# 面向对象系统分析与设计课后作业01

## In an action adventure game, each character can make use of one weapon at a time, but can change weapons at any time during the game. Your job is to program according to the following class diagram.（一个动作冒险游戏，每个角色一次只能使用一种武器，但是可以在游戏的过程中换武器。下面是类图，请根据类图编写代码。）

package game;

public class AxeBehavior implements WeaponBehavior {

    /\*\*

     \* @see WeaponBehavior#useWeapon()

     \*/

    public void useWeapon() {

    }

}

package game;

public class BowAndArrowBehavior implements WeaponBehavior {

    /\*\*

     \* @see WeaponBehavior#useWeapon()

     \*/

    public void useWeapon() {

    }

}

package game;

public abstract class Character {

    protected WeaponBehavior weapon;

    public abstract void fight();

    public void setWeapon(WeaponBehavior w) {

        this.weapon = w;

    }

}

package game;

public class King extends Character {

    @Override

    public void fight() {

        System.out.println("King is fighting");

    }

}

package game;

public class KnifeBehavior implements WeaponBehavior {

    /\*\*

     \* @see WeaponBehavior#useWeapon()

     \*/

    public void useWeapon() {

    }

}

package game;

public class Knight extends Character {

    public void fight() {

        System.out.println("Knight is fighting");

    }

}

package game;

public class Queen extends Character {

    public void fight() {

        System.out.println("Queen is fighting");

    }

}

package game;

public class SwordBehavior implements WeaponBehavior {

    /\*\*

     \* @see WeaponBehavior#useWeapon()

     \*/

    public void useWeapon() {

    }

}

package game;

public class Test {

    public static void main(String[] args) {

        Character character = new King();

        character.setWeapon(new AxeBehavior());

        character.fight();

    }

}

package game;

public class Troll extends Character {

    public void fight() {

        System.out.println("Troll is fighting");

    }

}

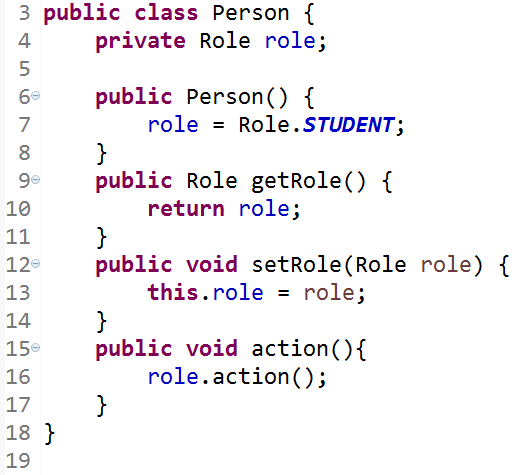
package game;

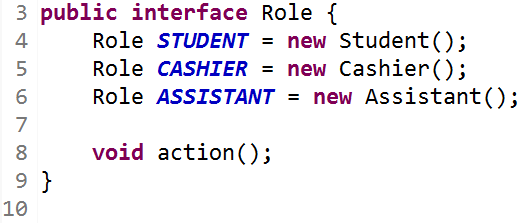
public interface WeaponBehavior {

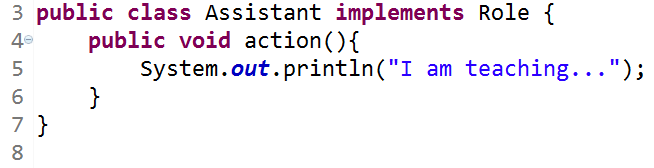
    public abstract void useWeapon();

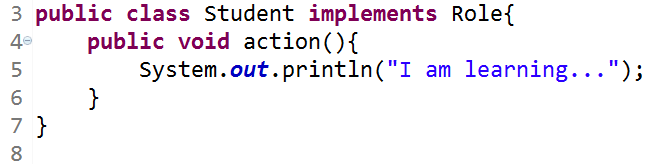
}

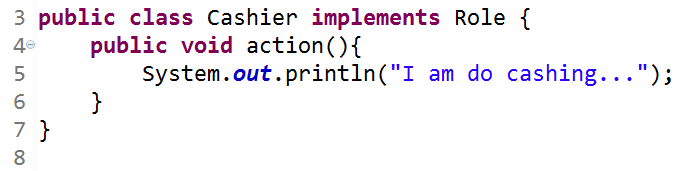
## In a university simulation system, a graduate can have three roles, and can change roles during the runtime, but can have only one role at a time. Your job is to draw the class diagram according to the following code in Java.（一个研究生在校期间，可以具有三种角色：学生、助教、收银员，但同一时刻只能具有一种角色。实现这个系统的代码如下，请根据代码画出类图。）

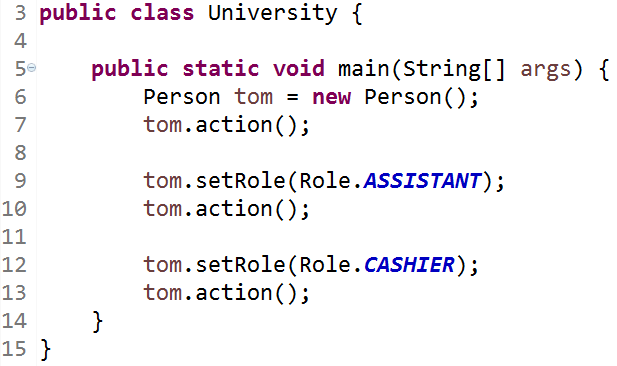


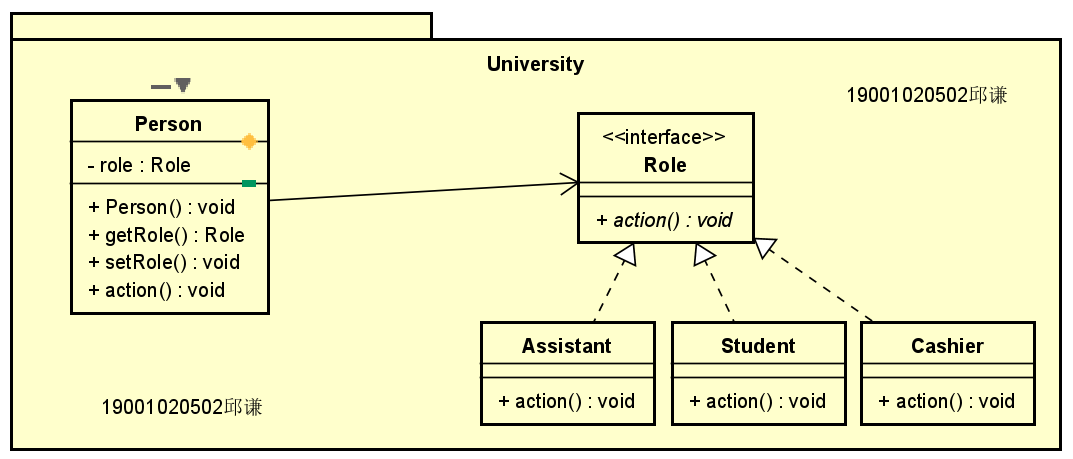












## In a an online bookstore system, there are a 10% discount on every computer book, a ￥2 discount on every language book, and a ￥10 discount for every ￥100 on all the novel books .（设计一个网上书店，该系统中所有计算机类图书(ComputerBook)每本都有10%的折扣，所有语言类图书(LanguageBook)每本都有2元的折扣，小说类图书(NovelBook)每100元有10元的折扣。请使用策略模式来实现系统，绘制类图并编程实现。）

1. package BookStore;
2. public interface Book {
3. public abstract double calculate(double price);
4. }

package BookStore;

public class ComputerBook implements Book {

    public double calculate(double price) {

        System.out.println("所有计算机类图书(ComputerBook)每本都有10%的折扣");

        return price \* 0.9;

    }

}

package BookStore;

public class LanguageBook implements Book {

    public double calculate(double price) {

        System.out.println("所有语言类图书(LanguageBook)每本都有2元的折扣");

        return price - 2;

    }

}

package BookStore;

public class NovelBook implements Book {

    public double calculate(double price) {

        System.out.println("小说类图书(NovelBook)每100元有10元的折扣");

        //计算有多少个100元

        int t = (int)price / 100;

        return price - t \* 10;

    }

}

package BookStore;

public class Price {

    private Book book;

    public double sum(double price) {

        return this.book.calculate(price);

    }

}

package BookStore;

public class Test {

    public static void main(String[] args) {

        Book book = new LanguageBook();

        System.out.println(book.calculate(102));

    }

}

