

# Java 技术

(霍秋艳老师)

---

## 实验一

孙 晖

1513012 班 | 15130120141

## (一) 实验题目

面向命令的个人管理和日程程序

### 1. 第一题：面向命令的个人信息管理。

这个信息管理系统可以解决四个项目：待办事项，注解，约定，以及联系人。作业要求我们为每一个项目创建一个 `class`，而每一个类可以扩展出一个抽象基础类 (`abstract base class`)。除了创建四个类，你还需要创建一个类叫做 `manager`，这个类要能够支持一些简单的基于文本的命令，可以创建和管理项目。

**PIMTodo:** `Todo` 这个项目一定要由 `PIMEntities` 来定义，类名叫做 `PIMTodo`。每一个 `todo` 项目必须包含一个 (`string` 类型的) 优先级，一个日期，以及一个 (`string` 类型) 包含 `todo` 的内容。

**PIMNote:** 每一个 `note` 要包含一个优先级和一个用于存储文本的 `string` 类型。

**PIMAppointment:** 包含一个优先级，一个日期，以及一个说明，都是 `string` 类型。

**PIMContact:** 优先级，`firstname`，`lastname`，邮箱地址。

对于上面列出的 4 个条目类的实现还有一个额外的要求，包含日期的两个类必须共享一个您定义的接口。您必须正式地创建这个接口，并同时使用 `PIMTodo` 和 `PIMTodo` 来实现这个接口。

**PIMManager:** 你必须创建一个类叫做这个名字，包括一个 `main` 和提供一些方法来创建和管理这些类 (通过终端)。你必须提供以下的这些功能命令：

**List:** 打印出所有的 `PIM` 中的项目

**Create:** 添加一个新的项目

**Save:** 存储整个项目列表，就打印出来；复杂一点就是放在一个文件里。

**Load:** 从一个文件中读取这个 `list`。

### 2. 第二题：

创建一个类叫做 `Substring`，三个参数，第一个是 `string`，第二个是索引，第三个是长度，输出的就是要求的東西，类似于 `python` 中的列表：`Jello 1 3`，就输出 `ell`。

### 3. 第三题：

```
> java cal 4 2025
April 2025
Su Mo Tu We Th Fr Sa
      1  2  3  4  5
 6  7  8  9 10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28 29 30
```

日历生成程序：

Main 类的名字要叫做“cal”，可以输出标准的日历形式，给一个示例。

## (二) 实验思路

### 1. 第一题：

先把框架搭起来，这里我决定用两个.java文件，一个就是老师给的 PIMEntity，另一个就是 PIManager 文件。其中 PIMEntity 是老师给定的，我只需要在 PIManager 文件中搭好继承框架就可以了。

可以从给定的 PIMEntity 中推测出这一题的目的是为了考察继承。

我认为最难的地方有三个，一个是排版要按照老师给的例子来排版，另一个就是存储的问题，还有一个就是 interface 和继承的关系，有点迷茫，具体的其内部的变量和方法的默认限定有些模棱两可。具体的等到写完后看书总结一下。

### 2. 第二题：

我认为我应该先建立这样一种思想，就是创建一个类时，要想操控好这个类按照我的想法去做事，那么我的第一个想法应该是要创建方法来实现，而不是毫无头绪，当然可能随着我继续学习，对 Java 语言有更深入的了解会发现这种想法是错误的，到那个时候再说，至少现在先建立好这样一个反射，使得我在处理问题时不至于无从下手。

Static 修饰的成员变量或者方法都是独立于该类的任何对象的。就是说被修饰的变量或方法可以在对象被创建之前就可以使用。可以通过类名访问。

看别的同学用了 substring 这个方法，感觉非常的简单，老师在 HW2 中也提到了在 string 里找这种用法，但是我却没有找，具体来说，substring 的用法就是（字符串.substring（index，index+length））。

### 3. 第三题：

我的思路就是，先创建 cal 类，然后通过分析题目，不停的往里面填东西，老师给的有 tomorrow 例子。这里我百度了别人写的博客关于 calendar 的说明。Calendar 在 java 中是一个抽象类，这里为了方便我直接整理到我的博客里，对与

抽象类和接口进行一个小结，之后再说 calendar 类。

先把 tomorrow 中的 days 和 months 拷贝到 cal 类里面去，通过看老师给的代码总结出 daysinmonth 的用法，dayname 的用法，整理后放在实验总结里面，并且统统复制到 cal 类中，因为后面要用到。

这里需要注意的是 set 方法要在 get 的时候才会生效，所以需要用 clear 来清空缓存。

比较麻烦一点就是调日期之间的间隔，10 以下的数字占位置占的是一个字节的位置，10 和 10 以上占的是两个字节，不知道其他同学是否有办法解决这个问题，我只能在输出时加上空格，但是结果并不是很理想，总是对不齐。后来听了同学的方法，之间的空格用“\t”来代替空格，一下子好了很多，决定做一个鸵鸟，实行鸵鸟方案，如果有机会还是希望能有更好的方法来解决这个问题的。

### (三) 实验结果

#### 第一题：

```
D:\大三下学期\Java实验\第二次>java PIMManager
Welcome to PIM.
---Enter a command (suported commands are List Create Save Load Quit)---
List
There are 0 items.
---Enter a command (suported commands are List Create Save Load Quit)---
Create
Enter an item type ( todo, note, contact or appointment )
todo
Enter date for todo item:
04/20/2018
Enter todo text:
Submit java homework.
Enter todo priority:
urgent
---Enter a command (suported commands are List Create Save Load Quit)---
List
There are 1 items.
Item 1: TODO urgent 04/20/2018 Submit java homework..
---Enter a command (suported commands are List Create Save Load Quit)---
Save
Items have been saved.
---Enter a command (suported commands are List Create Save Load Quit)---
Quit
```

运行的结果如图所示，与老师给出的例子相差不大。

#### 第二题：

```
D:\大三下学期\Java实验\第二次>java Substring "hello world" 0 5
hello
```

运行结果如图所示。

### 第三题：

```
D:\大三下学期\Java实验\第二次>java cal 4 2018
April 2018
Su Mo Tu We Th Fr Sa
 1  2  3  4  5  6  7
 8  9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28
29 30
```

一开始我用空格来分隔，发现总是对不齐，后来听同学的讲解，改为"\t"分隔，就对齐了。

```
D:\大三下学期\Java实验\第二次>java cal 4 2018
April 2018
Su      Mo      Tu      We      Th      Fr      Sa
1        2        3        4        5        6        7
8        9       10       11       12       13       14
15       16       17       18       19       20       21
22       23       24       25       26       27       28
29       30
```

## (四) 实验总结

第一题：

1.继承：子类继承父类的特征和行为，使得子类具有父类的各种属性和方法。

2.继承的创建方法：

class 子类 extends 父类{}

3..next(): 只能读取空格、换行符等之前的第一个字符串；

.nextLine(): 能读取一整行的字符串。

4.scanner 类：是 Java5 的新特征，可以通过 scanner 类来获取用户的输入。创建的基本语法是：

Scanner s = new Scanner(System.in);//从键盘接受数据。

5.

BufferedReader bufReader = new BufferedReader(new InputStreamReader(System.in));

作用是缓冲 System.in 输入流，通过 InputStreamReader 将其转换为字符流。

第二题：

**抽象类方法：**是只有声明没有具体的实现的方法。

Abstract void fun();

**抽象类：**1.一个类中含有抽象方法就叫抽象类，抽象类必须在创建时用 abstract 关键字修饰，因为抽象类中含有无具体实现的方法，所以不能用抽象类创建对象。

2.Java 编程思想中提到，抽象类中可以不包含抽象方法，这是一个特殊情况，且这种情况不会有人用。3.抽象类中可以有成员变量和普通的成员方法。4.抽象方法必须为 `public` 或者 `protected`，因为 `private` 不能被子类继承，子类也就无法实现这种方法，缺省情况下默认为 `public`。5.抽象类不能用来创建对象。6.如果一个类继承于一个抽象类，那么子类必须实现父类的抽象方法。如果子类没有实现父类的抽象方法，则必须将子类也定义为 `abstract` 类。

**接口：**供别人调用的方法或者函数。定义接口的形式为：  
`[public] Interface InterfaceName{}`

**接口：**接口中可以含有变量和方法。但是接口中的变量会被隐式地指定为 `public static final` 变量，而方法会被隐式地指定为 `public abstract` 方法。

**让一个类遵循某组特地的接口需要使用 `implements` 关键字：**

`class ClassName implements Interface1,Interface2,...{}`

如果一个非抽象类遵循了某个接口，就必须实现该接口中的所有方法。对于遵循某个接口的抽象类，可以不实现该接口中的抽象类。

其实说再多，也会有遗漏，说再多，也会忘记，还是需要使用后才能加深理解。

第三题：

**1.import \* 和 import 某一个包中的类之间差别大不大？**

这里要说明的是，还是有差别的，直接引用包中的类或接口，这样名字是确定的，所以只需要按需查找就可以，而如果引用\*，那么当用到那个类的时候，就需要匹配最合适的例子，比如，我用到了 `File`，那么究竟是 `util.file` 还是 `io.file` 呢，这里编译器就需要匹配一下了，所以能限定好的，就限定好，这样省去了匹配的时间。当然，编译完一次之后就不用操心这件事了，这样看来，也就无所谓了。

**2. 下面就说明 `calendar` 类：**

`Calendar calendar = Calendar.getInstance();` //从一个 `calendar` 对象中获取 `date` 对象

`Date date = calendar.getTime();` //将 `Date` 对象反映到一个 `calendar` 对象中

**Set 方法：**

`Set (int,int,int,int,int,int)` 年，月，日，时，分钟，秒

月份起始值为 0，所以八月用 7 表示。

set(Calendar.MONTH,7)

calendar.getActualMaximum(int field)获得当年，月，周的天数。

```
public static final int ERA = 0;
public static final int YEAR = 1;
public static final int MONTH = 2;
public static final int WEEK_OF_YEAR = 3;
public static final int WEEK_OF_MONTH = 4;
public static final int DATE = 5;
public static final int DAY_OF_MONTH = 5;
public static final int DAY_OF_YEAR = 6;
public static final int DAY_OF_WEEK = 7;
```