**期末复习知识点2**

Week6 – week12

考试形式：

手机屏幕截图

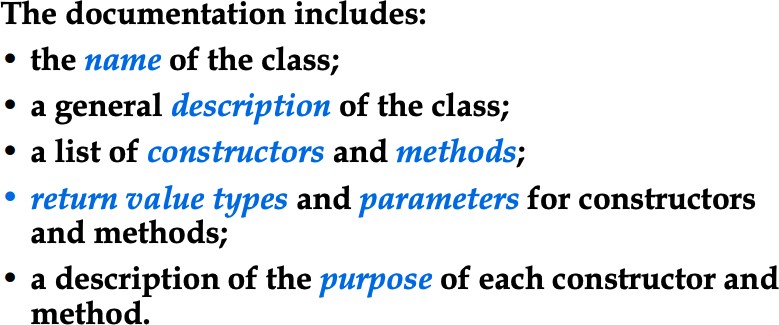
描述已自动生成

注意事项：一定不要花时间耗在改错题目上

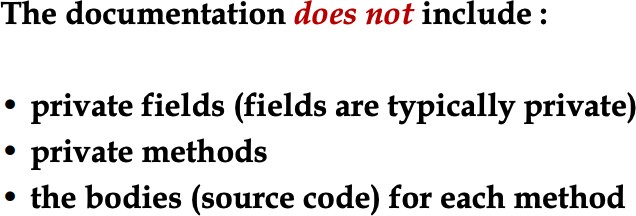
# Week6

第五周我们介绍了 library，知道可以利用别人写好的 class，那别人的东西肯定有个说明书，不然不知道怎么用

Java API **包含以下内容：**



Java API **不包含以下内容：**



String **这个特殊的数据类型**

*String 的比较，不使用==， 比较的是 memory address，比较 object 的值是否相同用*

*equals*

创建字符串的内容，StringBuffer **比** String **更加快速，**更加优化内存，在进行多个数据类型的拼接中，效率更高。

StringBuffer bufferName = new StringBuffer（”Jerry”）；

bufferName.append(“is”)

String name = “Jerry”;

String name = new StringBuffer(“Jerry”); 错误

String name = bufferName.toString();

**关于** String **常考的一些知识点：**

* trim()去掉首尾空格
* toUpperCase（）变成大写的形式
* 检查 String 当中是否包含数字

if(str.trim().length()==0)

* 检查 String 是否为空

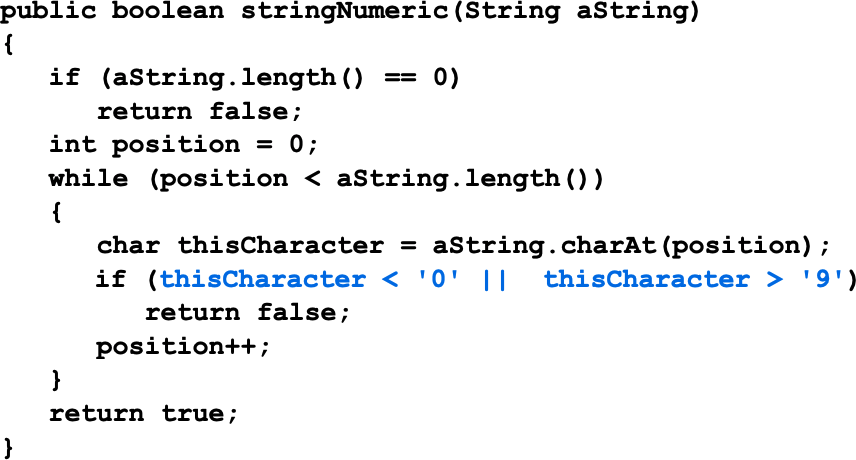
图形用户界面, 应用程序

描述已自动生成

图形用户界面, 文本, 应用程序

描述已自动生成

**检查** String **是不是是个数字类型的** String



图形用户界面, 文本, 应用程序

描述已自动生成

Iterator **迭代器**

图形用户界面, 文本, 应用程序

描述已自动生成

ArrayList 实现了 Iterator 的接口？如何理解实现接口？Iterator 中有的方法功能，

ArrayList 也可以用，有以下三种常见的 method

1、 hasNext（）；还有没有下一个元素

2、 next（）取到下一个元素3、 remove 删除元素

如何使用迭代器呢？

Iterator<Car> it = cardb.iterator();

While(it.hasNext())

{

Car car = it.next(); System.out.println( car.getRegNu() );

}

Iterator<Koala> it = koalaList.iterator();

While(it.hasNext())

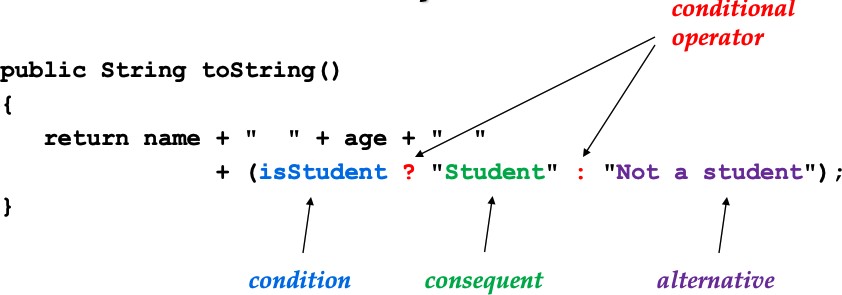
{

Koala koala = it.next(); System.out.println( koala.getStatus() );

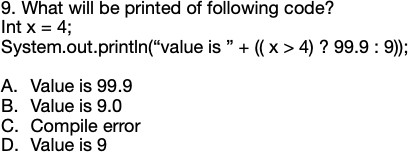
}

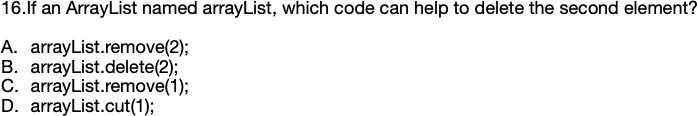
1. 变化过程：将ArrayList变成interator
2. 利用循环hasNext方法取出对应的元素
3. 用途

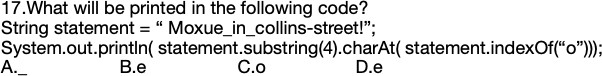
条件表达式：



**定长数组：**Array **上节课讲过了**

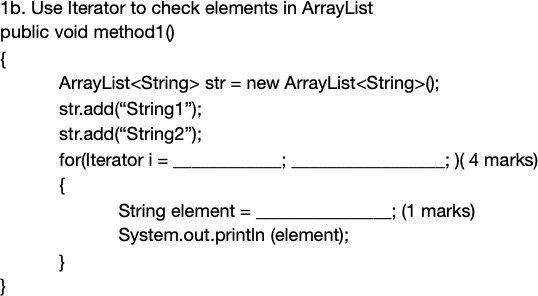
习题练习：





subString()从哪里开始切割

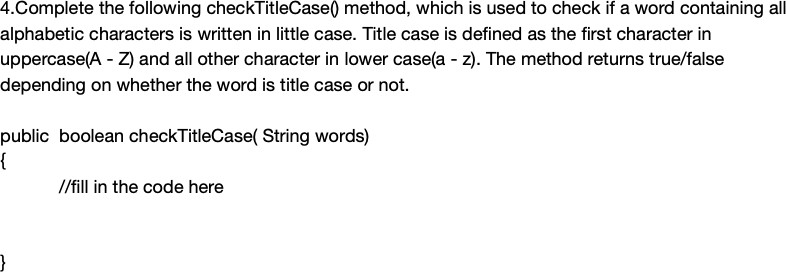
statement.indexOf(“o”)= 1

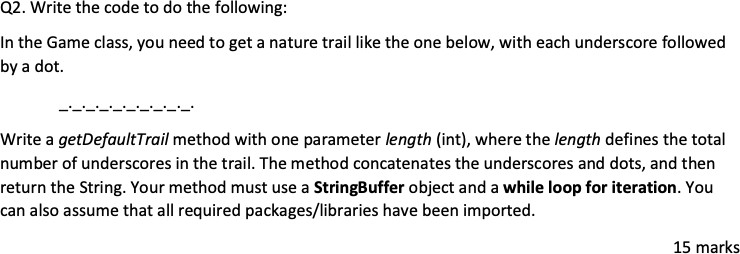


i.next()

i.hasNext()

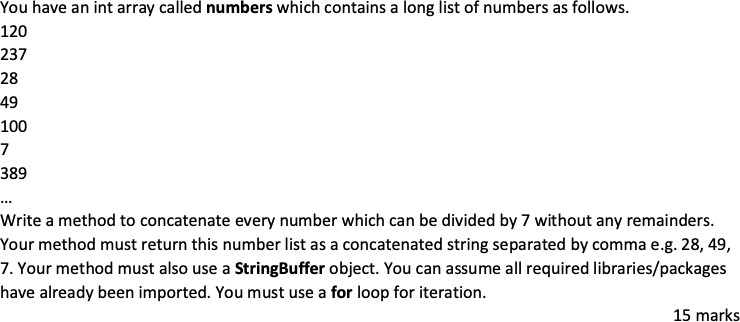
str.interator()

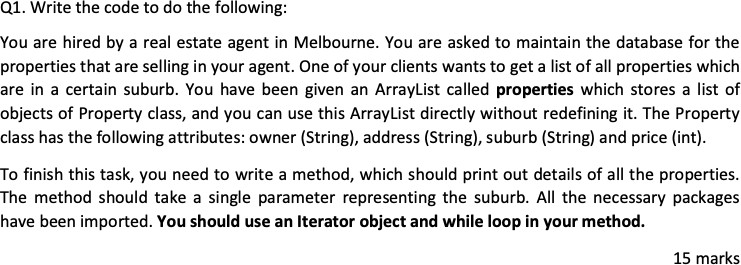




图形用户界面, 文本, 应用程序, 表格

中度可信度描述已自动生成





# Week7

程序设计的步骤：

1、 设计阶段

2、 应用阶段

3、 维护阶段

常见的错误：

1、 语法错误

2、 运行错误

3、 逻辑错误

Debugging 和 test 的区别

Debugging 是 try to make program no error before use, 随机性，持续性

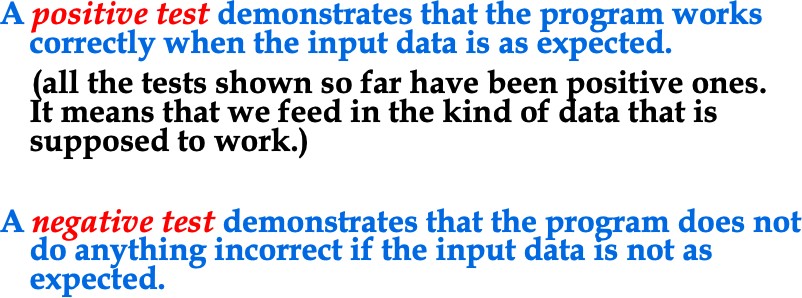
Test 是 try to find hidden error Test Strategy

1、 test plan ：你将会测试什么

2、 The actual test：测试的数据 + 实际测试的结果

positive test

negative test



# Week8

程序的封装：私有数据保护起来，留出 set/get 等缺口方法的有参数/无参数

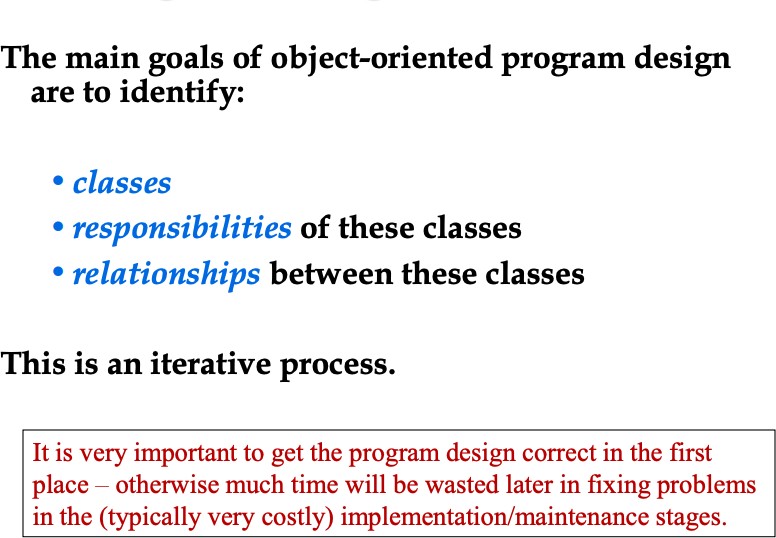
Scope 作用域

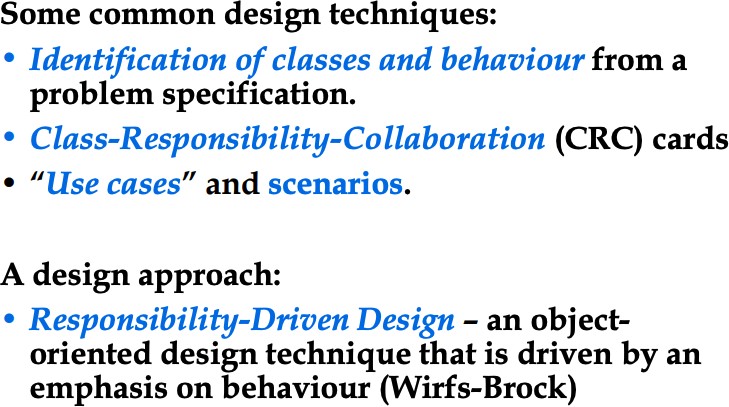
之前的章节都讲过

属于 class 的 method，public static 静态方法

NullPointerException

# Week9





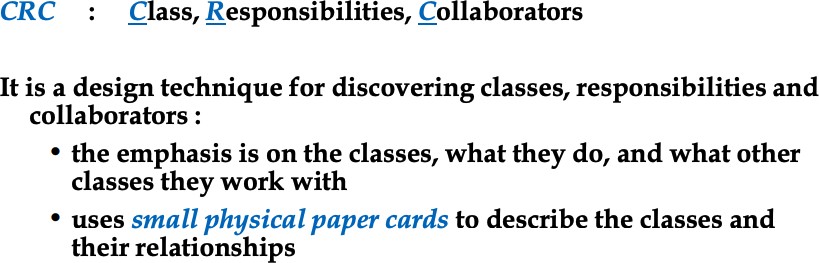
Because RDD focus on behaviours The first is the process of validation

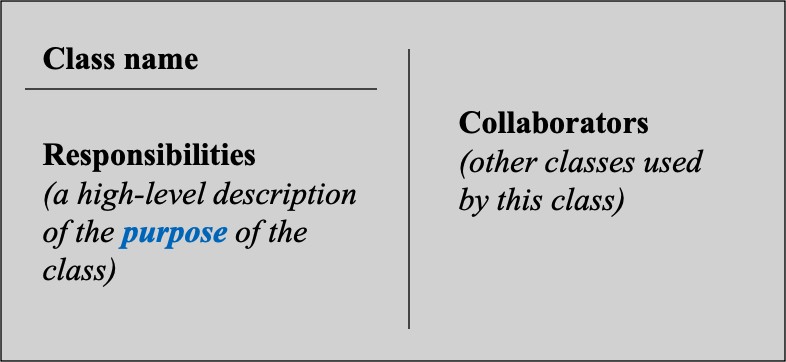
and the second is the process of assignment.

第一种：找出名次 – class ，动词 – method

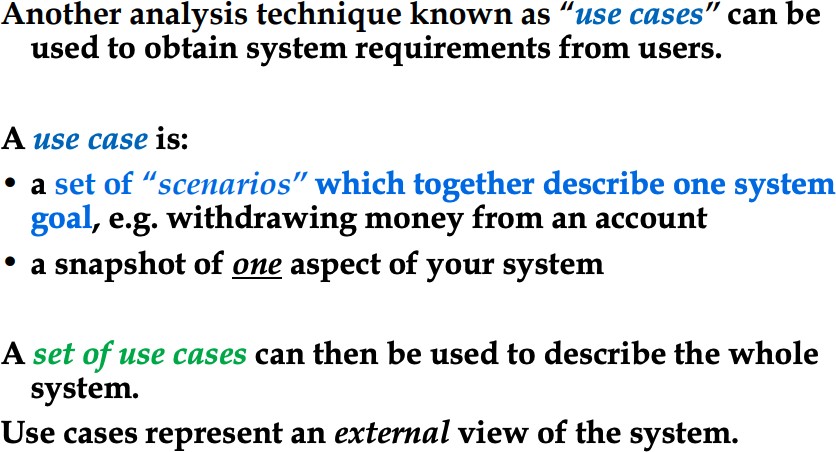
第二种：

CRC

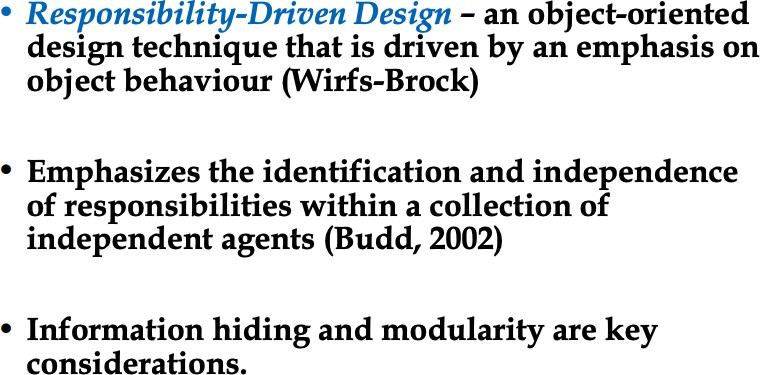




第三种：Use case



设计程序的方法论：责任驱动设计方法论。什么 class 专门负责做什么事情，然后想着如何通过 method 链接起来

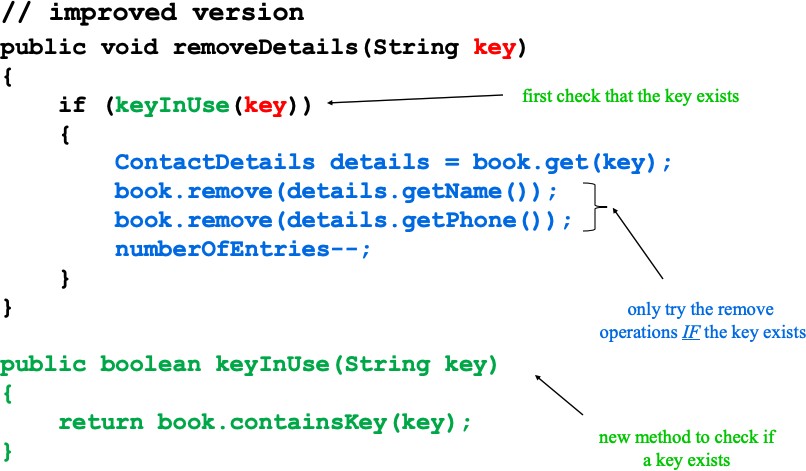


# Week10

客户端/服务器端开发模式

客户端经常调用服务端的方法，来完成某些功能

keyinuse



为了减轻 server 端的负担，我们就需要对错误进行预防，检查返回值

try catch finally block 预防一些错误发生

checked exception 指的是程序必须会处理的异常

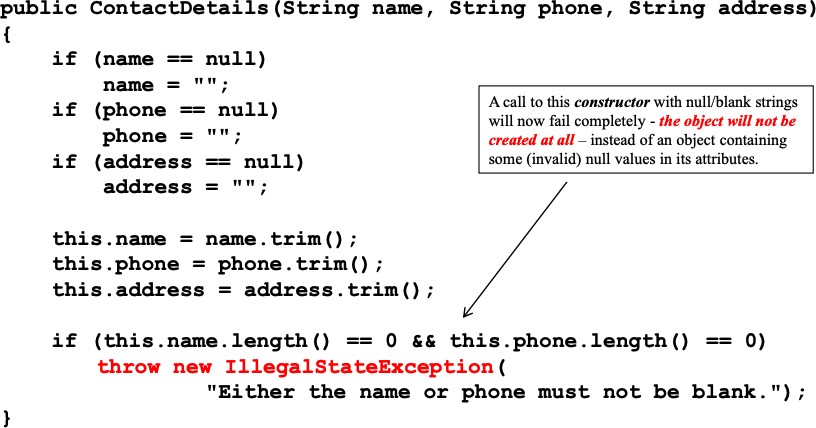
Refers to exceptions that the program must be able to handle

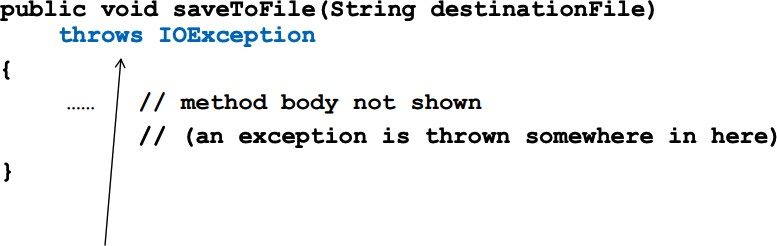
catch(Exception e)

un-checked exception 指的是不一定会执行的异常，可能会导致程序崩溃，只有当它运行到某一步才会检测异常，如果你没有写的很漂亮，比如一般我们不会写NullPointerException 异常，程序就崩溃了

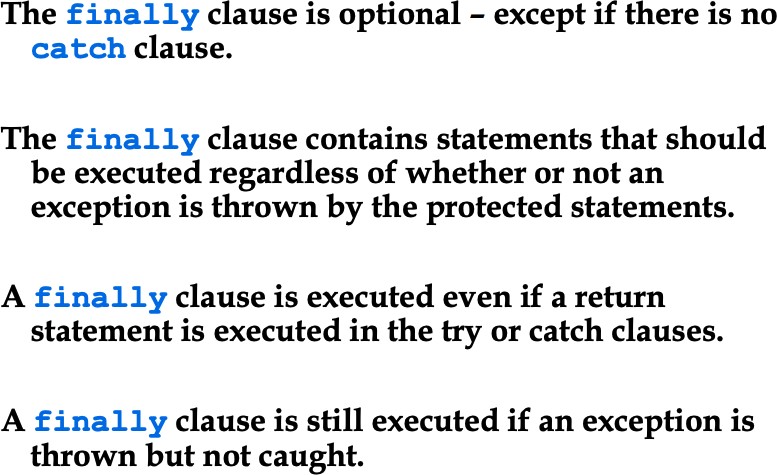
refers to exceptions that are not necessarily executed and may cause the program to crash,

and will only be detected when it runs to a certain point





try catch 的语法理解以及 finally 的特征



图形用户界面, 文本, 应用程序, 电子邮件

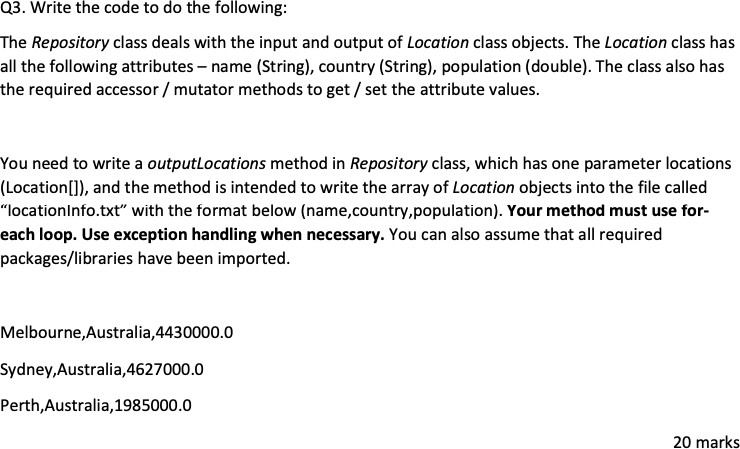
描述已自动生成

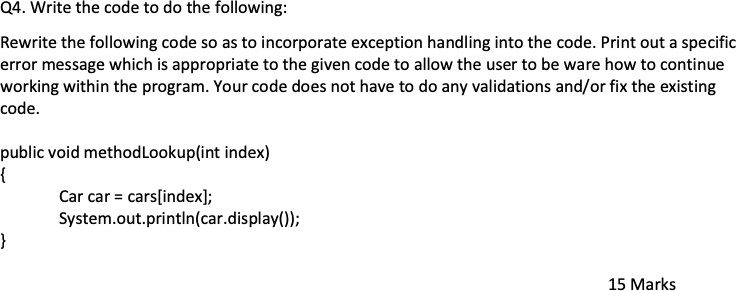
背景图案

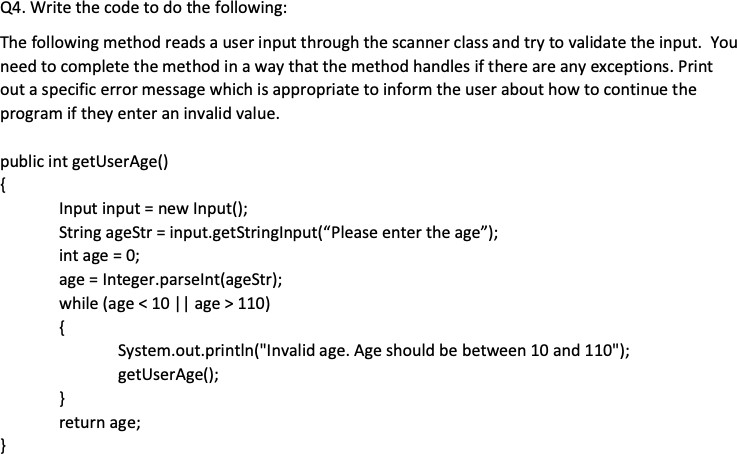
描述已自动生成

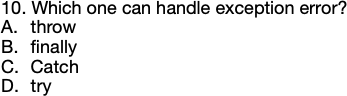
FileReader class 和 PrintWriter class

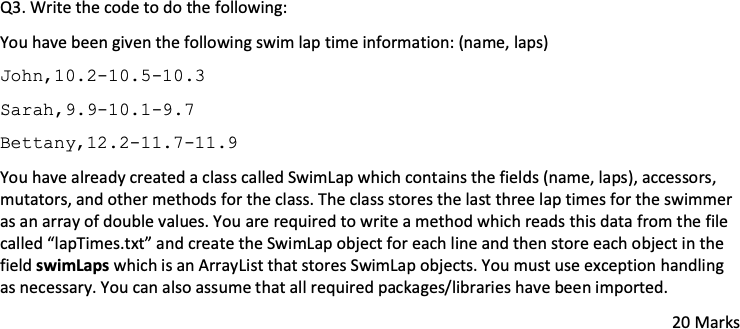
习题练习

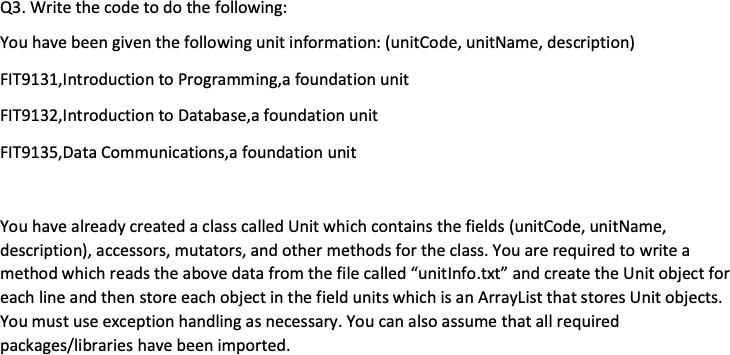












Cohesion/Couple 的理解

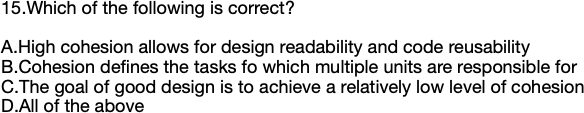
# Week11

cohesion:class 内部，每个class之间的关系

couple：不同class之间的关系

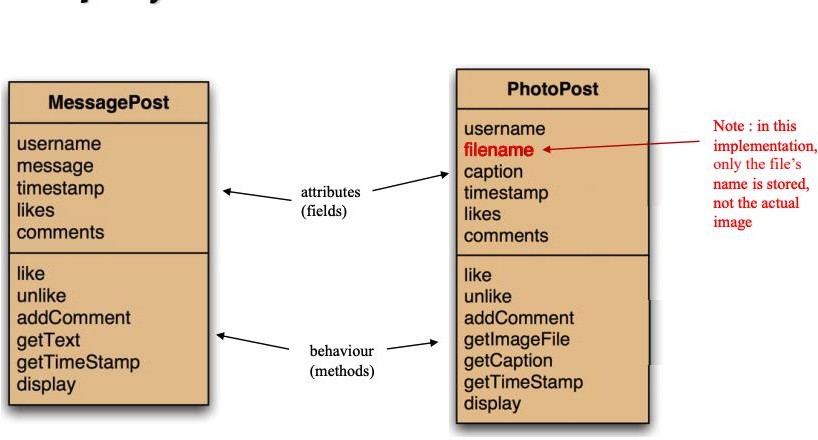
high cohesion low coupling

checkLetter(String name) high cohesion

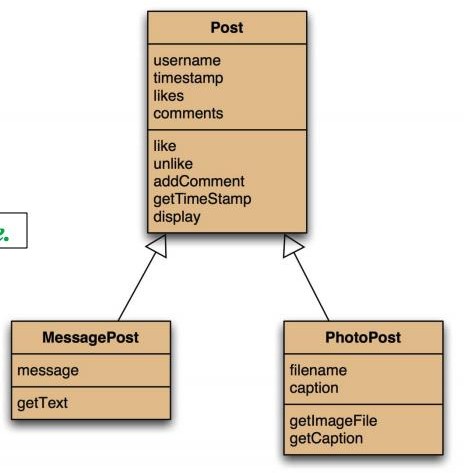


Week12

继承的举例



一看发现很多方法是类似/重复的，是不是可以复用一些代码？



图示

描述已自动生成

Public class MessagePost extends Post()

{

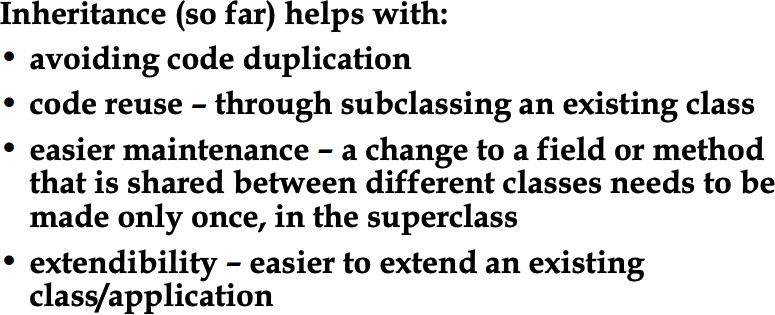
super关键字去初始化来初始化父类中的 field value super 关键字也可以用来使用父类的方法

}

overloading constructor 发生在同一个class

方 法 的 Overwrite （override）发生在不同的class之间Private/public/protected

继承的优点



Polymorphism多态

多态存在的必要条件：

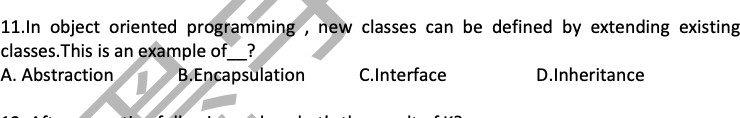
1、 发生了继承

2、 发生了重写

3、 父类对象引用指向之类对象



习题练习：



改错题：

