2022面向对象编程与设计复习提纲

Part1 面向对象技术

（一）基本概念

1、面向对象基本思想；

2、面向对象的特性（抽象、封装、继承、多态）；

3、类的概念，类的组成，类与类之间的关系（继承、关联）；

4、UML类图，如何表示类及其之间的关系。

1. 综合能力

1、根据给定的需求，逐步确定系统所需的类，类的属性、方法，以及类与类之间的关系，并绘制相应的UML类图。

**Part2 Java程序设计**

（一）基本概念

1、Java基本语法；

2、Java类/抽象类/接口，及其属性、方法（含构造方法）的声明，明确每一个关键字的含义；

3、Java异常处理机制，掌握try、catch、finally各自的作用；

4、Java中类的继承，接口的实现；

5、Java多态性的体现；

6、泛型与常用集合类（List、Set、Map），迭代器（Iterator、Iterable），比较器（Comparator、Comparable）的用法；

7、常用Java API，Object、String、封装类等；

8、Java I/O相关概念。

9、Java UI编程

（二）综合能力

1、能够根据给定的UML类图，采用Java语言编程实现；

2、能够灵活处理一对多的类关系，选择适合的集合类，并通过迭代器实现遍历、通过比较器实现比较和排序；

3、掌握文件读、写的基本方法并能编程实现。

**Part3 设计模式**

（一）基本概念

1、单例模式的概念及其应用

2、策略模式的概念及其应用

3、工厂模式的概念

4、观察值模式的概念

5、适配器模式的概念

6、装饰者模式的概念

7、组合模式的概念

（二）综合能力

1、能够灵活应用单例模式并编程实现；

2、能够灵活应用策略模式，能根据给定的应用场景绘制其类图，并编程实现。

**复习题**

1. （填空）面向对象程序设计将描述事物的\_\_\_\_\_\_\_\_\_\_\_与\_\_\_\_\_\_\_\_\_\_\_封装在一起,作为一个相互依存、不可分割的整体来处理。
2. （填空）java类库中用于布局界面的布局管理器和相应的布局管理算法采用的设计模式为\_\_\_\_\_\_\_\_\_\_\_。
3. （设计方案）请给出下述文献管理系统的类图设计方案（类名、属性、方法等可以用英文字母描述，也可以采用汉语拼音）。

文献管理系统管理三种类型的文献：

1）会议论文（信息包括会议ID，论文标题，论文作者姓名）；

2）期刊论文（信息包括期刊名称，论文标题，论文作者姓名）；

3）图书（信息包括书籍名称，书籍标题，书籍作者姓名）。

1. （简述）给出策略模式的设计类图，说明策略模式的工作原理，并举例说明策略模式的应用。
2. （简述）简要叙述图1所示的所有类与类之间关系的含义。

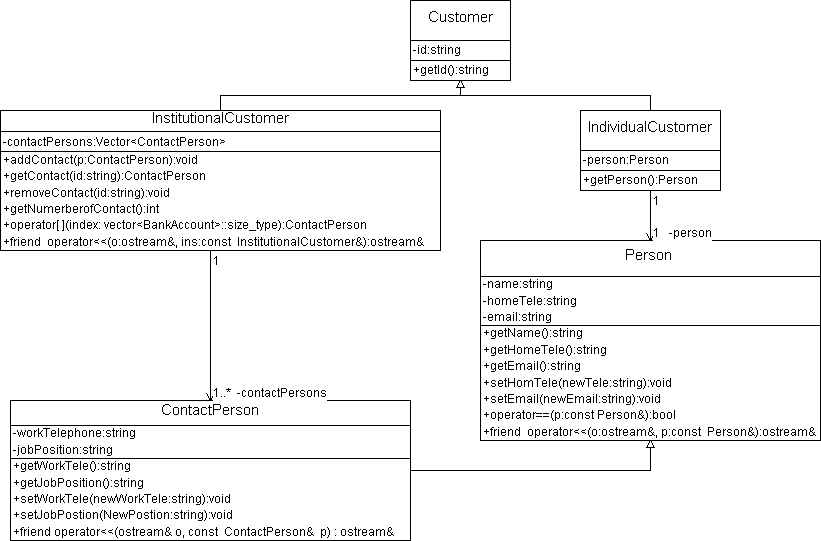


图1 类与类之间关系示意图

1. （选择）下列关于Java应用程序在结构上特点的描述中，错误的是（ ）
2. Java程序的一个编译单元（即一个.java源文件）可以由一个或多个类组成。
3. 组成Java程序的若干个编译单元可以通过包进行组织。
4. 无论类的访问权限如何，Java程序的文件名必须与文件中类名相同。
5. Java程序的一个编译单元可以包含若干带有main方法的类。
6. （选择）下列关于数组的定义形式，错误的是（ ）
7. String[] names = new String[3];
8. int[5] a; a=new int[5];
9. int[] ages = new  int[20];
10. String[] names = {"Bob", "Achebe", null};
11. （选择）有关类的静态变量和实例变量说法正确的是（ ）
12. 1）-7）中所有说法都是正确的。
13. 除了2）之外，所有描述都是正确的。
14. 除了2）与6）之外，所有描述都是正确的。
15. 除了 2）与4）之外，所有描述都是正确的。
16. 在类的定义中，如果属性变量前没有static关键字，则该属性变量是类的实例变量，否则该属性变量是类的静态变量。
17. 可以通过类名调用类的静态方法，但是不可以通过对象调用类的静态方法。
18. 类的每个对象都有一份实例变量的内存，类的所有对象共享静态变量的内存。
19. 统计某个类到目前为止共创建了多少对象，可以使用类的静态变量进行计数。
20. 在类的定义中，如果方法前没有static关键字，则该方法是类的实例方法，否则该方法是类的静态方法。
21. 类的实例方法形参有一个默认的this变量，实例方法体可以使用该this变量，该this变量指向调用该方法的对象。
22. 类的静态方法只能直接访问该类的静态变量或静态方法，否则必须通过该类的对象访问该类的实例方法或实例变量。
23. （填空）在Java输入\输出（即Java IO） 编程机制中，过滤输入流( 例如：类DataInputStream)与输入流（例如：InputStream）之间关系为\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_。
24. （填空）请初始化一个变量fileIn，可以实现从指定的文件a.txt中逐行读取文件内容：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_。
25. （选择）下列关于Java的异常处理机制的描述中，不完全正确的是（ ）。
    * 1. 在Java的异常处理机制中，异常类分为两大类，一类是继承Exception的异常类，另一类是继承RuntimeException的异常类，如果程序员定义一个新异常类，需要继承Exception或RuntimeException。
      2. 如果一个方法可能抛出异常，Java编译器要求该方法必须捕获(通过try和catch关键字)或者必须声明掷出(通过throws关键字)该类异常。
      3. Throw关键字用于抛出异常，目的是通知应用程序发生了错误，以给应用程序提供从问题中恢复和处理的机会。
      4. Catch关键字出现在try关键字之后，用于捕获和处理try代码块中可能抛出的异常，Catch关键字之后的“（ ）”必须指明它能捕获的异常类型，同时，它包含对异常进行处理的代码块。
26. （编程实现）请编程实现图2所示的类图。

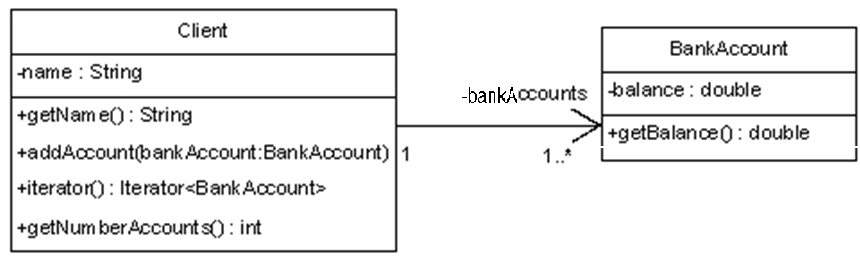
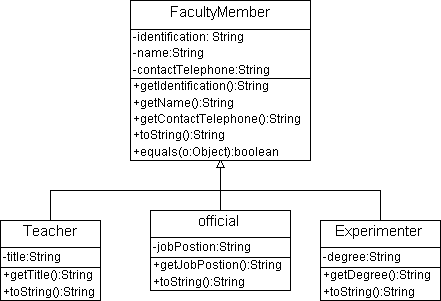


图2 银行业务系统的类图

1. （方法描述）以下述所示类图为例，阐述面向对象的多态性含义。

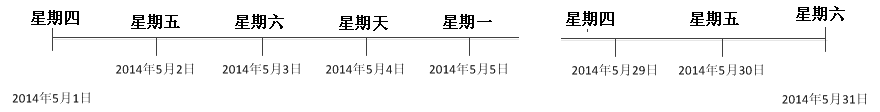


1. （方法描述）Java类库中有一个类java.util.StringTokenizer，该类可以实现的功能为：根据分割符将字符串划分为若干词汇单元，并可以逐一返回字符串内的词汇单元。该类的方法有：
2. public StringTokenizer(String str)
   * str是用户指定的要分割的字符串
   * 默认的分割符：空格，换行符和制表符等
3. public StringTokenizer(String str, String delim)
   * str是用户指定的要分割的字符串
   * delim是用户指定的分割符
4. public String nextToken()
   * 该类内部维护一个指针，该指针初始化时指向第一个被分割的词汇单元，如果该方法被调用，该指针指向的词汇单元作为字符串对象返回，同时指针指向下一个要分隔的词汇单元。
5. public boolean hasMoreTokens()
   * 测试是否还可以返回下一个词汇单元。

请阐述如何使用该类实现分隔01字符串：“0\_1\_0\_1\_1\_0\_1\_0\_0\_1\_1\_0\_1\_0\_1”的功能，并给出具体步骤和方法。

1. （设计模式应用）现需要设计一款灵活的月日历显示软件，目前支持三种月日历显示格式（见下述格式1-3），请采用所学设计模式给出该款软件的设计方案，设计方案要便于扩展，以适应该软件根据需要增加新月日历显示格式。

格式1：



格式2：



格式3（其中灰色部分对应的是星期六和星期天）：

