// Armor.java
package Equipment;
import java.util.Random;

public class Armor extends Equipment {
    private int defense;
    private Random rnd = new Random();

    public Armor() {
        super("Armor" + new Random().nextInt(100));
        this.defense = rnd.nextInt(10) + 1;
    }

    public Armor(String name) {
        super(name);
        this.defense = rnd.nextInt(10) + 1;
    }

    public int getDefense() {
        return defense;
    }

    public void setDefense(int defense) {
        this.defense = defense;
    }

    @Override
    public int getItemValue() {
        return defense * 20;
    }
}
```

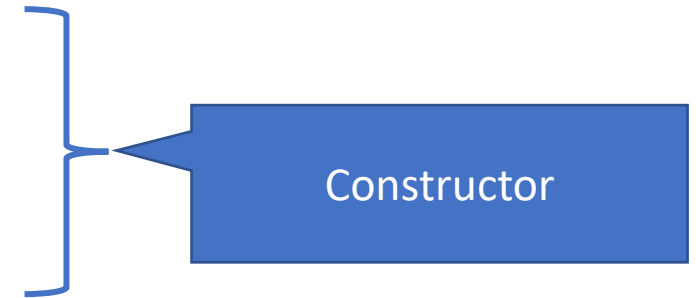
Practice – code (Weapon)

```
Weapon.java
1 package Equipment;
2 import java.util.Random;
3
4 public class Weapon {
5     private String name;
6     private int power;
7     Random rnd = new Random();
8     public Weapon() {
9         this.name="Weapon"+ rnd.nextInt(100);
10        this.power=rnd.nextInt(100)+1;
11    }
12    public Weapon(String name){
13        this.name=name;
14        this.power=rnd.nextInt(100)+1;
15    }
16    public String getName() {
17        return name;
18    }
19    public int getPower() {
20        return power;
21    }
22    public void setPower(int power) {
23        this.power = power;
24    }
25 }
```

Constructor

Getter and Setter

Practice – code (Armor)



Practice – code (Character)

Character.java

```
1 package Character;
2 import java.util.Random;
3 import Equipment.*;
4 public class Character {
5     private static String server_name = "CS172";
6     private Armor armor = new Armor();
7     private Weapon weapon = new Weapon();
8     private int hp;
9     private int power;
10    private String name;
11    private int defense;
12    private int mp;
13    Random rnd = new Random();
14
15    public Character(String name){
16        this.hp = rnd.nextInt(100);
17        this.power = rnd.nextInt(100);
18        this.name = name;
19        this.defense = rnd.nextInt(10);
20        this.mp = rnd.nextInt(100);
21    }
22    public Character(int hp, int power, String name){
23        this.hp = hp;
24        this.power = power;
25        this.name = name;
26        this.defense = rnd.nextInt(10);
27        this.mp = rnd.nextInt(100);
28    }
```

Import equipment package

Add armor and weapon object

```
30    public Armor getArmor() {
31        return armor;
32    }
33    public void setArmor(Armor armor) {
34        this.armor = armor;
35    }
36    public Weapon getWeapon() {
37        return weapon;
38    }
39    public void setWeapon(Weapon weapon) {
40        this.weapon = weapon;
41    }
42    public static String getServer_name() {
43        return server_name;
44    }
45    public static void setServer_name(String server_name) {
46        Character.server_name = server_name;
47    }
48    public int getHp() {
49        return hp;
50    }
51    public void setHp(int hp) {
52        this.hp = hp;
53    }
54    ... ..
```

Add Getter and Setter

Practice – code (Character) – cont.

```
64*   public int getDefense() {
65       return defense;
66   }
67*   public void setDefense(int defense) {
68       this.defense = defense;
69   }
70*   public int getMp() {
71       return mp;
72   }
73*   public void setMp(int mp) {
74       this.mp = mp;
75   }
76*   public int hit_point(){
77       int hit_point = this.power+this.weapon.getPower();
78       return hit_point;
79   }
80
81*   public int damage(int enemy_hit_point){
82       int damage = enemy_hit_point - this.defense - this.armor.getDefense();
83       if(damage<0){ // avoid healing by damage
84           damage = 0;
85       }
86       this.hp=this.hp - damage;
87       if(this.hp<0){ // avoid minus hp
88           this.hp = 0;
89       }
90       return damage;
91   }
92 } //End Class
```

Add hit_point method

Modify damage

Practice – code (Main)

Import Character Package

```
*Main.java
1 import java.util.Scanner;
2 import Character.Character;
3 public class Main {
4     public static void main(String[] args) {
5         // TODO Auto-generated method stub
6         Scanner scanner = new Scanner(System.in);
7         System.out.println("Input Main Character Name:");
8         String main_name=scanner.nextLine();
9         System.out.println("Input Main Character Power:");
10        int main_power=scanner.nextInt();
11        System.out.println("Input Main Character HP:");
12        int main_hp=scanner.nextInt();
13
14        //Generate Main Character Object (Declare and initialize)
15        Character main_ch = new Character(main_hp,main_power,main_name);
16        show_status(main_ch);
17
18        Character enemies[] = new Character[5]; //Generate objects
19        for(int i =0 ; i<enemies.length;i++){
20            enemies[i]=new Character("enemy " + i); //initialize
21            show_status(enemies[i]);
22        }
23        show_status(enemies[0]);
24        enemies[0].damage(main_ch.hit_point()); //give damage to enemy
25        show_status(enemies[0]);
26
27        System.out.println("Main Character Equipment:");
28        System.out.println("Name:"+main_ch.getWeapon().getName());
29        System.out.println("Power:"+main_ch.getWeapon().getPower());
30        System.out.println("Name:"+main_ch.getArmor().getName());
31        System.out.println("Power:"+main_ch.getArmor().getDefense());
```

Give damage using
damage method in
Character class

Get Weapon
and Armor info

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

➤ Hit_point method

- ❑ Get hit point of character

➤ Weapon and Armor Class

- ❑ Increase power

- ❑ Increase defense

Result



```
=====Character Status=====
Character Name:enemy 0
Character hp:90
Character Name:30
Character Name:4
Character hp:66
=====Character Status=====
Character Name:enemy 0
Character hp:0
Character Name:30
Character Name:4
Character hp:66
Input Main Character Equipment:
Name:Weapon96
Power:90
Name:Armor70
Power:9
```

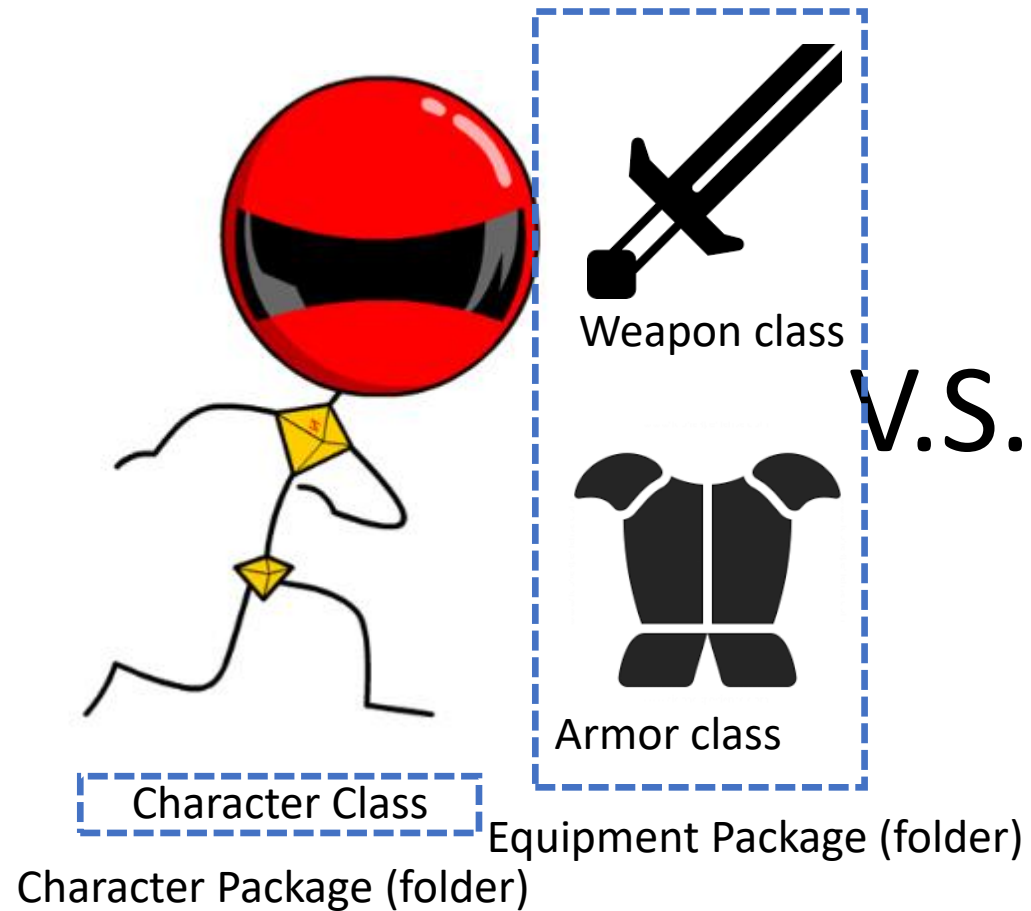
Before damaging

After damaging

Weapon and Armor info

Summary

Main Character



Enemy

