

## day02课后练习

### 基础题目

#### 第一题：概念辨析

1. 什么是接口，如何定义接口？

接口，是Java语言中一种引用类型，是方法的集合。

使用interface关键字定义接口，其中可以定义抽象方法，默认方法，私有方法，静态方法。

```
interface 接口名 {  
    // 抽象方法  
    // 默认方法  
    // 静态方法  
    // 私有方法  
}
```

#### 第二题：语法练习

- 语法点：接口
- 按步骤编写代码，效果如图所示：



- 编写步骤：
  1. 定义接口A，普通类B实现接口A
  2. A接口中，定义抽象方法showA。
  3. A接口中，定义默认方法showB。
  4. B类中，重写showA方法
  5. 测试类中，创建B类对象，调用showA方法，showB方法。
- 参考答案：

```
public class Test2 {  
    public static void main(String[] args) {  
        B b = new B();  
        b.showA();  
        b.showB();  
    }  
}  
  
interface A {  
    public abstract void showA();  
  
    public default void showB() {  
        System.out.println("BBBB");  
    }  
}
```

```
}  
}  
  
class B implements A {  
    @Override  
    public void showA() {  
        System.out.println("AAAA");  
    }  
}
```

## 第三题：语法练习

- 语法点：接口
- 按步骤编写代码，效果如图所示：

```
AAAA  
BBBB BBBB BBBB BBBB BBBB BBBB BBBB BBBB BBBB BBBB  
CCCC CCCC CCCC CCCC CCCC CCCC CCCC CCCC CCCC CCCC
```

- 编写步骤：
  1. 定义接口AA，普通类BB实现接口AA
  2. AA接口中，定义抽象方法showA
  3. AA接口中，定义私有方法show10 (String str) ，循环打印10次str
  4. AA接口中，定义默认方法show10B，show10C，分别调用show10方法，传入参数
  5. 测试类中，创建BB对象，调用showA方法，show10B方法，show10C方法
- 参考答案：

```
public class Test3 {  
    public static void main(String[] args) {  
        BB b = new BB();  
        b.showA();  
        b.show10B();  
        b.show10C();  
    }  
}  
  
interface AA {  
    public abstract void showA();  
  
    public default void show10B() {  
        show10("BBBB");  
    }  
  
    public default void show10C() {  
        show10("CCCC");  
    }  
  
    private void show10(String str) {  
        for (int i = 0; i < 10; i++) {  
            System.out.print(str + " ");  
        }  
    }  
}
```

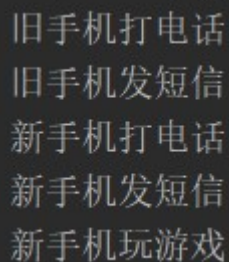


```
        System.out.println();
    }
}

class BB implements AA {
    @Override
    public void showA() {
        System.out.println("AAAA");
    }
}
```

## 第四题：语法练习

- 语法点：接口,继承
- 两个手机类OldPhone和NewPhone都有call()和sendMessage()方法.定义接口Play,Play中有一个抽象的玩游戏的方法playGame(),NewPhone实现Play接口有玩游戏的功能; 要求:分别测试OldPhone和NewPhone的call()和sendMessage()方法,再测试新手机palyGame()的方法
- 按步骤编写代码，效果如图所示：



旧手机打电话  
旧手机发短信  
新手机打电话  
新手机发短信  
新手机玩游戏

- 编写步骤
  1. 定义接口Play
  2. 在Play中定义一个抽象的玩游戏的方法playGame()
  3. 定义OldPhone类
  4. 在OldPhone类中定义call()和sendMessage()方法
  5. 定义NewPhone类,继承OldPhone,实现Play接口
  6. 创建旧手机对象
  7. 使用旧手机打电话
  8. 使用旧手机发信息
  9. 创建新手机对象
  10. 使用新手机打电话
  11. 使用新手机发信息
  12. 使用新手机玩游戏
- 参考答案：

```
public class Test4 {
    public static void main(String[] args) {
        // 6.创建旧手机对象
        OldPhone oldPhone = new OldPhone();
        // 7.使用旧手机打电话
        oldPhone.call();
        // 8.使用旧手机发信息
```



```
oldPhone.sendMessage();

// 9.创建新手机对象
NewPhone newPhone = new NewPhone();
// 10.使用新手机打电话
newPhone.call();
// 11.使用新手机发信息
newPhone.sendMessage();
// 12.使用新手机玩游戏
newPhone.playGame();
}
}

// 1.定义接口Play
interface Play {
    // 2.在Play中定义一个抽象的玩游戏的方法playGame()
    public abstract void playGame();
}

// 3.定义OldPhone类
class OldPhone {
    // 4.在OldPhone类中定义call()和sendMessage()方法
    public void call() {
        System.out.println("旧手机打电话");
    }

    public void sendMessage() {
        System.out.println("旧手机发短信");
    }
}

// 5.定义NewPhone类,继承OldPhone,实现Play接口
class NewPhone extends OldPhone implements Play {
    @Override
    public void playGame() {
        System.out.println("新手机玩游戏");
    }

    @Override
    public void call() {
        System.out.println("新手机打电话");
    }

    @Override
    public void sendMessage() {
        System.out.println("新手机发短信");
    }
}
```

## 第五题：语法练习

- 语法点：枚举

- 按步骤编写代码，效果如图所示：

benz牌的车正在往EAST方向行驶

- 编写步骤
  - 定义方向枚举Orientation，包含EAST, SOUTH, WEST, NORTH四个方向值
  - 定义小汽车Car类，包含String类型的品牌(brand)，Orientation枚举类型的方向(orientation)两个属性
  - 小汽车Car类包含行驶的方法(run)，在run方法中打印某个品牌的车正在往某个方向行驶
  - 定义测试类Test05，并包含main方法
  - 在main方法中创建Car对象，并调用Car对象的run方法
- 参考答案：

Orientation.java

```
// 方向枚举
public enum Orientation {
    EAST, SOUTH, WEST, NORTH;
}
```

Car.java

```
public class Car {
    private String brand;
    private Orientation orientation;

    public Car() {
    }

    public Car(String brand, Orientation orientation) {
        this.brand = brand;
        this.orientation = orientation;
    }

    public void run() {
        System.out.println(brand + "牌的车正在往" + orientation + "方向行驶");
    }

    public String getBrand() {
        return brand;
    }

    public void setBrand(String brand) {
        this.brand = brand;
    }

    public Orientation getOrientation() {
        return orientation;
    }

    public void setOrientation(Orientation orientation) {
        this.orientation = orientation;
    }
}
```



```
@Override
public String toString() {
    return "Car{" +
        "brand='" + brand + '\'' +
        ", orientation=" + orientation +
        '}';
}
}
```

Test05.java

```
public class Test05 {
    public static void main(String[] args) {
        Car c = new Car("benz", Orientation.EAST);
        c.run();
    }
}
```

## 第六题：语法练习

- 语法点：单例模式
- 使用两种单例模式编写AudioPlayer类

参考答案：

饿汉单例设计模式

```
class AudioPlayer {
    // 1.私有构造方法
    private AudioPlayer() {
    }

    // 2.定义成员变量保存创建的对象
    private static AudioPlayer instance = new AudioPlayer();

    // 3.提供公共的访问方式
    public static AudioPlayer getAudioPlayer() {
        return instance;
    }
}
```

懒汉单例设计模式

```
class AudioPlayer {
    private AudioPlayer() {
    }

    private static AudioPlayer instance;

    public static AudioPlayer getAudioPlayer() {
        if (instance == null) {
```

```
        instance = new VideoPlayer();  
    }  
  
    return instance;  
}  
}
```