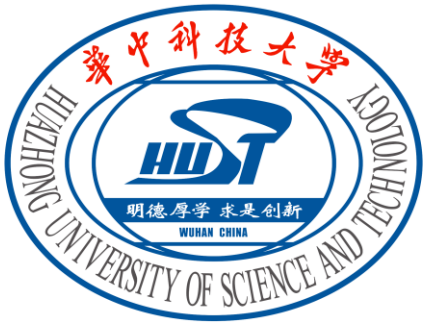
基于Java的面向对象程序设计

陈维亚

weiya_chen@hust.edu.cn

华中科技大学软件学院

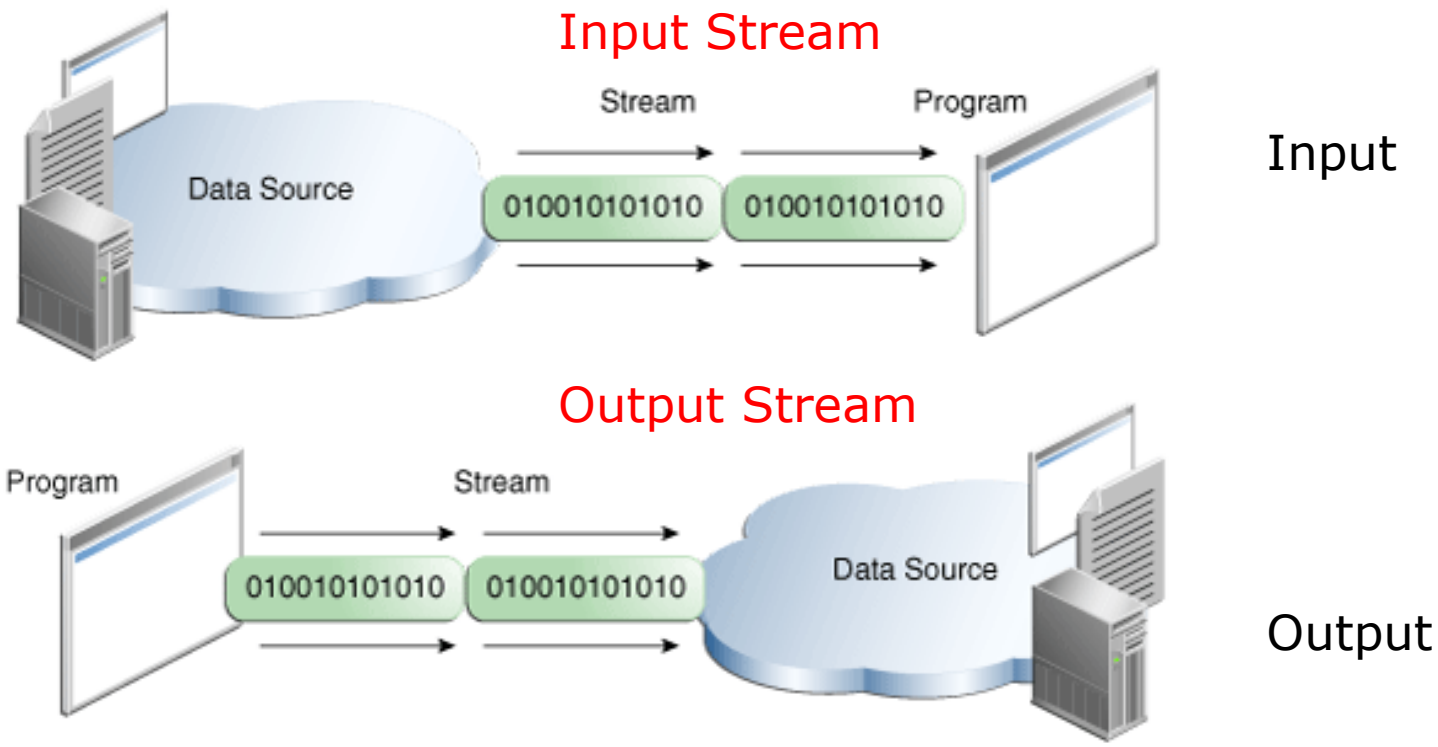
第18讲：Java IO



目录

1. IO Stream

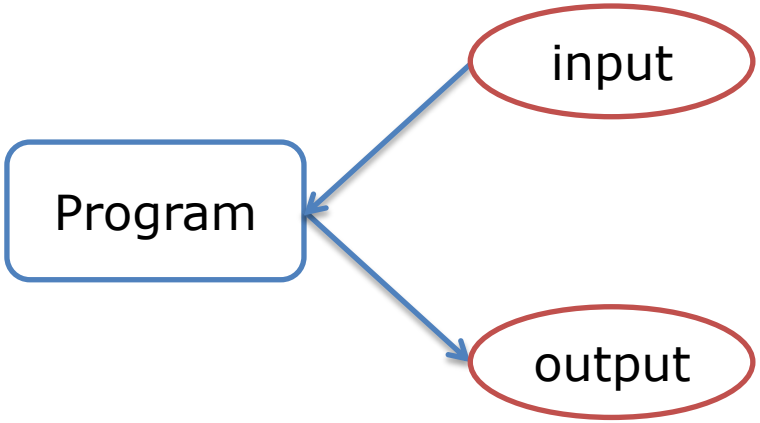
- IO - Input/Output



Stream：一个数据的序列。

1. IO Stream

❑ 流 Stream



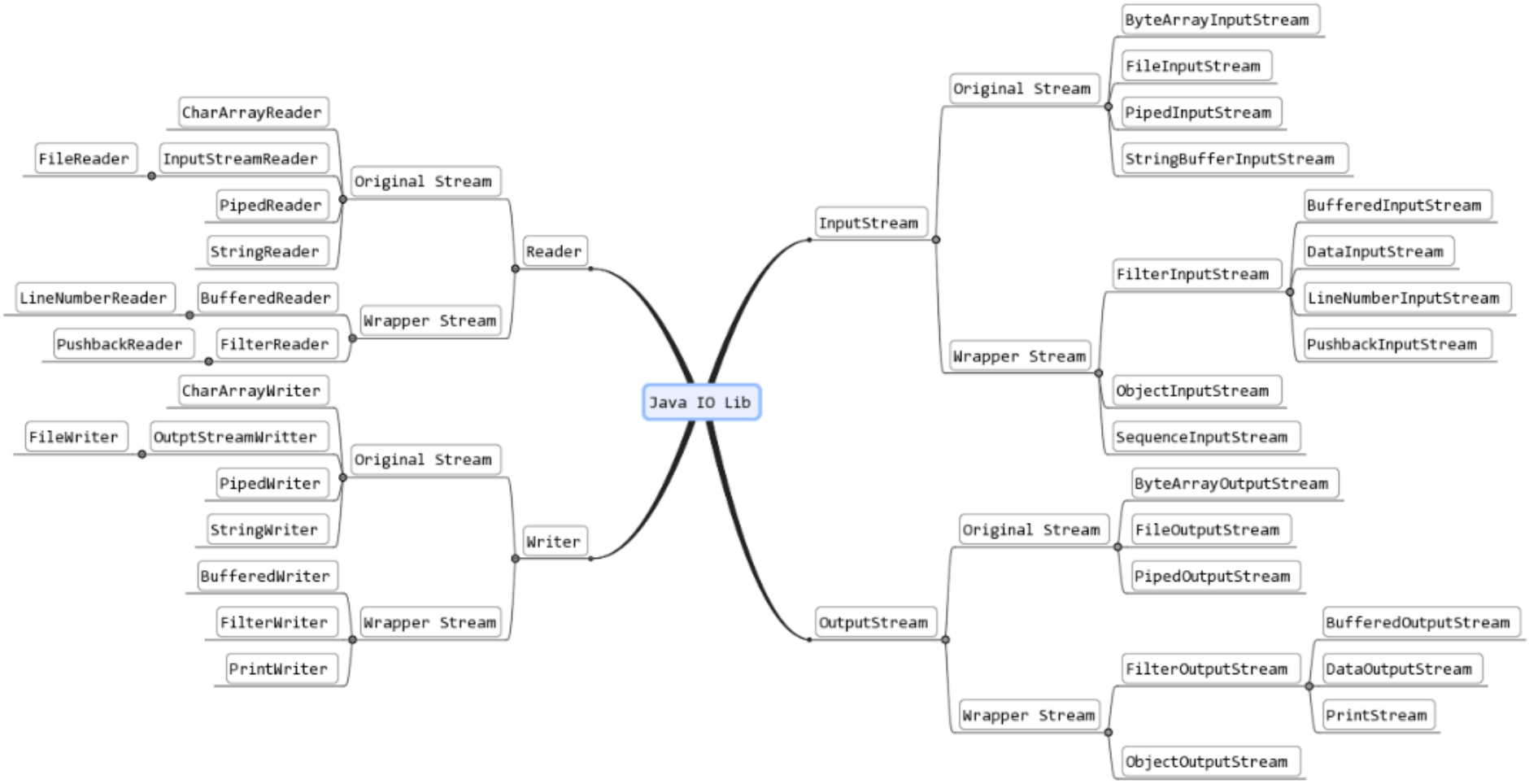
Stream代表一个数据的序列。

java.io包中的流支持多种数据类型，比如字节、基本数据类型、字符和对象。

java.io包几乎包含了所有操作输入、输出需要的类。所有这些Stream类代表了输入源和输出目标 。

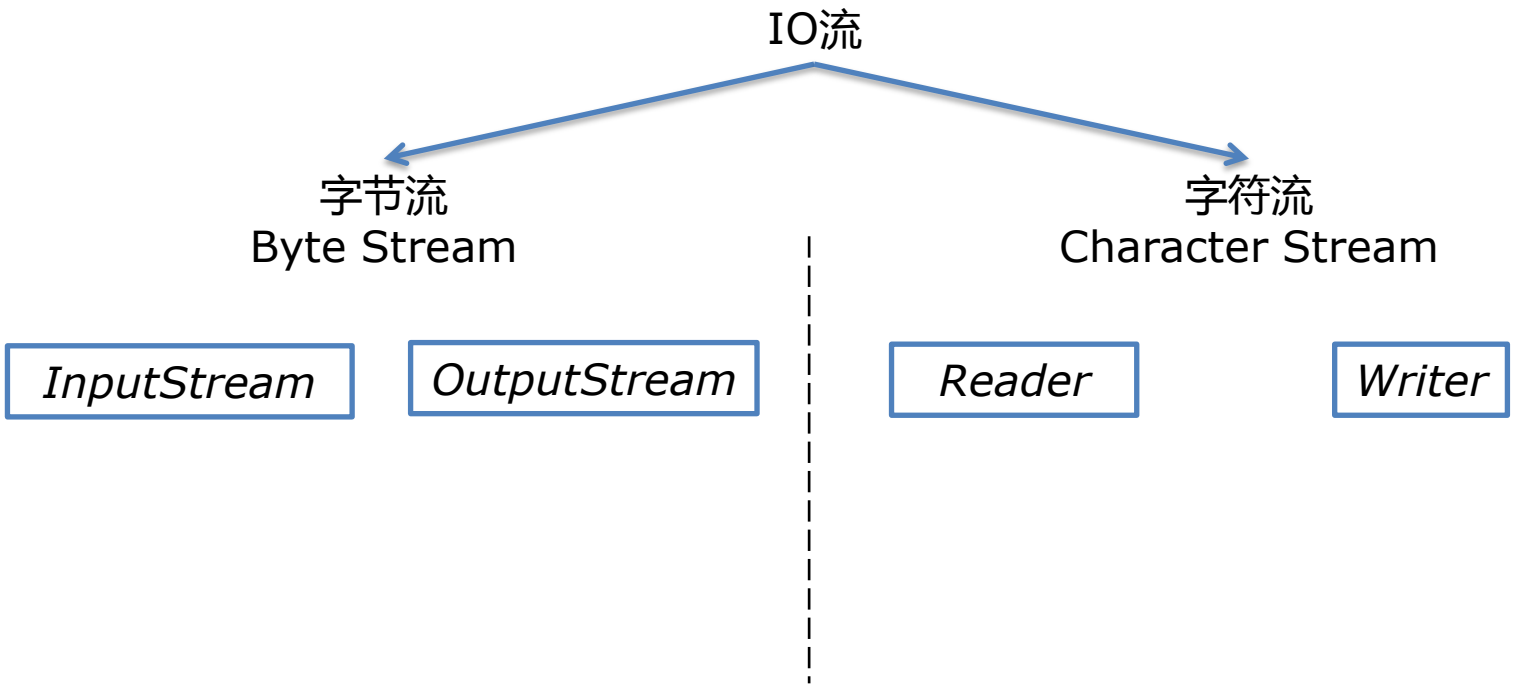
1. IO Stream

- java.io



1. IO Stream

- java.io



2. 控制台IO

❑ 标准流 Standard Stream

java.io.InputStream
　　java.io.FilterInputStream
　　　　java.io.BufferedInputStream

System.in

java.io.OutputStream
　　java.io.FilterOutputStream
　　　　java.io.PrintStream

System.out
System.err

输入字节

```
byte[] b = new byte[5];
int n = System.in.read(b);
```

输出字节

```
int c = 'A';
System.out.write(c);
System.out.write('\n');
```

2. 控制台IO

❑ 标准流 Standard Stream

java.io.InputStream
　　java.io.FilterInputStream
　　　　java.io.BufferedInputStream

System.in

java.io.OutputStream
　　java.io.FilterOutputStream
　　　　java.io.PrintStream

System.out
System.err

输入字符

```
char[] c = new char[5];
InputStreamReader cin = new InputStreamReader(System.in);
int n = cin.read();
```

输出字符

```
char c = 'A';
String s = “hello”;
System.out.print(c);
System.out.println(s);
```

2. 控制台IO

❑ 控制台 java.io.Console

输入字符

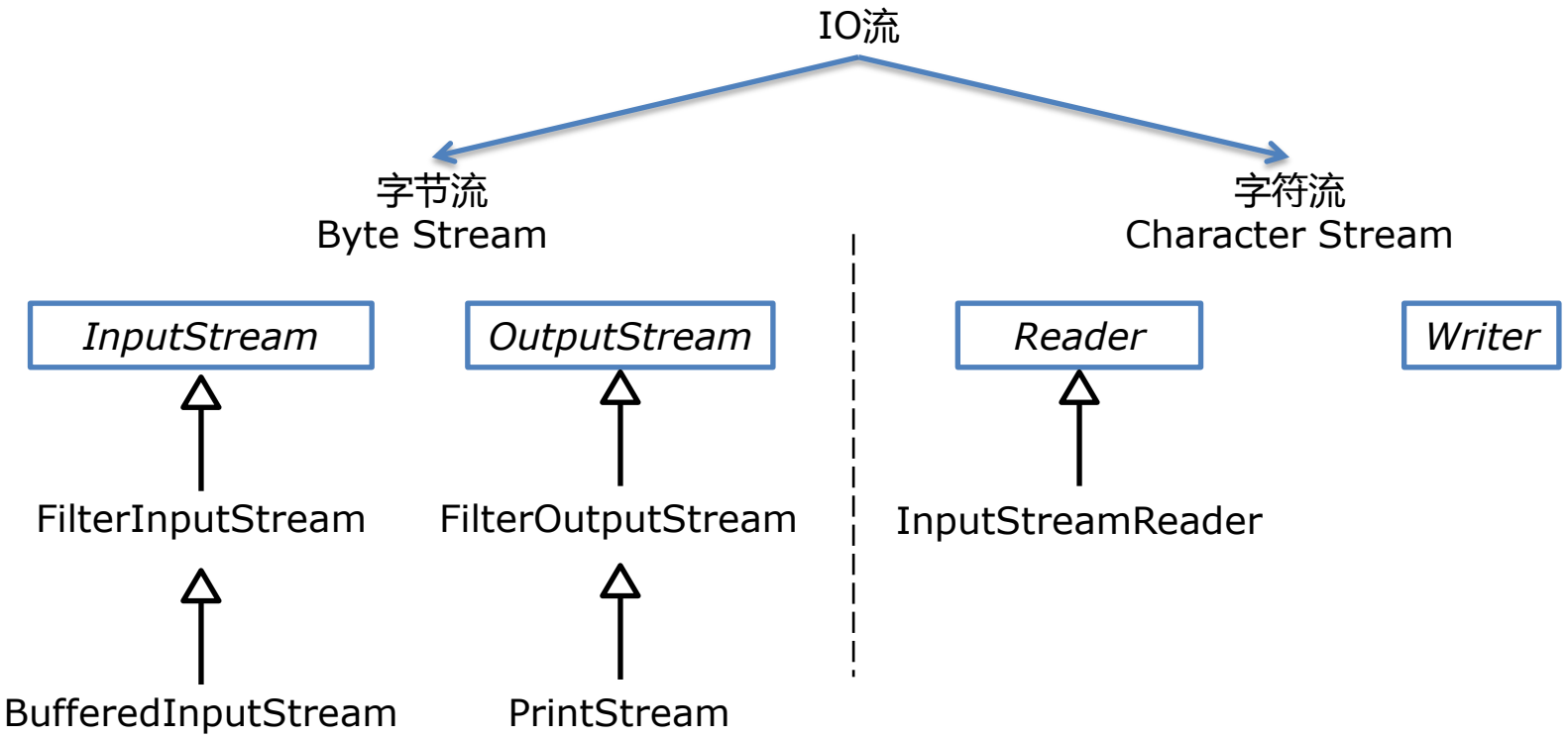
```
Console con = System.console();
if (con == null) {
    System.err.println("No console.");
    System.exit(1);
}

String login = con.readLine("Enter your login: ");
char [] password = con.readPassword("Enter your password: ");
```

输出字符

```
con.printf(login);
con.printf(oldPassword.toString());
```

2. 控制台IO

- java.io



3. 文件IO

❑ 读写文件

以下3种操作方式：

A - 按字节读写文件内容： FileInputStream / FileOutputStream

B - 按字符读写文件内容：FileReader / FileWriter

C - 随机读写文件内容：RandomAccessFile

3. 文件IO

❑ File 类

File用来代表文件系统中的文件，构造方法如下：

- File(File parent, String filename)
- File(String filename)
- File(String parent, String filename)

```
File file1 = new File("Lena.bmp");

File file2 = new File("/usr/local/bin");
```

常见成员方法：

boolean createNewFile()
boolean isDirectory()
String getName()

3. 文件IO

❑ 按字节读写

FileInputStream

read()方法

FileOutputStream

write()方法

```
byte[] message = "You know nothing!".getBytes();
try{
    File file = new File("test.txt");
    FileOutputStream out = new FileOutputStream(file);
    out.write(message);
    out.close();
} catch(IOException e){
    e.printStackTrace();
}
```

3. 文件IO

❑ 按字节读写

FileInputStream	FileOutputStream
read()方法	write()方法

```
byte[] text = new byte[20];
try{
    File file = new File("test.txt");
    FileInputStream in = new FileInputStream(file);
    in.read(text);
    System.out.println(new String(text));
    in.close();
} catch(IOException e){
    e.printStackTrace();
}
```

3. 文件IO

❑ 按字符读写

FileReader

read()方法

FileWriter

write()方法

```
char[] message = “Are you OK?”.toCharArray();
try{
    File file = new File(“test.txt”);
    FileWriter out = new FileWriter(file);
    out.write(message);
    out.close();
} catch(IOException e){
    e.printStackTrace();
}
```

3. 文件IO

❑ 按字符读写

FileReader

read()方法

FileWriter

write()方法

```
char[] text = new char[20];
try{
    File file = new File("test.txt");
    FileReader in = new FileReader(file);
    in.read(text);
    System.out.println(new String(text));
    in.close();
} catch(IOException e){
    e.printStackTrace();
}
```

3. 文件IO

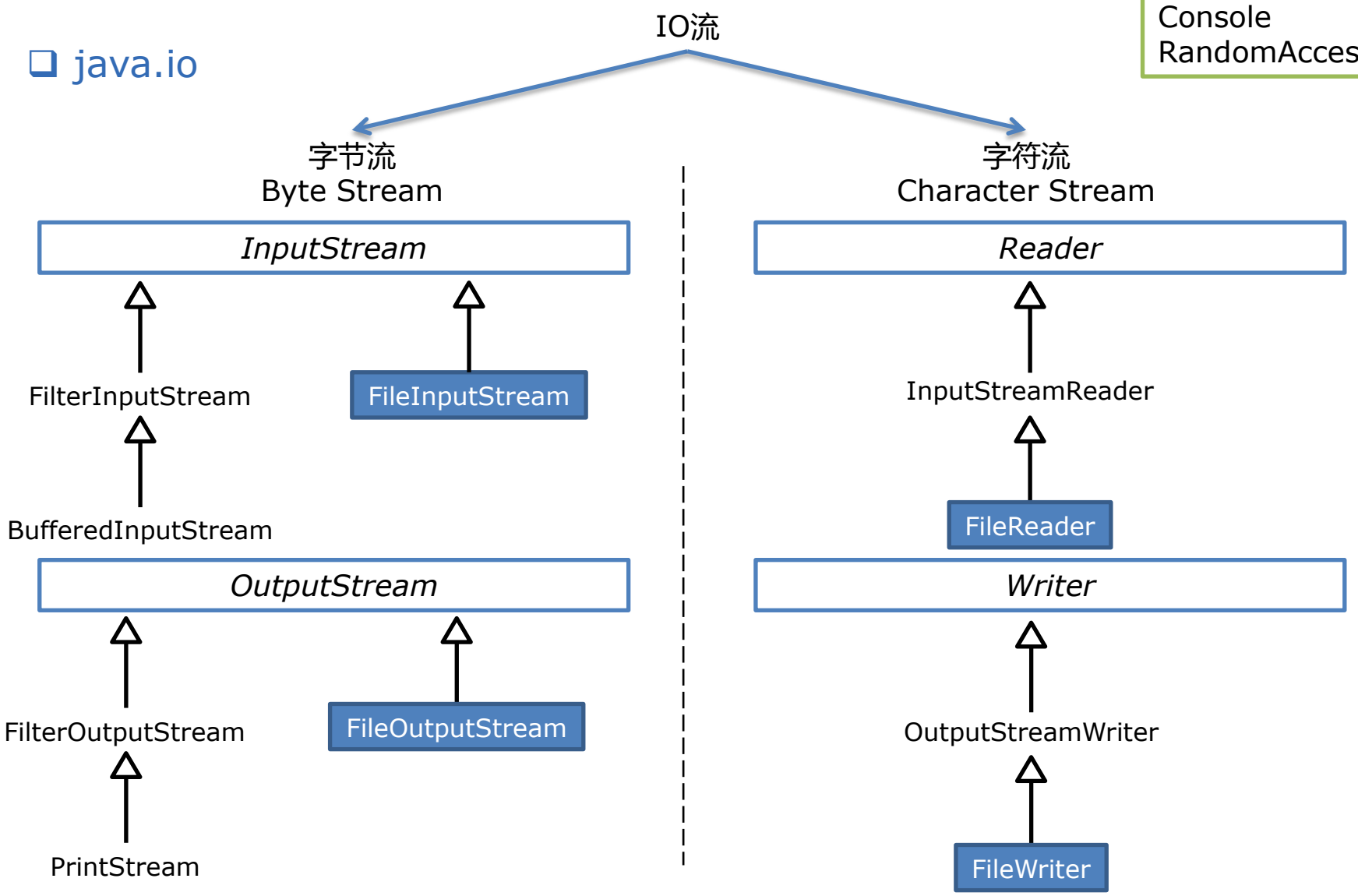
❑ 随机文件的读写

java.io.RandomAccessFile
- write()
- read()
- skipBytes(int n)
- seek(long pos)

```
byte[] text = new byte[20];
try{
    File file = new File("test.txt");
    RandomAccessFile rf = new RandomAccessFile(file, "rw");
    rf.read(text);
    rf.skipBytes(10);
    rf.read(text);
    rf.close();
} catch(IOException e){
    e.printStackTrace();
}
```

3. 文件IO

- java.io



Console
RandomAccessFile

3. 文件IO

❑ Buffer IO

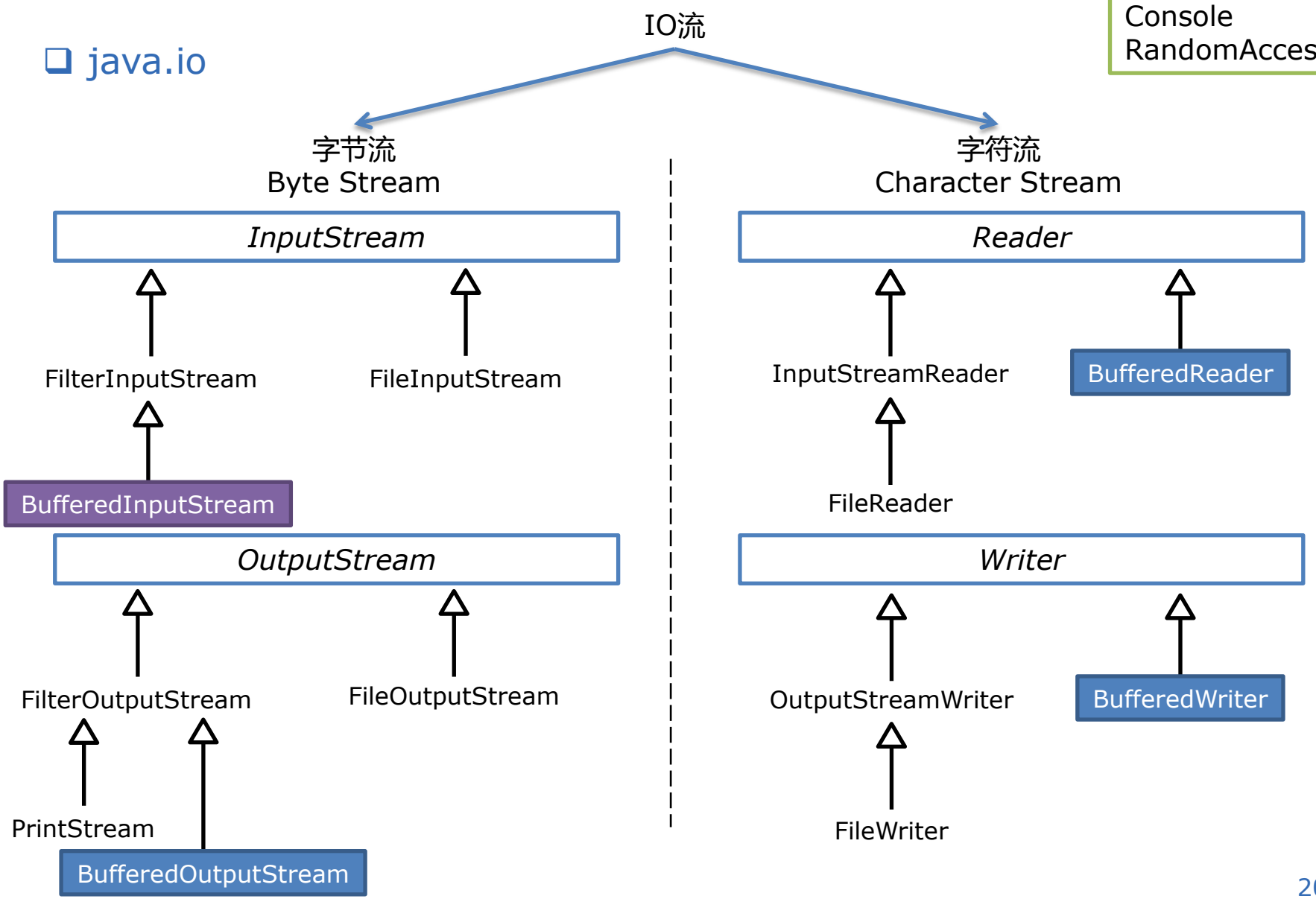
BufferedReader	BufferedInputStream
BufferedWriter	BufferedOutputStream

```
FileReader inputStream = new FileReader("xanadu.txt");
FileWriter outputStream = new FileWriter("output.txt");
```

```
BufferedReader inputStream = new BufferedReader(new FileReader("xanadu.txt"));
BufferedWriter outputStream = new BufferedWriter(new FileWriter("output.txt"));

String s;
while ((s = inputStream.readLine()) != null) {
    outputStream.write(s);
}
…
outputStream.flush();
```

3. 文件IO

- java.io



Console
RandomAccessFile

练习1

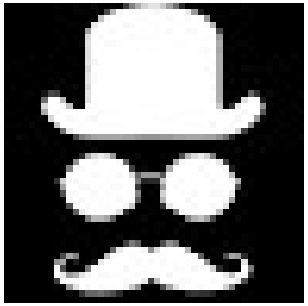
请指出完成以下IO操作分别应该使用哪个（哪类）IO类：

1）读取一个日志文件中的记录；
2）记录一个长度为200的整型数组的值；
3）读取一份配置文件中的用户参数（用户名、身份证等）；
4）从控制台读取用户输入的字符串；
5）在控制台显示当前的系统时间；

练习2

1）演示一个图像文件的读和写

2）演示一个文本文件的读和写



48x48

白日依山尽
黄河入海流

3. 文件IO

❑ java.util.Scanner类

public final class Scanner
extends Object
implements Iterator<String>, Closeable

- 可用于读取基本数据类型和String类型的流；
- 可将数据流用分隔符切割为token，默认的分隔符是空格；
- token可被不同的next方法转换为对应的数据类型；

```
Scanner sc = new Scanner(System.in);
int i = sc.nextInt();
```

```
Scanner sc = new Scanner(new File("myNumbers"));

while (sc.hasNextLong()) {
    long aLong = sc.nextLong();
}
```

练习3

Scanner

```
Scanner s = new Scanner(System.in);
String str1 = s.nextLine();
System.out.println(str1);
String str2 = s.next();
System.out.println(str2);
s.close();
```

>> How are you?
>> How are you?
>> Fine, thank you.
>> Fine,

4. 数据流与对象流

❑ Data Stream

接口	DataInput	DataOutput
实现类	DataInputStream	DataOutputStream

对字节流进行包装，更方便地读写基本数据类型：

```
try {
    while (true) {
        price = in.readDouble();
        unit = in.readInt();
        desc = in.readUTF();
        System.out.format("You ordered %d" + " units of %s at
$%.2f%n", unit, desc, price);
        total += unit * price; }
} catch (EOFException e) {…}
```

4. 数据流与对象流

❑ Object Stream

接口	ObjectInput	ObjectOutput
实现类	ObjectInputStream	ObjectOutputStream

输出对象：

```
ObjectOutputStream out = null;
try {
    out = new ObjectOutputStream(new BufferedOutputStream(new
FileOutputStream("invoicedata")));
    out.writeObject(Calendar.getInstance());
    for (int i = 0; i < prices.length; i ++) {
        out.writeObject(prices[i]);
        out.writeInt(units[i]);
        out.writeUTF(descs[i]);
    }
} finally { out.close(); }
```

❑ Object Stream

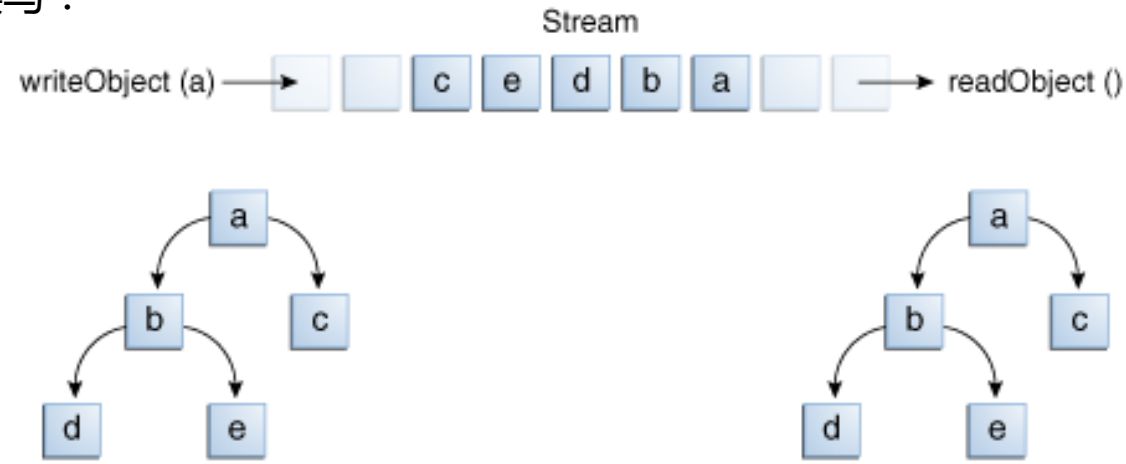
接口	ObjectInput	ObjectOutput
实现类	ObjectInputStream	ObjectOutputStream

读入对象：

```
try {
    ObjectInputStream in = new ObjectInputStream(new
BufferedInputStream(new FileInputStream(dataFile)));
    while (true) {
        price = (BigDecimal) in.readObject();
        unit = in.readInt();
        desc = in.readUTF();
       System.out.format("You ordered %d units of %s at $%.2f%n",
unit, desc, price);
        total = total.add(price.multiply(new BigDecimal(unit)));
    }
} catch (EOFException e) {}
```

❑ Object Stream

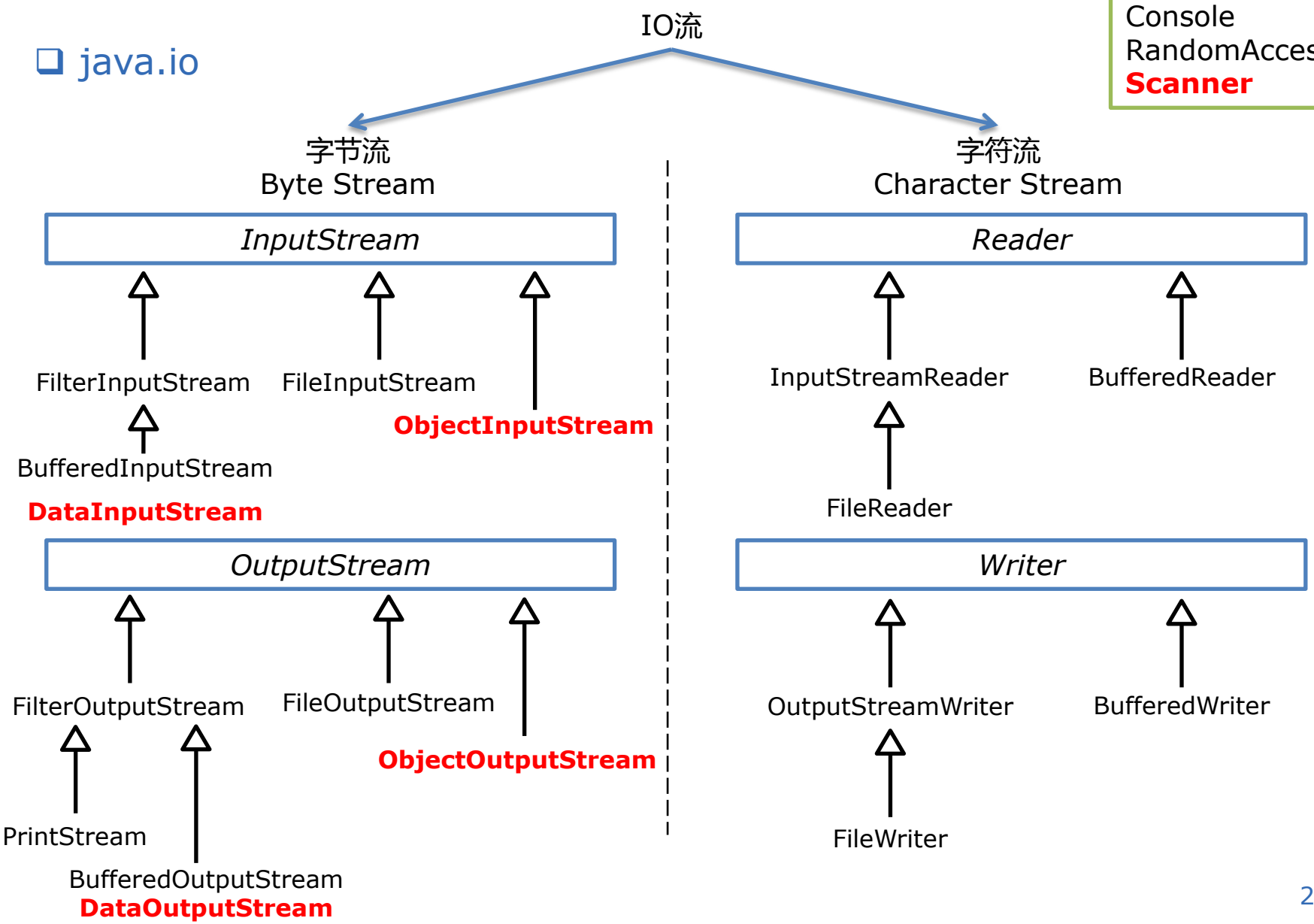
复杂对象的读写：



如果readObject()不能返回预期的对象类型，强制转换会抛出ClassNotFoundException

总结

- java.io



Console
RandomAccessFile
Scanner

下节预告

String