Java作业4

题目一

```
//声明接口ComputerWeight
interface ComputerWeight {
    double computeWeight();
}


//声明实现ComputerWeight接口的类Television，重写computeWeight()方法
class Television implements ComputerWeight {
    @Override
    public double computeWeight() {
        return 20.0;
    }
}


//声明实现ComputerWeight接口的类Computer，重写computeWeight()方法
class Computer implements ComputerWeight {
    @Override
    public double computeWeight() {
        return 5.0;
    }
}


//声明实现ComputerWeight接口的类WashMachine，重写computeWeight()方法
class WashMachine implements ComputerWeight {
    @Override
    public double computeWeight() {
        return 70.0;
    }
}


class Truck {
   ComputerWeight[] goods;
   double totalWeights = 0;
   Truck(ComputerWeight[] goods) {
       this.goods = goods;
   }

   public void setGoods(ComputerWeight[] goods) {
       this.goods = goods;
```

```
    }

    public double getTotalWeights() {
        totalWeights = 0;
        if (goods != null) {
            for (ComputerWeight good : goods) {
                if (good != null) { // 检查对象是否为空
                    totalWeights += good.computeWeight();
                }
            }
        }
        return totalWeights;
    }
}

public class CheckCarWeight {
    public static void main(String args[]) {
        ComputerWeight[] goods = new ComputerWeight[650]; // 650件货物
        for(int i = 0; i < goods.length; i++) { // 简单分成三类
             if(i % 3 == 0)
               goods[i] = new Television();
             else if(i % 3 == 1)
               goods[i] = new Computer();
             else if(i % 3 == 2)
               goods[i] = new WashMachine();
      }
      Truck truck = new Truck(goods);
      System.out.printf("\n货Weight of goods loaded on trucks:%-8.5f kg\n",
truck.getTotalWeights());
      goods = new ComputerWeight[68]; // 68件货物
      for(int i = 0; i < goods.length; i++) { // 简单分成两类
           if(i % 2 == 0)
             goods[i] = new Television();
           else
             goods[i] = new WashMachine();
      }
      truck.setGoods(goods);
      System.out.printf("Weight of goods loaded on trucks:%-8.5f kg\n",
truck.getTotalWeights());
    }
}
```

输出结果

src > CheckCarWeight.java > Language Support for Java(TM) by Red Hat > Computer

```
1   //声明接口ComputerWeight
2   interface ComputerWeight {
3       double computeWeight();
4   }
5
6   //声明实现ComputerWeight接口的类Television，重写computeWeight()方法
7   class Television implements ComputerWeight {
8       @Override
9       public double computeWeight() {
10          // 假设电视的重量为20kg
11          return 20.0;
12      }
13  }
14
15  //声明实现ComputerWeight接口的类Computer，重写computeWeight()方法
16  class Computer implements ComputerWeight {
17      @Override
18      public double computeWeight() {
19          // 假设计算机的重量为5kg
20          return 5.0;
21      }
22  }
23
24  //声明实现ComputerWeight接口的类WashMachine，重写computeWeight()方法
25  class WashMachine implements ComputerWeight {
26      @Override
```

问题 1　输出　调试控制台　终端　端口　筛选器(例如 text、!exclude、\escape)

```
Weight of goods loaded on trucks:20545.00000 kg
Weight of goods loaded on trucks:3060.00000 kg
```

题目二

```
//声明接口DogState
interface DogState {
    void showState();
}

//MeetEnemyState类实现接口DogState，重写public void showState()方法
class MeetEnemyState implements DogState {
    @Override
    public void showState() {
        System.out.println("狂叫，并冲上去咬敌人");
    }
}

//MeetFriendState实现接口DogState，重写public void showState()方法
class MeetFriendState implements DogState {
    @Override
    public void showState() {
        System.out.println("摇动尾巴，表示欢迎");
    }
}

//MeetAnotherDog实现接口DogState，重写public void showState()方法
```

```
class MeetAnotherDog implements DogState {
    @Override
    public void showState() {
        System.out.println("嬉戏");
    }
}

//编写Dog类
class Dog {
    private DogState state;
    public Dog() {
        this.state = new SoftlyState();
    }
    public void setState(DogState newState) {
        this.state = newState;
    }
    public void show() {
        state.showState();
    }
}

//SoftlyState类实现接口DogState，重写public void showState()方法
class SoftlyState implements DogState {
    @Override
    public void showState() {
        System.out.println("听主人的命令");
    }
}

public class CheckDogState {
    public static void main(String args[]) {
        Dog yellowDog = new Dog();
        System.out.print("狗在主人面前:");
        yellowDog.show();

        System.out.print("狗遇到敌人:");
        yellowDog.setState(new MeetEnemyState());
        yellowDog.show();

        System.out.print("狗遇到朋友:");
        yellowDog.setState(new MeetFriendState());
        yellowDog.show();

        System.out.print("狗遇到同伴:");
        yellowDog.setState(new MeetAnotherDog());
        yellowDog.show();
    }
}
```

输出结果

src > CheckDogState.java > Language Support for Java(TM) by Red Hat > MeetAnotherDog

```java
//声明接口DogState
interface DogState {
    void showState();
}

//MeetEnemyState类实现接口DogState，重写public void showState()方法
class MeetEnemyState implements DogState {
    @Override
    public void showState() {
        System.out.println(x:"狂叫，并冲上去咬敌人");
    }
}

//MeetFriendState实现接口DogState，重写public void showState()方法
class MeetFriendState implements DogState {
    @Override
    public void showState() {
        System.out.println(x:"摇动尾巴，表示欢迎");
    }
}

//MeetAnotherDog实现接口DogState，重写public void showState()方法
class MeetAnotherDog implements DogState {
    @Override
    public void showState() {
        System.out.println(x:"嬉戏");
```

问题　输出　调试控制台　终端　端口

```
ROG\Desktop\Java作业2\bin' 'CheckDogState'
狗在主人面前:听主人的命令
狗遇到敌人:狂叫，并冲上去咬敌人
狗遇到朋友:摇动尾巴，表示欢迎
狗遇到同伴:嬉戏
PS C:\Users\ROG\Desktop\Java作业2>
```