# Java07Task2

## 执行流程:

首先定义了一个 InsufficientFundsException 类，自定义了一个异常类作为 Exception 的子类。

在该类中定义了一个构造方法带有 message 参数

利用 super（）显示调用了父类的构造函数，并上传 message 参数。

在 BankAccount 类中:

定义了一个 double 类型参数 balance 并设置为私有。

定义了一个构造方法（参数类型为 double 的 initialBalance），然后利用 this 关键字将构造器中的参数 initialBalance 赋值给成员变量 balance。该构造方法设置为公开。

下一步进行了方法 getBalance（）的声明，该方法会返回一个双精度浮点数的值 即 return balance 返回 balance。该方法设置为公开。

再下步进行方法 withdraw（）的声明，该方法接收一个 double 类型的参数 amount。在该方法后跟了 throws 关键字说明这个方法会抛出一个异常，异常类名为 InsufficientFundsException。

因为该方法可能有异常，在方法内给予规定，如果 amount 的值比 balance 的值大，利用 throw 关键字，会抛出一个 InsufficientFunds Exception 对象，打印出（“余额不足，无法取款，当前余额·为：和 balance 对应的值），抛出该异常，与方法后跟的 throws 后的异常类型对应。

在将 balance 与 amount·相减的结果赋值给 balance，该类执行完毕。

公开定义了一个 BankAccountExample 类，公开定义一个 main 方法：
BankAccount account = new BankAccount（Math.random()*200）新建了一个 BankAccount 对象，其中 double 类型参数由 Math.random（）*200 随机返回一个 0 ~ 200 范围内的一个双精度浮点数。
打印输出“当前余额：+account.getBalance();”时，会正常通过 account 对象访问 getBalance（）方法，该步正常执行；
执行到 account.withdraw()时，由于会引用 withdraw（）方法，该方法是抛出过 InsufficientFunds 异常，所以谁引用该方法，就要在引用该方法外加上 try- catch 来处理，执行到该步会跳出 try，捕获到 catch 中的 InsufficientFunds 对象 e，执行“错误：”+通过对象 e 访问到 getMessage（）方法（利用传入的 message 参数）；
注意：“取款成功”的打印代码不会被执行！

最后打印输出“程序结束”