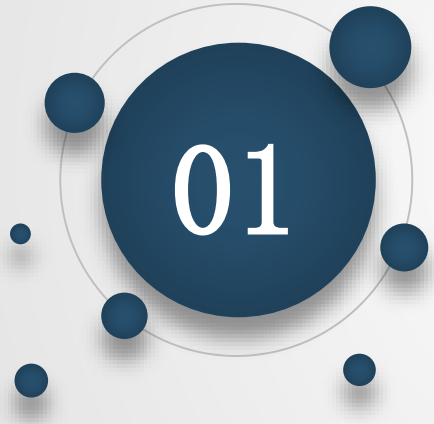

java.io

目录



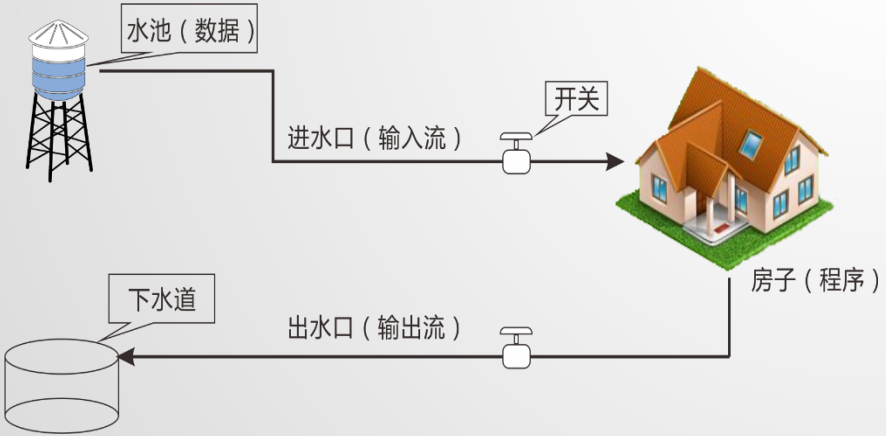
介绍

INTRODUCTION

IO简介

PART ONE

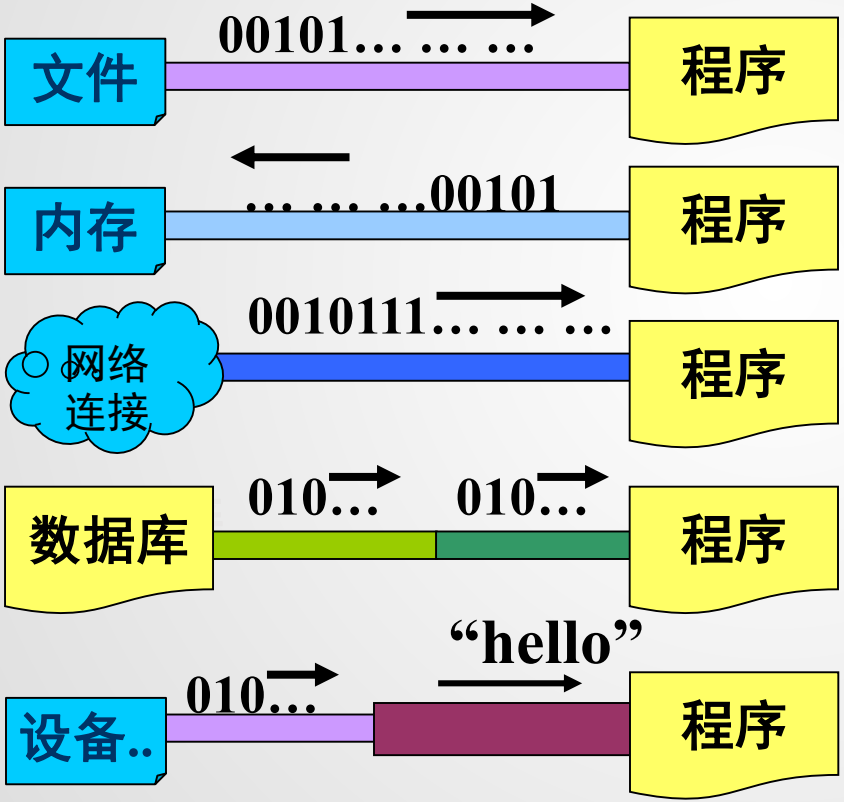
流



流:流动、流向，从一端移动到另一端。流是一个抽象、动态的概念，是一连串连续动态的数据集合。

数据源

data source。**提供原始数据**的原始媒介，常见的：数据库、文件、其他程序、内存、网络连接、IO设备。

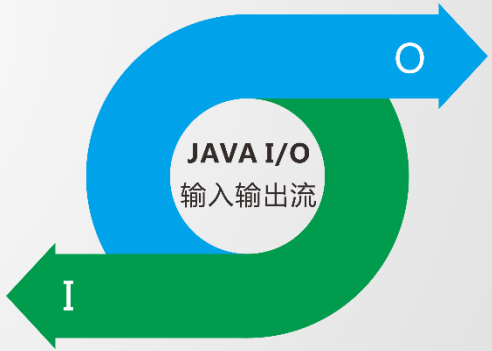


- 在Java程序中，对于数据的输入/输出操作以“流”（stream）方式进行；
- J2SDK提供了各种各样的“流”类，用以获取不同种类的数据；程序中通过标准的方法输入或输出数据。
- Java的流类型一般位于java.io包中

核心类

类	说明
File	文件类
InputStream	字节输入流
OutputStream	字节输出流
Reader	字符输入流
Writer	字符输出流
Closeable	关闭流接口
Flushable	刷新流接口
Serializable	序列化接口

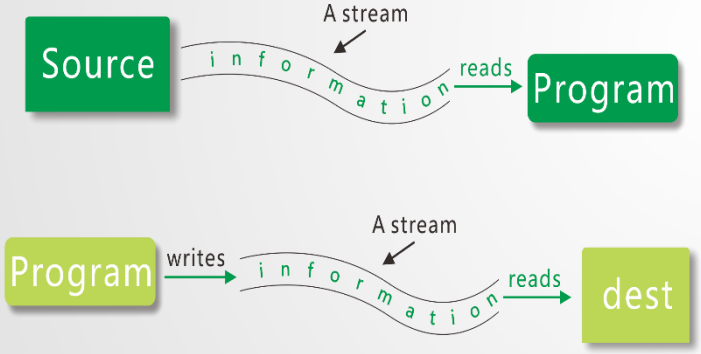
在整个Java.io包中最重要的就是5个类和3个接口，掌握了这些IO的核心操作那么对于Java中的IO体系也就有了一个初步的认识了。

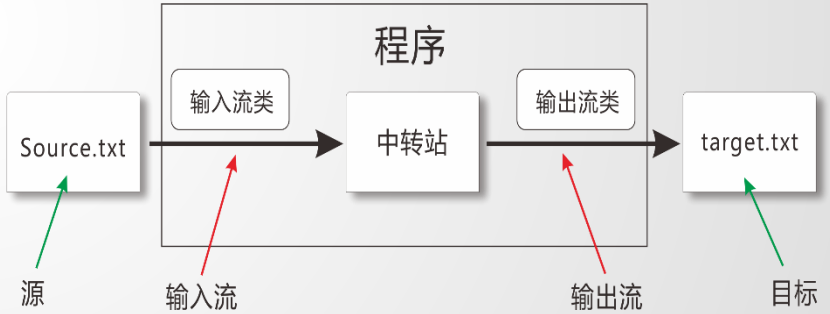


流分类

方向

- 输入流：数据源到程序(InputStream、Reader读进来)
- 输出流：程序到目的地(OutputStream、Writer写出去)

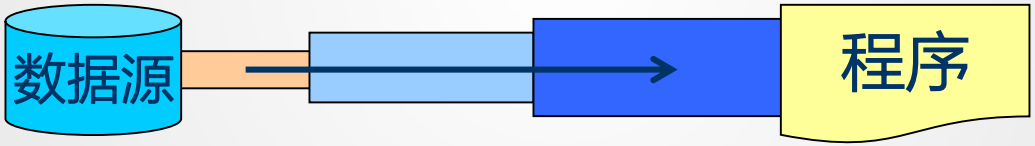




流分类

- 节点流：可以直接从数据源或目的地读写数据

- 处理流(包装流)：不直接连接到数据源或目的地，是其他流进行封装。目的主要是简化操作和提高性能。

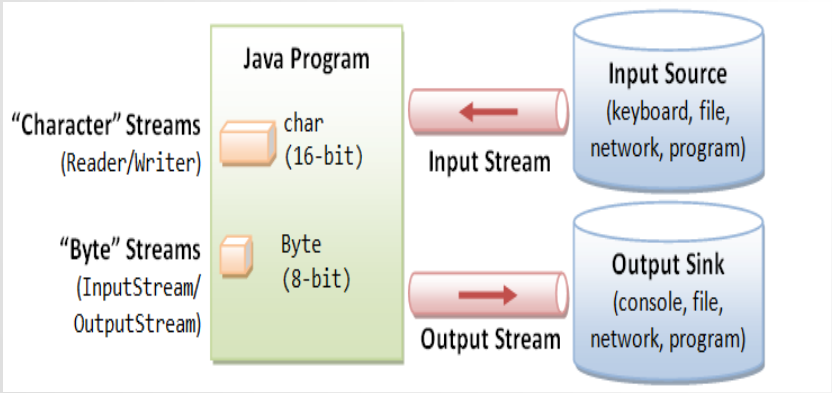


- 节点流和处理流的关系：
  ① 节点流处于io操作的第一线，所有操作必须通过他们进行；
  ② 处理流可以对其他流进行处理(提高效率或操作灵活性)。

流分类

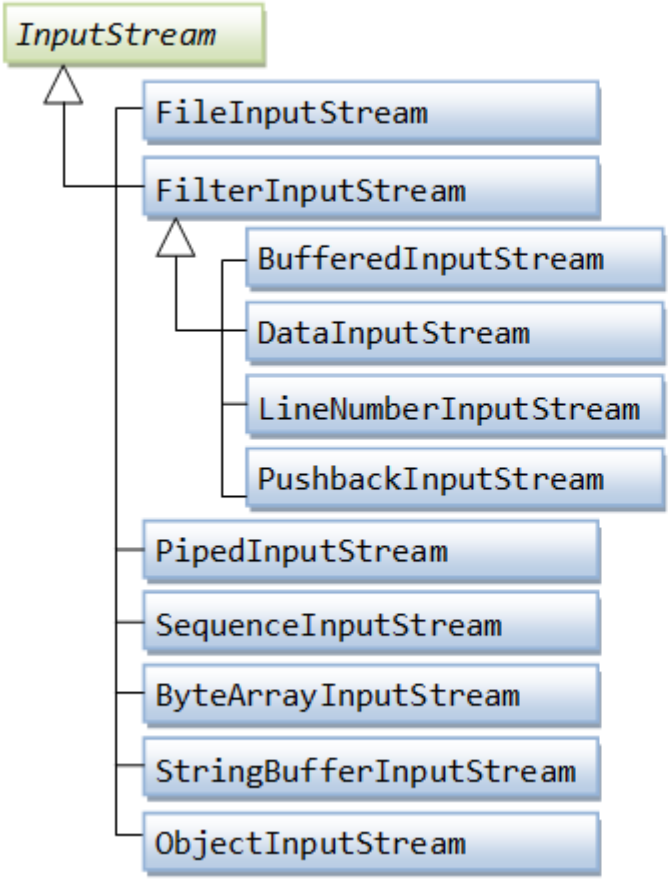
- 字节流：按照字节读取数据(InputStream、OutputStream)
- 字符流：按照字符读取数据(Reader、Writer），因为文件编码的不同，从而有了对字符进行高效操作的字符流对象。

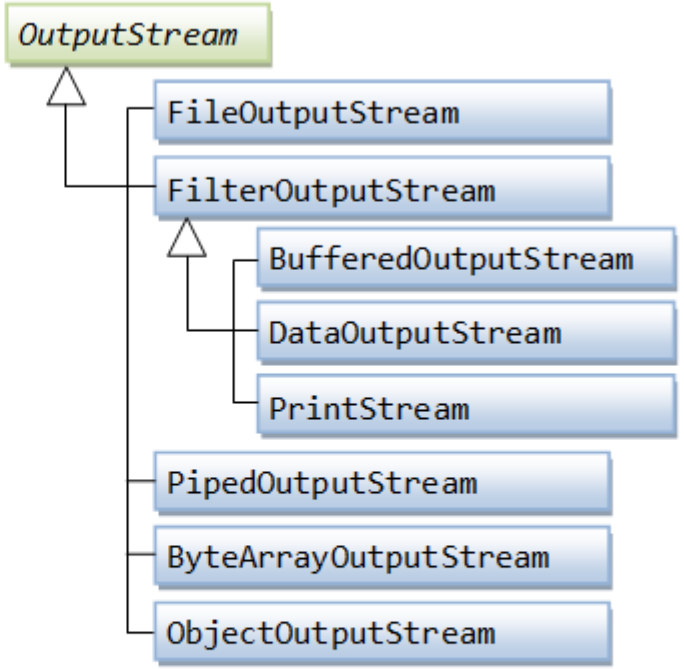
  原理：底层还是基于字节流操作，自动搜寻了指定的码表。



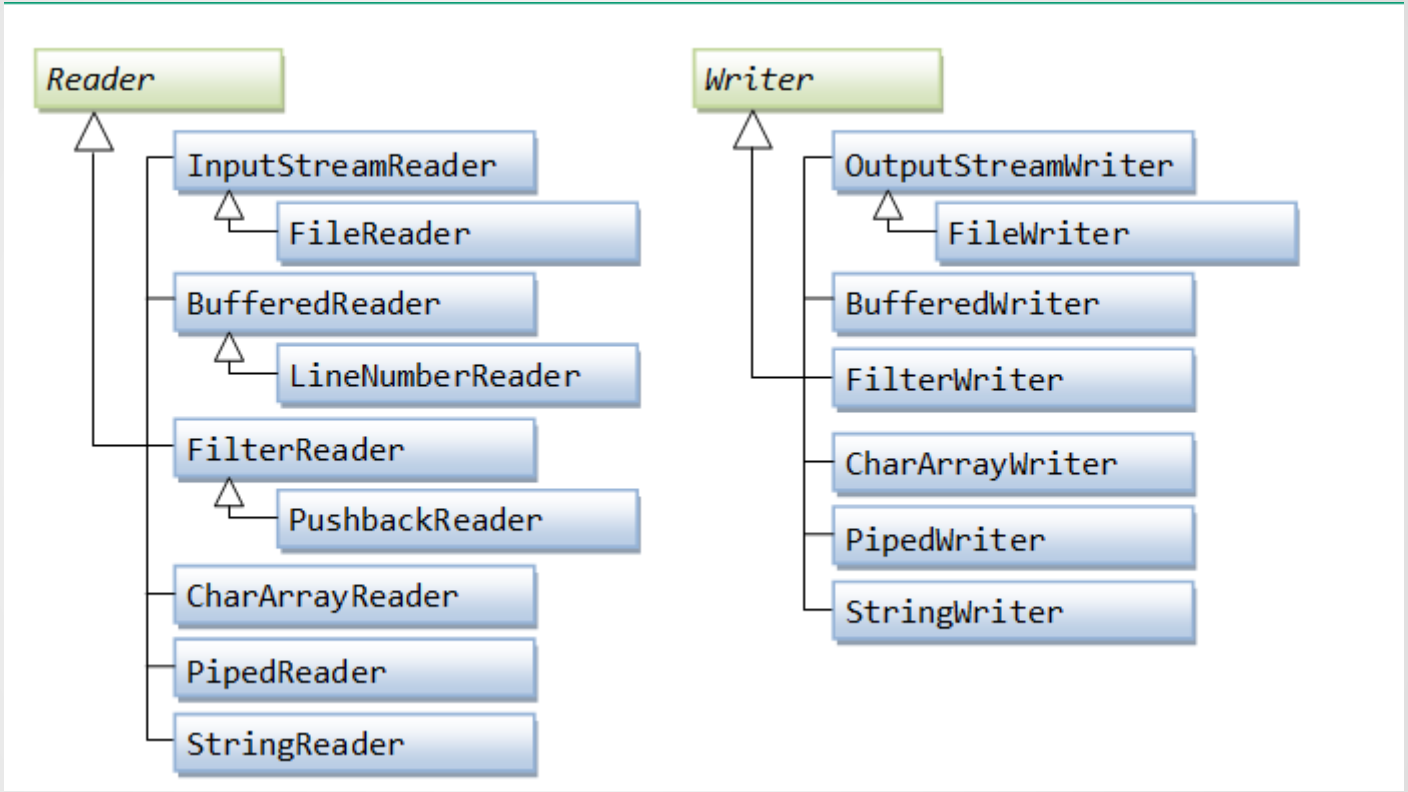
Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
32	20	040		Space	64	40	100	@	@	96	60	140	`	`
33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
36	24	044	$	$	68	44	104	D	D	100	64	144	d	d
37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
40	28	050	(	(	72	48	110	H	H	104	68	150	h	h
41	29	051	)	)	73	49	111	I	I	105	69	151	i	i
42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o

字节流


InputStream
FileInputStream
FilterInputStream
BufferedInputStream
DataInputStream
LineNumberInputStream
PushbackInputStream
PipedInputStream
SequenceInputStream
ByteArrayInputStream
StringBufferInputStream
ObjectInputStream

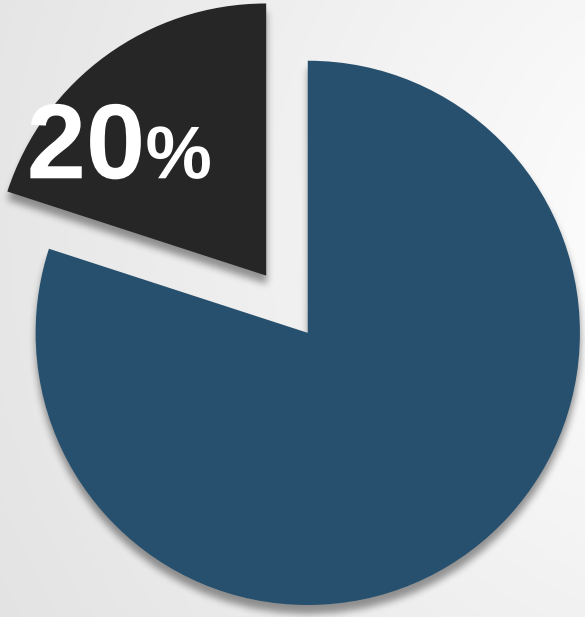

OutputStream
FileOutputStream
FilterOutputStream
BufferedOutputStream
DataOutputStream
PrintStream
PipedOutputStream
ByteArrayOutputStream
ObjectOutputStream

字符流


Reader
InputStreamReader
FileReader
BufferedReader
LineNumberReader
FilterReader
PushbackReader
CharArrayReader
PipedReader
StringReader
Writer
OutputStreamWriter
FileWriter
BufferedWriter
FilterWriter
CharArrayWriter
PipedWriter
StringWriter

学习方法

二八法则、量变到质变

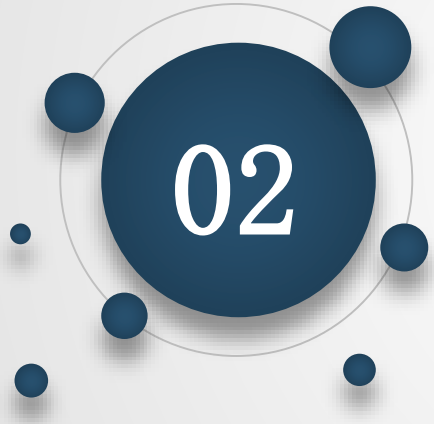


$$\begin{cases} 1.01^{365} = 37.8 \\ 0.99^{365} = 0.03 \end{cases}$$

这个等式告诉我们，积跬步以致千里，积怠惰以致溃败。

$$\begin{cases} 1.02^{365} = 1377.4 \\ 0.98^{365} = 0.0006 \end{cases}$$

这个等式则告诉我们，只比你努力一点的人，其实已经甩你太远。



File

常用API及文件编码

PART TWO

File

API	说明
pathSeparator separator	路径|路径分隔符
File(String parent,String child) File(File parent, String child) File(String name)	构造器 没有盘符以user.dir作为相对目录
getName() getPath() getAbsolutePath() getParent()	文件名、路径名
exists() isFile() isDirectory()	判断状态
length()	文件长度
createNewFile() delete()	创建新文件 删除文件

File

API	说明
mkdir() mkdirs()	创建目录，如果父目录链不存在一同创建
list()	下级名称
listFiles()	下级File
listRoots()	根路径

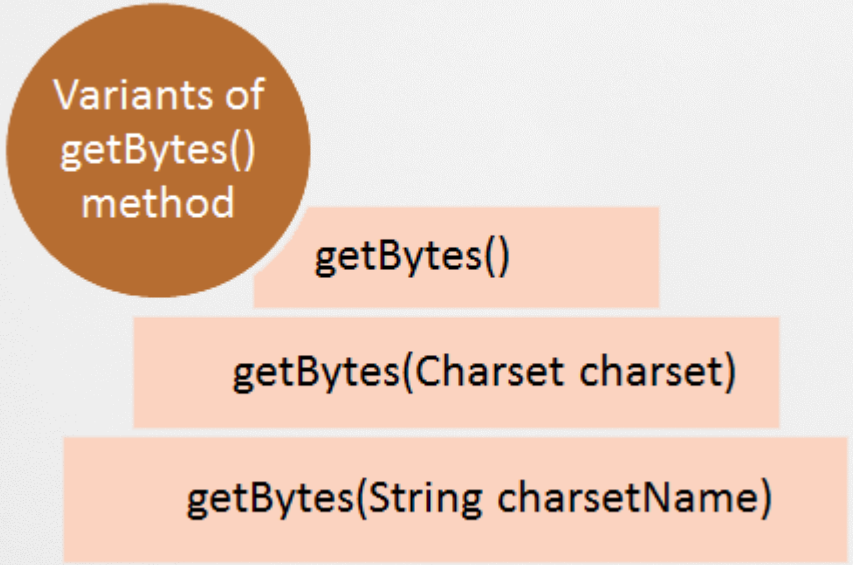
如何统计文件夹大小??

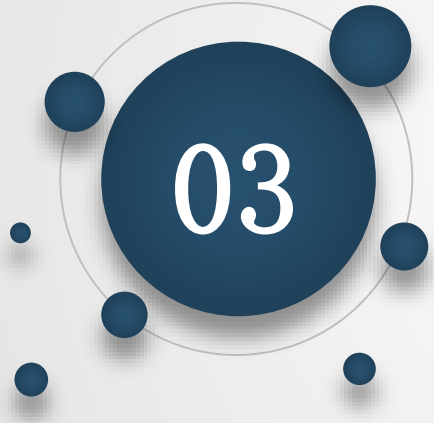
文件编码

字符集:Java字符使用16位的双字节存储，但是在实际文件存储的数据有各种字符集，需要正确操作，否则就有乱码的发生。

字符集	说明
US-ASCII	即英文的ASCII
ISO-8859-1	Latin-1 拉丁字符，包含中文、日文等
UTF-8	变长unicode字符(1-3个字节)，国际通用
UTF-16BE	定长unicode字符(2个字节)，大端Big-endian表示 **高字节低地址** 0x12 | 0x34 | 0x56 | 0x78
UTF-16LE	定长unicode字符(2个字节)，小端little-endian表示 **低字节低地址** 0x78 | 0x56 | 0x34 | 0x12
UTF-16	文件中开头指定大端还是小端表示方式，即BOM(Byte-Order-Mark) ：FE FF 表示大端，FF FE 表示小端.

文件编码


Variants of getBytes() method
getBytes()
getBytes(Charset charset)
getBytes(String charsetName)

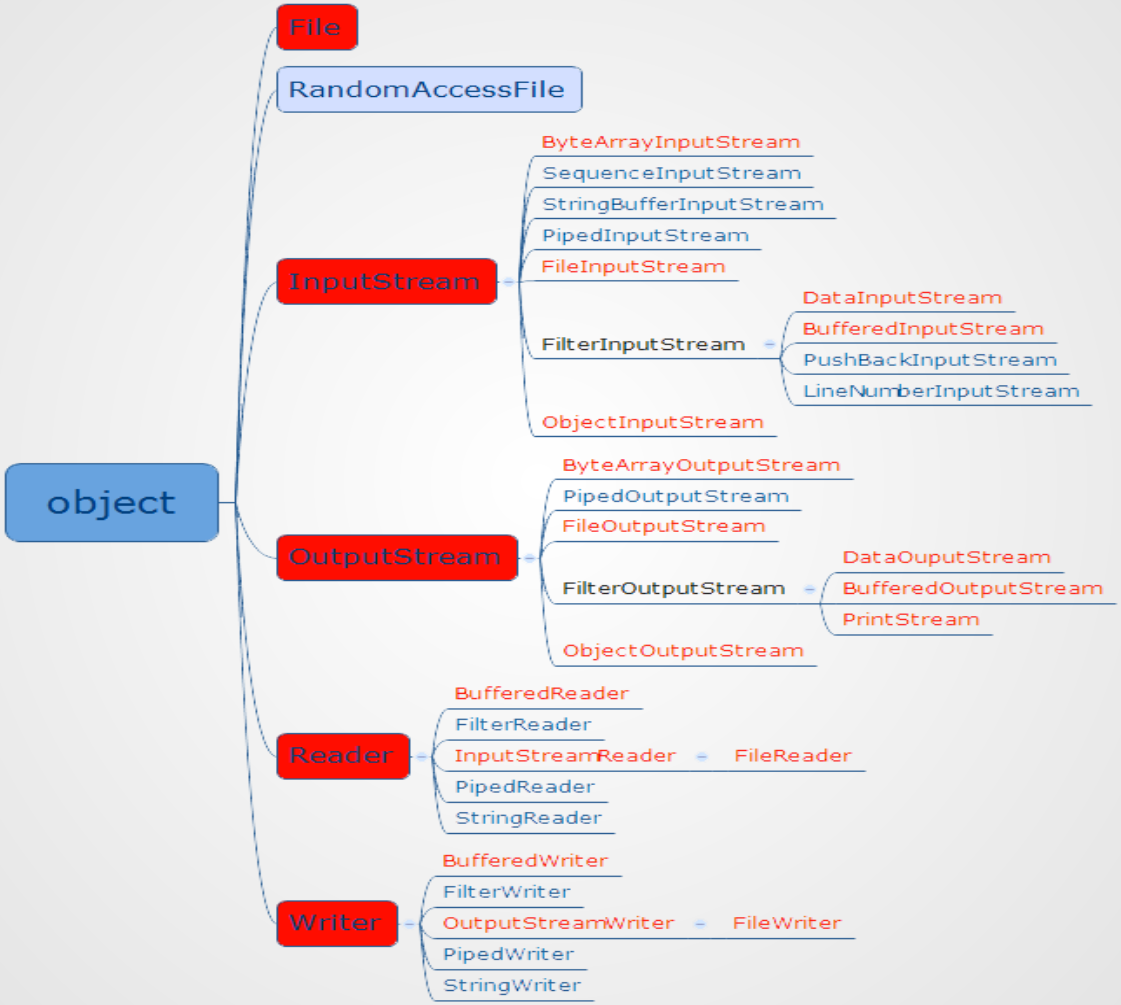


IO流

流读写操作

PART THREE

总揽


object
File
RandomAccessFile
InputStream
ByteArrayInputStream
SequenceInputStream
StringBufferInputStream
PipedInputStream
FileInputStream
FilterInputStream
DataInputStream
BufferedInputStream
PushBackInputStream
LineNumberInputStream
ObjectInputStream
OutputStream
ByteArrayOutputStream
PipedOutputStream
FileOutputStream
FilterOutputStream
DataOuputStream
BufferedOutputStream
PrintStream
ObjectOutputStream
Reader
BufferedReader
FilterReader
InputStreamReader
FileReader
PipedReader
StringReader
Writer
BufferedWriter
FilterWriter
OutputStreamWriter
FileWriter
PipedWriter
StringWriter

四个抽象类

抽象类	说明	常用方法
InputStream	字节输入流的父类，数据单位为字节。	• int read() • void close()
OutputStream	字节输出流的父类，数据单位为字节。	• void write(int) • void flush() • void close()
Reader	字符输入流的父类，数据单位为字符。	• int read() • void close()
Writer	字符输出流的父类，数据单位为字符。	• void write(String) • void flush() • void close()

第一个程序

创建源

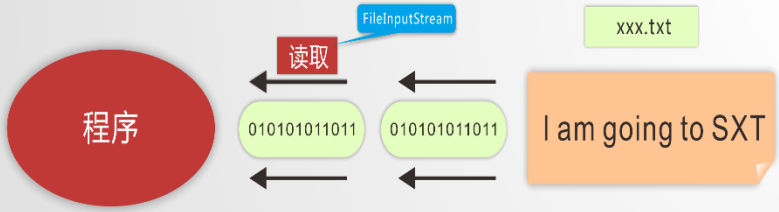
选择流

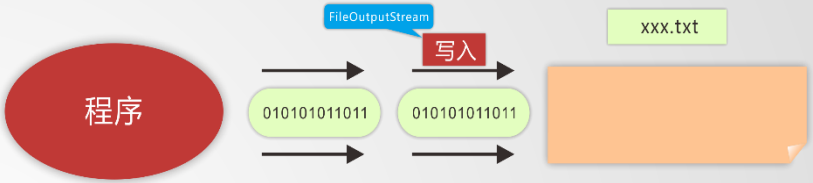
操作

释放

FileInputStream&FileOutputStream

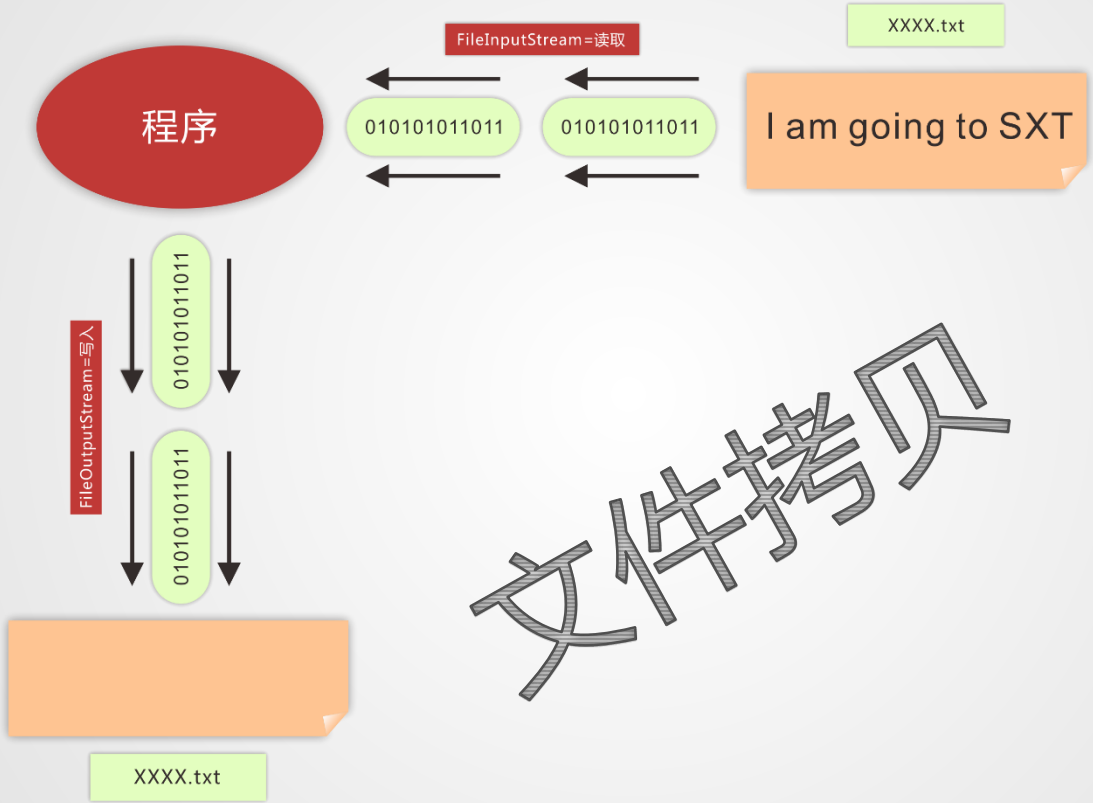
- **FileInputStream**:通过字节的方式读取文件，适合读取所有类型的文件(图像、视频等)，全字符请考虑FileReader





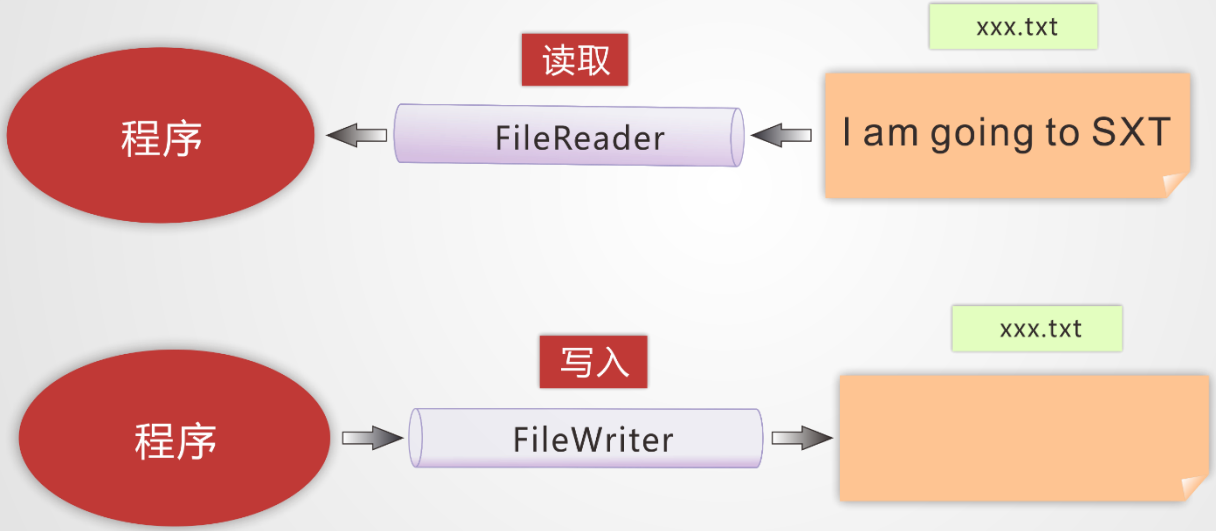
- **FileOutputStream**:通过字节的方式写出或追加数据到文件，适合所有类型的文件(图像、视频等),全字符请考虑FileWriter

拷贝


FileInputStream=读取
XXXX.txt
程序
010101011011
010101011011
I am going to SXT
FileOutputStream=写入
010101011011
010101011011
XXXX.txt
文件拷贝

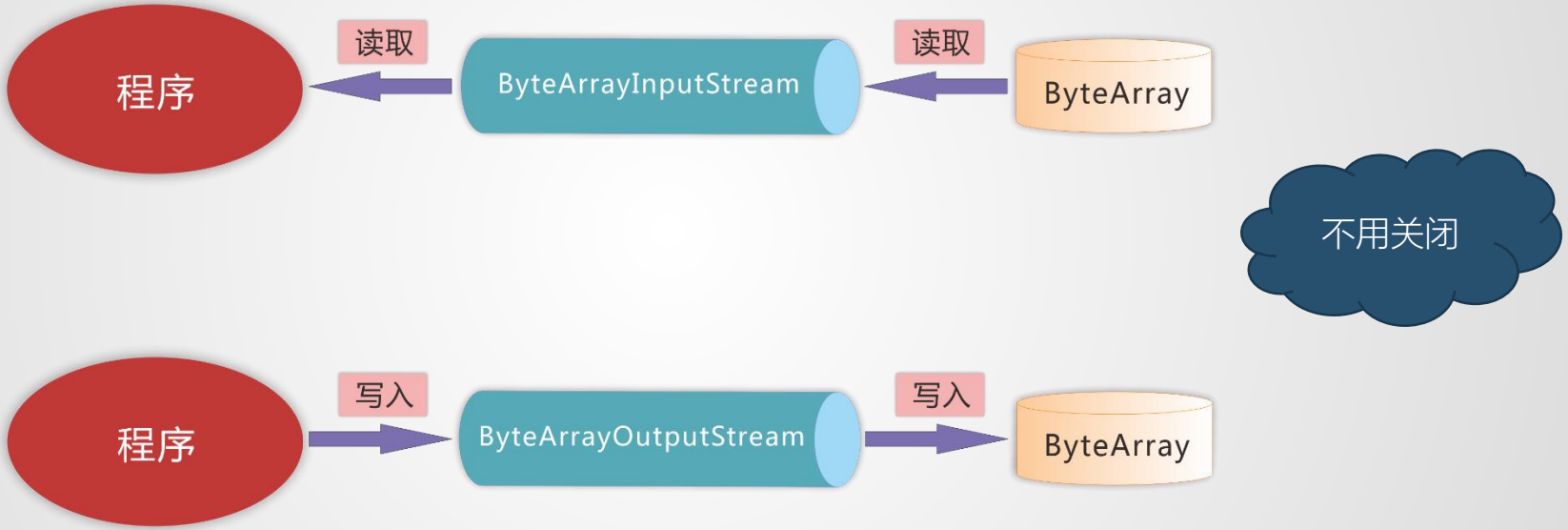
FileReader&FileWriter

- FileReader :通过字符的方式读取文件，仅适合字符文件
- FileWriter :通过字节的方式写出或追加数据到文件中，仅适合字符文件

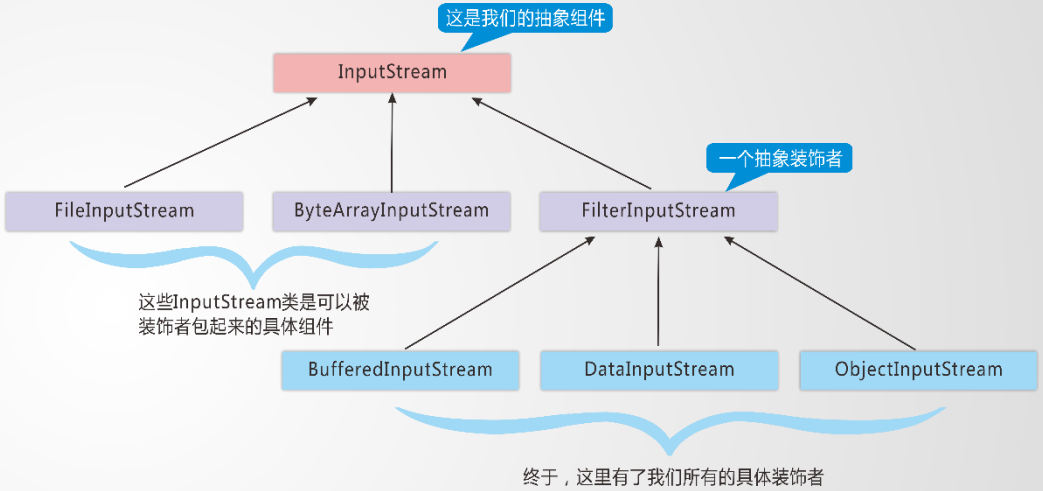


ByteArrayInputStream&ByteArrayOutputStream

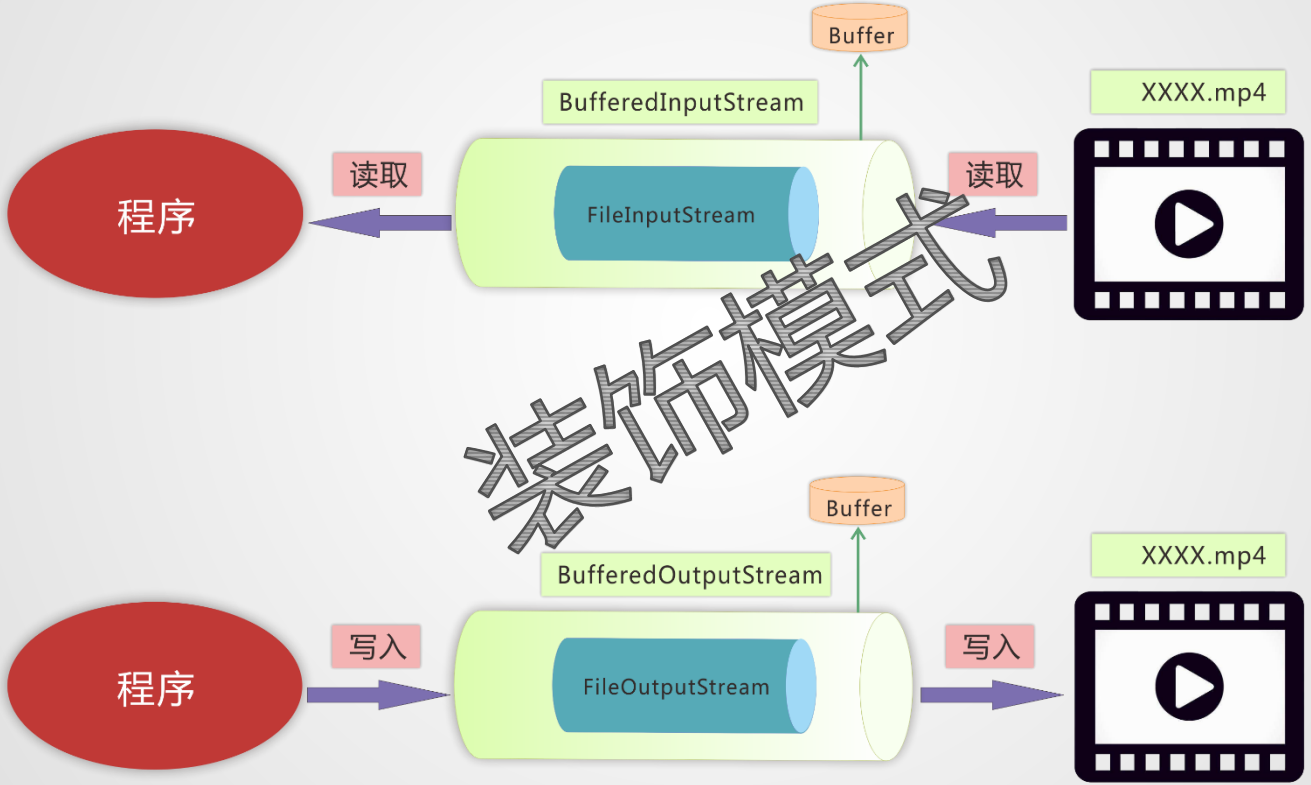
字节数组流



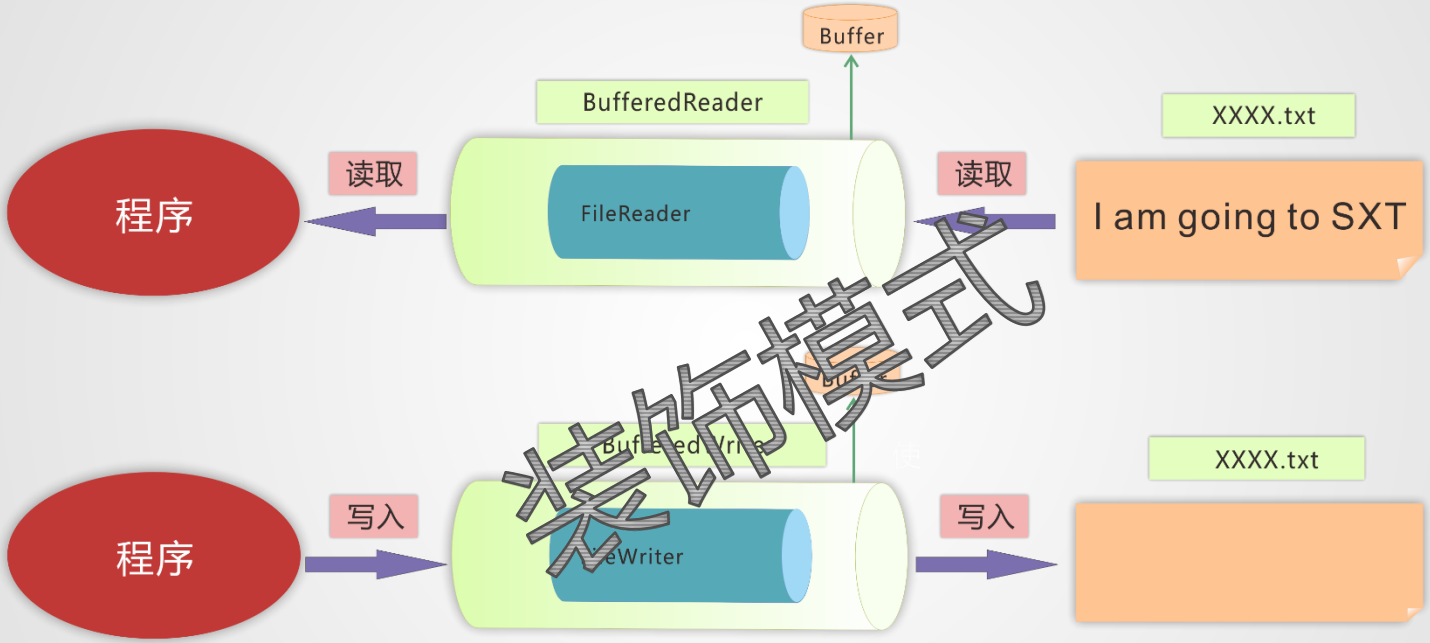
装饰器模式


这是我们的抽象组件
InputStream
一个抽象装饰者
FileInputStream
ByteArrayInputStream
FilterInputStream
这些InputStream类是可以被
装饰者包起来的具体组件
BufferedInputStream
DataInputStream
ObjectInputStream
终于，这里有了我们所有的具体装饰者

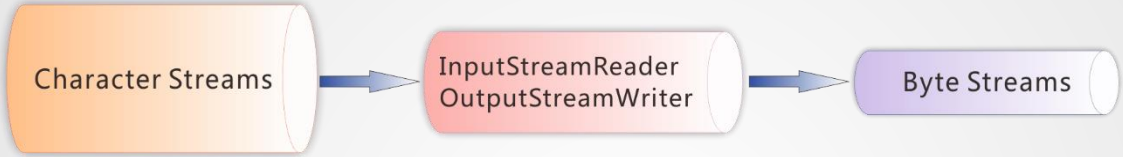
BufferedInputStream&BufferedOutputStream



BufferedReader&BufferedWriter

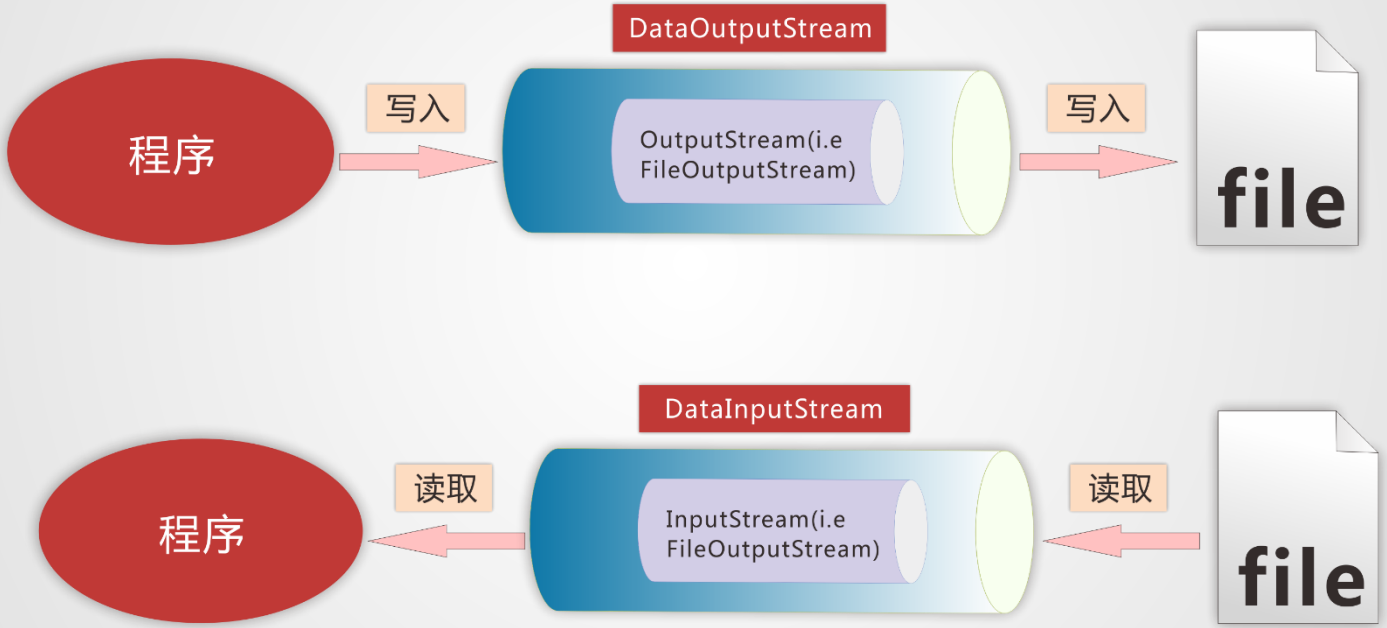


InputStreamReader&OutputStreamWriter

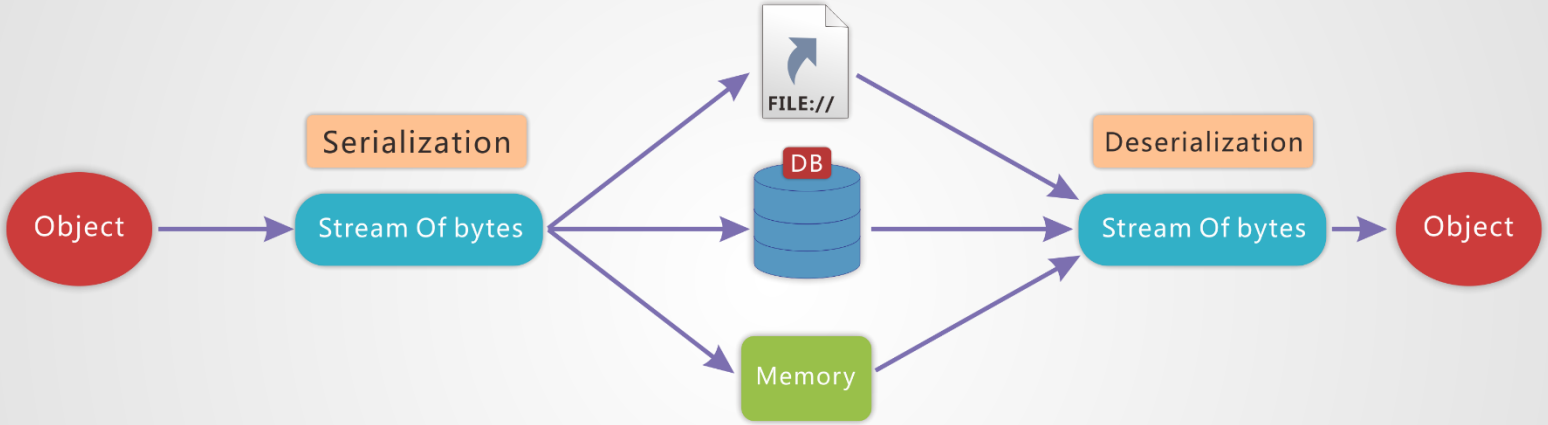


InputStreamReader/OutputStreamWriter： 是字节流与字符流之间的桥梁，能将字节流转换为字符流，并且能为字节流指定字符集，可处理一个个的字符

DataInputStream&DataOutputStream

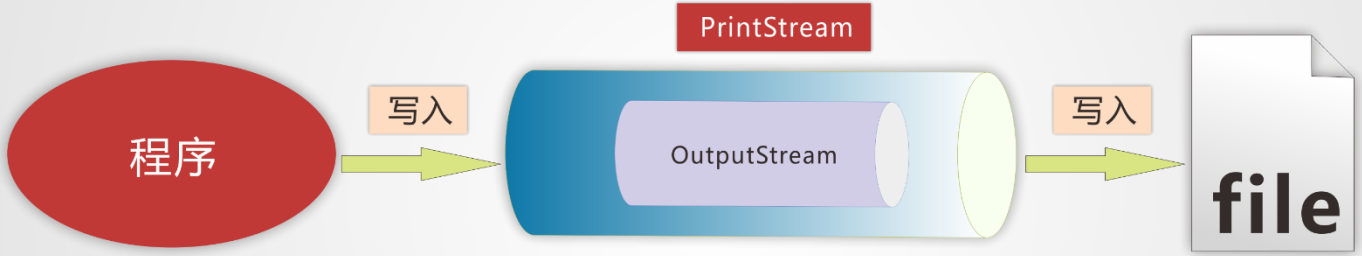


ObjectInputStream&ObjectOutputStream



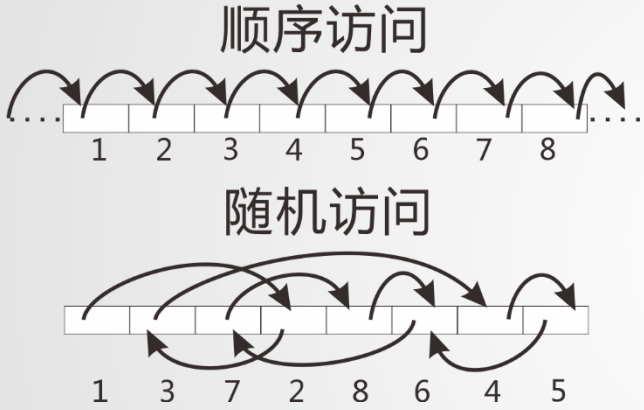
PrintStream

打印流



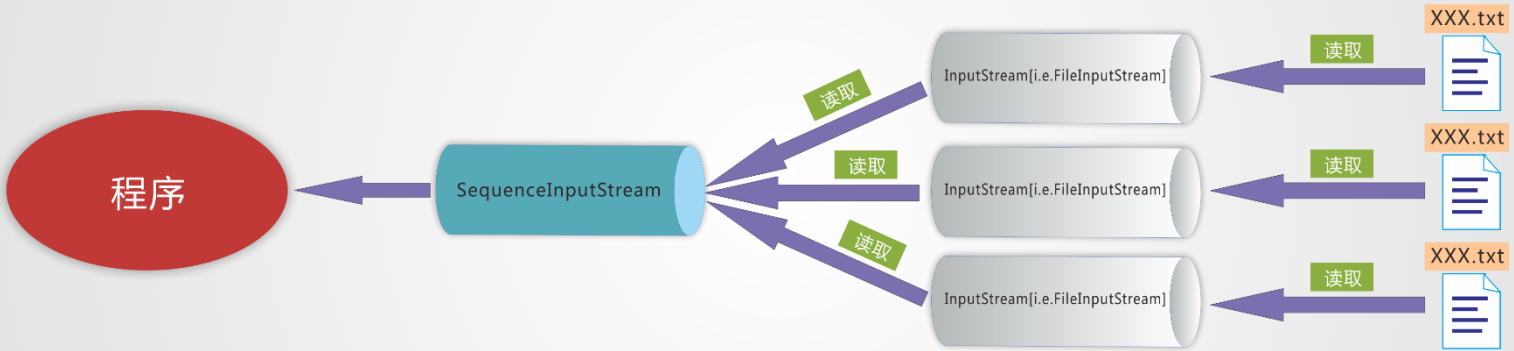
RandomAccessFile

随机流



SequenceInputStream

合并流



CommonsIO

FileUtils使用

PART Four

FileUtils

General file manipulation utilities.Facilities are provided in the following areas:

- writing to a file
- reading from a file
- make a directory including parent directories
- copying files and directories
- deleting files and directories
- converting to and from a URL
- listing files and directories by filter and extension
- comparing file content
- file last changed date
- calculating a checksum

commons IO™

感谢您的支持与信任

THANK YOU FOR WATCHING