

```
try
{
    // 程序代码
} catch (ExceptionName e1)
{
    //Catch 块
}
```

多重捕获

```
try{
    // 程序代码
} catch (异常类型1 异常的变量名1) {
    // 程序代码
} catch (异常类型2 异常的变量名2) {
    // 程序代码
} catch (异常类型2 异常的变量名2) {
    // 程序代码
}
```

- 当在try块或catch块中遇到return语句时，finally语句块将在方法返回之前被执行。当try语句和finally语句中都有return语句时，在方法返回之前，finally语句的内容将被执行，finally语句的返回值将会覆盖原始的返回值。
- catch不能独立于try存在。
- try代码后**不能**既没catch块也没finally块。
- try, catch, finally块之间不能添加任何代码。

以下4种特殊情况下，finally块不会被执行：

1. 在finally语句块第一行发生了异常。在其他行，finally块还是会得到执行
2. 在前面的代码中用了System.exit(int)已退出程序。exit是带参函数；若该语句在异常语句之后，finally会执行

3. 程序所在的线程死亡。
4. 关闭CPU。

throws/throw关键字：

一个方法没有捕获一个检查性异常，那么该方法必须使用throws 关键字来声明。throws关键字放在方法签名的尾部。也可以使用throw关键字抛出一个异常，无论它是新实例化的还是刚捕获到的。

抛出一个RemoteException异常：

```
import java.io.*;
public class className
{
    public void deposit(double amount) throws RemoteException
    {
        // Method implementation
        throw new RemoteException();
    }
    //Remainder of class definition
}
```

一个方法可以声明抛出多个异常，多个异常之间用逗号隔开。

```
import java.io.*;
public class className
{
    public void withdraw(double amount) throws RemoteException,
                                         InsufficientFundsException
    {
        // Method implementation
    }
    //Remainder of class definition
}
```

