

# 2020 年夏季 Java 小学期

## homework2

### 问答题

#### 【约定】

- 本题要求将答案写到答题卡.xlsx 的作答区域中。
  - 问题保证答题卡.xlsx 文档没有使用任何宏。
- 提交的答卷格式为 `xlsx`。如果不支持此格式，可以将文件转为答题卡.xls 或答题卡.csv。
  - 如果文件被保存为 `csv` 格式，请确保编码被设置为 `UTF-8`。但不建议在 `csv` 格式下作答，因为会丢失选择题的数据验证。
- 提交结果为半自动批阅，请不要私自修改非作答区域的内容，否则可能出现答案无法读取。
  - 可以修改行宽、列宽、字体、字号，但不要增加 `sheet` 或改变答案的行号、列号，也不可以合并单元格。
  - 建议直接使用默认字体，而不要使用自定义字体，自定义字体可能使用特殊编码导致内容无法读取。
- 回答多选题时请从左到右按照编号顺序填写选项，一个区域一个选项，多余区域保留空白选项。

问答题的描述见编程题 1.3，1.4。

### 编程题

#### 【约定】

时间限制：5 s。

空间限制：1 GB。（VM 参数：-Xmx1GB）

程序入口类为 `Main`。

#### 【第一题】OOP 基础知识——人类感受继承与多态

##### 题目描述

有一个接口 `Person`。

```
interface Person
{
    String getName();
    String getDescription();
    int changeSomething();
}
```

还有一个未知的类 `BasePerson`。

```
class BasePerson
{
}
```

## 题目要求

1.1 创建类 Man，该类实现接口 Person，并继承类 BasePerson。

(1) 该类具有三个成员变量，分别为 name，description，count。这三个变量均在构造函数中初始化。name 为保存实例名称的不变量，description 为保存描述信息的不变量，它们都是私有权限，类型为 String。count 为 Integer 类型变量，初始值为 0，这个变量是保护权限。

(2) Man 只有一个默认权限的构造方法，构造方法要求形参必须是(final String name, final String description)。

(3) 在类中增加一个保护的方法 move，没有参数和返回值，调用该方法时输出 “I'm moving...\n”。

(4) 定义接口 Person 中的方法，方法 getName()与 getDescription()均直接返回相应的域，方法 changeSomething 在调用时令 count=count-1，并返回 count。

1.2 创建类 SuperMan，该类继承 Man。

(1)重写父类方法，在调用 move 方法时输出 “I am flying...\n”。

(2)添加一个方法 fly，方法的访问权限为默认权限，没有参数和返回值。调用该方法时输出 “Fly! SuperMan!\n”。

(3)重写父类方法，在调用 changeSomething 方法时令 count=count+1，并返回 count; (count 初始值为 0)

(4)实现两个默认权限的构造方法，一个构造方法与 Man 的构造方法的形参相同。另一个构造方法没有任何参数，这个构造方法需要用 this 关键字调用前一个构造方法，用该构造方法实例化后的对象，其 getName()和 getDescription()必须返回 “superMan” 和 “I can fly.”。

## 提示

想要自己检查函数声明是否与题目描述符合，可以尝试使用工具 javap (jdk 自带)。

最基本的指令为: javap (class)

例如: javap Man

若要查看 private 成员，可以加上选项-p，即: javap -p Man

## 附加代码

src/Person.java

src/BasePerson.java

src/Man\_template.java

## 提交文件

src/Man.java

## 附加问答

1.3 (此题是问答题)

```
Man man=new Man("man","nothing");
SuperMan superman=new SuperMan("superman","nothing");
Person pman=new Man("pman","nothing");
Person psman=new SuperMan("psman","nothing");
Man msman=new SuperMan("msMan","nothing");
```

利用 1.1, 1.2 中定义的类, 将上述代码中变量 man, superman, pman, psman, msman 所能调用的方法和调用结果列出, 并综合分析产生该结果的方法调用过程。(假设这段代码与类的定义在同一个包)

说明:

- 变量 man 能调用的方法, 指可以直接用合法代码 man.\$function(...)调用的方法\$function。(不考虑 java.lang.Object 中定义的方法)
- 需要列出变量可调用的方法, 以及方法属于哪个类, 比如 man 可以调用类 Man 的方法 move, 但不可能调用类 Person 的方法 move, 因为变量 man 引用的类型是 Man, 该类型已经重写了父类 Person 的 move 方法。而变量 pman 可以调用 Person 的 move 方法, 因为引用的类型是 Person。
- 对每个调用, 还要指出实际被调用的方法, 比如 pman 可以调用 Person 的 move 方法。而 pman 引用的实例的实际类型是 Man, 所以实际调用的是类 Man 定义的 move 方法。

请在答题卡上, 把(方法 1)到(方法 5)、(类 1)到(类 10)所对应的选项填上。

变量 man 能调用的方法:

1. getName, 声明方法的类为 Man, 定义方法的类为 Man
2. getDescription, 声明方法的类为 Man, 定义方法的类为 Man
3. changeSomething, 声明方法的类为 Man, 定义方法的类为 Man
4. (方法 1), 声明方法的类为 Man, 定义方法的类为 Man

变量 superman 能调用的方法:

1. getName, 声明方法的类为 Man, 定义方法的类为 (类 1)
2. (方法 2), 声明方法的类为 (类 2), 定义方法的类为 Man
3. changeSomething, 声明方法的类为 (类 3), 定义方法的类为 SuperMan
4. move, 声明方法的类为 SuperMan, 定义方法的类为 SuperMan
5. (方法 3), 声明方法的类为 SuperMan, 定义方法的类为 (类 4)

变量 pman 能调用的方法:

1. getName, 声明方法的类为 Person, 定义方法的类为 (类 5)
2. .....

变量 psman 能调用的方法:

1. getName, 声明方法的类为 Person, 定义方法的类为 (类 6)
2. (方法 4), 声明方法的类为 (类 7), 定义方法的类为 SuperMan
3. ....

变量 msman 能调用的方法:

1. getName, 声明方法的类为 Man, 定义方法的类为 (类 8)
2. (方法 5), 声明方法的类为 Man, 定义方法的类为 SuperMan
3. move, 声明方法的类为 (类 9), 定义方法的类为 (类 10)
4. ....

方法选项:

- A. getName
- B. getDescription
- C. changeSomething
- D. move
- E. fly
- F. Person
- G. Man
- H. SuperMan

类选项:

- a. Person
- b. Man
- c. SuperMan
- d. Object

1.4 (此题是问答题)

由 1.1, 1.2 分析下列代码是否可行。

(1)

```
Man man=new Man("man","nothing");  
SuperMan sman=(SuperMan)man;
```

(2)

```
Man man=new SuperMan("superman","nothing");  
SuperMan sman=(SuperMan)man;  
Man man2=(Man)sman;
```

(3)

```
Person man=new Man("man","nothing");  
SuperMan sman=(SuperMan)man;
```

## 【第二题】OOP 设计模式——单例模式

### 题目描述

在java中, 有一个被经常提到的名词——设计模式, 设计模式是一套被反复使用的、多数人知晓的、经过分类编目的、代码设计经验的总结。使用设计模式是为了重用代码、让代码更容易被他人理解、保证代码可靠性。

### 题目要求

单例模式就是众多优秀设计模式中的一种。现在请你设计一个单例模式的类(写出类的java代码), 类名为 Singleton, 该类需要继承抽象类BaseSingleton, 成员变量和成员函数可以随意设计。要求该类只能创建一个实例, 并且外围程序只能通过调用公开的静态方法 Singleton getInstance()方法获得该类的实例。

### 题目解释

Singleton需要实现如题所述的方法getInstance()。

除了规定的方法getInstance()以外, 类Singleton的任何非私有的方法或非私有的成员, 都不能返回类型为Singleton的返回值。

在此基础上，该类的其他方法完全随意设计（不得有恶意行为，否则可能不得分），但不要输出任何信息。

### 附加代码

src/BaseSingleton.java  
src/Singleton\_template.java

### 提交文件

src/Singleton.java

### 【第三题】OOP 应用实践——考勤记录

#### 问题

学校有一台考勤机，老师和学生都会在上面打卡，全校师生人数总共  $n$  人，一共  $m$  条考勤记录，现在请你输出打卡次数最多的人的个人信息。

#### 输入

第一行为两个数字  $n$ ， $m$ ，代表有  $n$  个老师/学生和  $m$  条记录。

接下来  $n$  行，每行为如下两种格式之一：

Teacher 203 M 33 CS 分别代表（类别，教师号，性别，年龄，所教专业）

Student 123 ttt F 22 2016 分别代表（类别，学号，姓名，性别，年龄，入学年份）

接下来  $m$  行，每行一个字符和一个数字，表示打卡人类型和该人教师号/学号。（S 代表学生，T 代表老师）

#### 输出

输出一行，打卡次数最多的人的个人信息，行尾需要换行。答案保证唯一。

#### 例子

输入：

```
4 8
Teacher 203 M 33 CS
Student 123 ttt F 22 2016
Student 889 fff M 21 2017
Student 203 ggg F 19 2018
S 203
T 203
T 203
S 123
S 889
S 203
S 123
S 203
```

输出：

```
Student 203 ggg F 19 2018
```

## 规模

$1 \leq n \leq 1000$

$1 \leq m \leq 10000$

字符串长度均不超过 50，且仅包括大小写字母和数字

## 要求

本题要求面向对象编程，即包含类体、成员函数和成员变量，直接将逻辑写在 main 函数里不得分。

本题提供了抽象类 BaseStaff，表示一个员工的基本信息，并重写了员工的 hashCode 与 equals 方法。同学需要自己实现 1 到 2 个 BaseStaff 的派生类，用来记录老师/学生的信息。

本题目有一个特别要求：提交程序必须根据每行输入信息构建一个新的 BaseStaff 对象，即总共要创建 n+m 个 BaseStaff 对象。

## 建议

本题提供了一个 BaseStaff 的简单实现 NaiveStaff，在 NaiveStaff 中重写了 BaseStaff 的抽象函数，并在此基础上提供了打卡操作（punch），可以方便的记录打卡信息。

本题推荐使用 `HashMap<BaseStaff, NaiveStaff>` 来记录师生名单，并使用 NaiveStaff 的 punch 方法更新信息，对此无强制要求。

## 附加代码

src/BaseStaff.java

src/NaiveStaff.java

## 提交文件

src/Main.java

## FAQ

Q：为打卡记录创建一个继承 BaseStaff 的实例符合 OOP 思想吗？

A：符合。

OOP（Object Oriented Programming），面向对象编程本来就以对象为基础，打卡记录当然也应该使用对象。使用接口（interface）编程是为了提供服务，指定特定的功能需求。而用抽象类（abstract class）编程多数则是为了提供对象化的算法。本题的 BaseStaff 其实是抽象了基于员工的比较规则与哈希算法，以便于使用基于哈希的容器。

Q：将打卡记录对象作为查找哈希映射时使用的键值，和映射中的员工对象去比较，这符合 OOP 思想吗？

A：符合。

在 Java 的比较集合/映射、哈希集合/映射中，作为比较键值的等价对象很少是使用相同引用的，多数情况会创建新的对象。有人可能就此认为，构建多个不同类型的对象，却让他们互相等价，这不符合 OOP。其实我们应该转换一下思维，在 OOP 中，对象可以克隆、可以派生。克隆对象等价符合我们的思考方式，而不同的派生对象等价则符合 OOP 的方式：里氏（Liskov）替换原则——派生类在程序中可以任意替换基类；最少知识原则——算法应该对对象有最少的了解。As a supplement to ideology, please think about how Marxist realism interprets reality. And how to use Marxist realism to explain OOP thought.

## 【提交方法】

程序入口类名称为 Main，包名为默认，并放入相应的文件夹内。将解题过程写入实验报告中。关于程序的其他说明写入 readme.txt 中。并将以上文件打包为 zip 文件。提交至网络学堂。

文件打包如下：

zip 归档文件（命名为：学号\_姓名\_homework2.zip）

-- 答题卡.xlsx(.xls,.csv)

-- 实验报告.pdf（解题过程）

-- programming-1（编程题 1）

    -- src（目录内为提交的程序代码）

    -- readme.txt（可选，建议写）

-- programming-2（编程题 2）

    -- src（目录内为提交的程序代码）

    -- readme.txt（可选，建议写）

.....

## 【注意事项】

- 1、 代码要求自己写。代码提交后会进行查重，如果被认定为抄袭，课程将会被记为 0 分。
- 2、 按时提交。如有迟交，作业成绩在原有作业成绩基础上每迟交一天（不到一天按一天计）扣 10%。
- 3、 题目给分结果由最终测试决定。

不要相信简单的附加代码与样例结果，最终测试时，附加代码，输入、输出文件都会变得更严格。