

IO 流的应用

IO 流就是完成文件传输（上传文件：发朋友圈、换头像，文件下载：CSDN 下载源代码、文档）

字符 a 你好

文本类型的数据（txt、word、Excel、MD）可以使用字符去读取（当然也可以用字节）

```
package com.southwind.demo;

import java.io.*;

public class Test3 {
    public static void main(String[] args) throws
Exception {
        Reader reader = new
FileReader("/Users/southwind/Desktop/test.txt");
        BufferedReader bufferedReader = new
BufferedReader(reader);

        Writer writer = new
FileWriter("/Users/southwind/myjava/test.txt");
        BufferedWriter bufferedWriter = new
BufferedWriter(writer);

        String str = "";
        int num = 0;
        while ((str =
bufferedReader.readLine()) != null) {
```

```

        bufferedWriter.write(str);
        num++;
    }

    System.out.println("传输完毕，共读取
了"+num+"次");
    bufferedWriter.flush();
    bufferedWriter.close();
    writer.close();
    bufferedReader.close();
    reader.close();

}
}

```

非文本类型的数据（图片、音频、视频）不能用字符去读取，只能用字节去读。

```

package com.southwind.demo;

import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.InputStream;
import java.io.OutputStream;

public class Test {
    public static void main(String[] args) throws
Exception {
        //1、通过输入流将文件读入到 Java
        InputStream inputStream = new
FileInputStream("/Users/southwind/Desktop/1.png");
        //2、通过输出流将文件从 Java 中写入到 myjava

```

```
        OutputStream outputStream = new
FileOutputStream("/Users/southwind/myjava/1.png");
        int temp = 0;
        int num = 0;
        long start = System.currentTimeMillis();
        while((temp = inputStream.read()) != -1){
            num++;
            outputStream.write(temp);
        }
        long end = System.currentTimeMillis();
        System.out.println("传输完毕，共耗时" + (end -
start));
        outputStream.flush();
        outputStream.close();
        inputStream.close();
    }
}
```

反射

地位：Java 中最核心的模块，Java 之所以称为动态语言的关键，大部分类库、企业级框架底层都是通过反射来实现的，非常重要。

反射顾名思义就反转执行，生活中的反射就是通过虚像映射到具体的实物，可以获取到实物的某些形态特征。

程序中的反射，通过一个实例化对象映射到类。

一句话理解反射：常规情况下是通过类来创建对象的，反射就是将这一过程进行反转，通过对象来获取类的信息。

通过对象来获取类的信息

类的信息我们也同样使用对象来描述，Class 类专门用来描述其他类的类，每一个 Class 的实例化对象都是对某个类的描述。

Class 是反射的源头

如何来创建 Class 的对象？

1、调用 Class 的静态方法 `forName(String name)`，将目标类的全限定类名（全类名，带着包名的类名）

```
package com.southwind.demo2;

public class Test {
    public static void main(String[] args) throws
Exception {
        User user = new User();
        Class clazz =
Class.forName("com.southwind.demo2.User");
        System.out.println(clazz.getName());
        System.out.println(clazz.getTypeName());

        System.out.println(clazz.getSuperclass().getName());
;

        Class[] array = clazz.getInterfaces();
        System.out.println("*****");
        for (Class aClass : array) {
            System.out.println(aClass);
        }
    }
}
```

```
/Library/Java/JavaVirtualMachines/jdk-10.0.1
com.southwind.demo2.User
com.southwind.demo2.User
java.lang.Object
*****
interface java.io.Serializable
interface java.lang.Comparable
```

2、通过目标类的 class 创建，Java 中的每一个类都可以调用类.class，class 不是属性也不是方法，叫做“类字面量”，作用是获取内存中目标类型对象的引用（类的结构）。

```
Class clazz2 = User.class;
System.out.println(clazz2.getName());
```

3、通过目标类的实例化对象获取，getClass()

```
Class clazz3 = user.getClass();
System.out.println(clazz3.getName());
```