**声明自定义异常**

在 Java 中你可以自定义异常。编写自己的异常类时需要记住下面的几点。

所有异常都必须是 Throwable 的子类。

如果希望写一个检查性异常类，则需要继承 Exception 类。

如果你想写一个运行时异常类，那么需要继承 RuntimeException 类。

可以像下面这样定义自己的异常类：

class MyException extends Exception{

}

只继承Exception 类来创建的异常类是检查性异常类。

下面的 InsufficientFundsException 类是用户定义的异常类，它继承自 Exception。

一个异常类和其它任何类一样，包含有变量和方法。

实例

以下实例是一个银行账户的模拟，通过银行卡的号码完成识别，可以进行存钱和取钱的操作。

InsufficientFundsException.java 文件代码：

// 文件名InsufficientFundsException.java

import java.io.\*;

//自定义异常类，继承Exception类

public class InsufficientFundsException extends Exception

{

//此处的amount用来储存当出现异常（取出钱多于余额时）所缺乏的钱

private double amount;

public InsufficientFundsException(double amount)

{

this.amount = amount;

}

public double getAmount()

{

return amount;

}

}

为了展示如何使用我们自定义的异常类，

在下面的 CheckingAccount 类中包含一个 withdraw() 方法抛出一个 InsufficientFundsException 异常。

CheckingAccount.java 文件代码：

// 文件名称 CheckingAccount.java

import java.io.\*;

//此类模拟银行账户

public class CheckingAccount

{

//balance为余额，number为卡号

private double balance;

private int number;

public CheckingAccount(int number)

{

this.number = number;

}

//方法：存钱

public void deposit(double amount)

{

balance += amount;

}

//方法：取钱

public void withdraw(double amount) throws

InsufficientFundsException

{

if(amount <= balance)

{

balance -= amount;

}

else

{

double needs = amount - balance;

throw new InsufficientFundsException(needs);

}

}

//方法：返回余额

public double getBalance()

{

return balance;

}

//方法：返回卡号

public int getNumber()

{

return number;

}

}

下面的 BankDemo 程序示范了如何调用 CheckingAccount 类的 deposit() 和 withdraw() 方法。

BankDemo.java 文件代码：

//文件名称 BankDemo.java

public class BankDemo

{

public static void main(String [] args)

{

CheckingAccount c = new CheckingAccount(101);

System.out.println("Depositing $500...");

c.deposit(500.00);

try

{

System.out.println("\nWithdrawing $100...");

c.withdraw(100.00);

System.out.println("\nWithdrawing $600...");

c.withdraw(600.00);

}catch(InsufficientFundsException e)

{

System.out.println("Sorry, but you are short $"

+ e.getAmount());

e.printStackTrace();

}

}

}

编译上面三个文件，并运行程序 BankDemo，得到结果如下所示：

Depositing $500...

Withdrawing $100...

Withdrawing $600...

Sorry, but you are short $200.0

InsufficientFundsException

at CheckingAccount.withdraw(CheckingAccount.java:25)

at BankDemo.main(BankDemo.java:13)