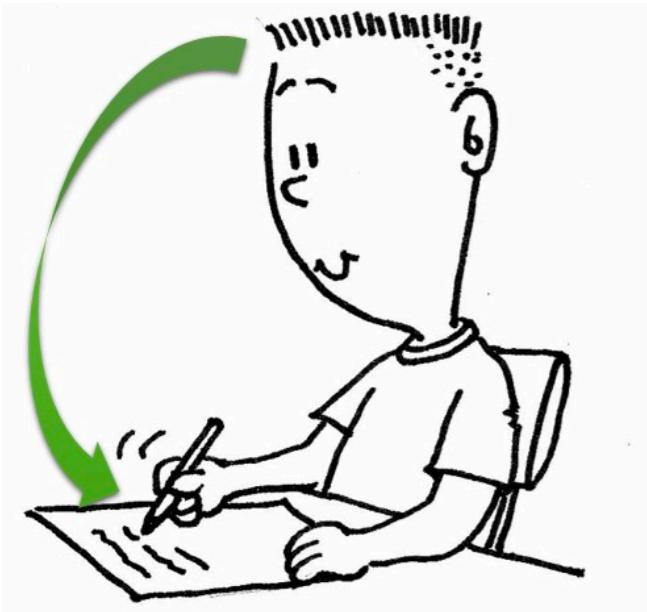
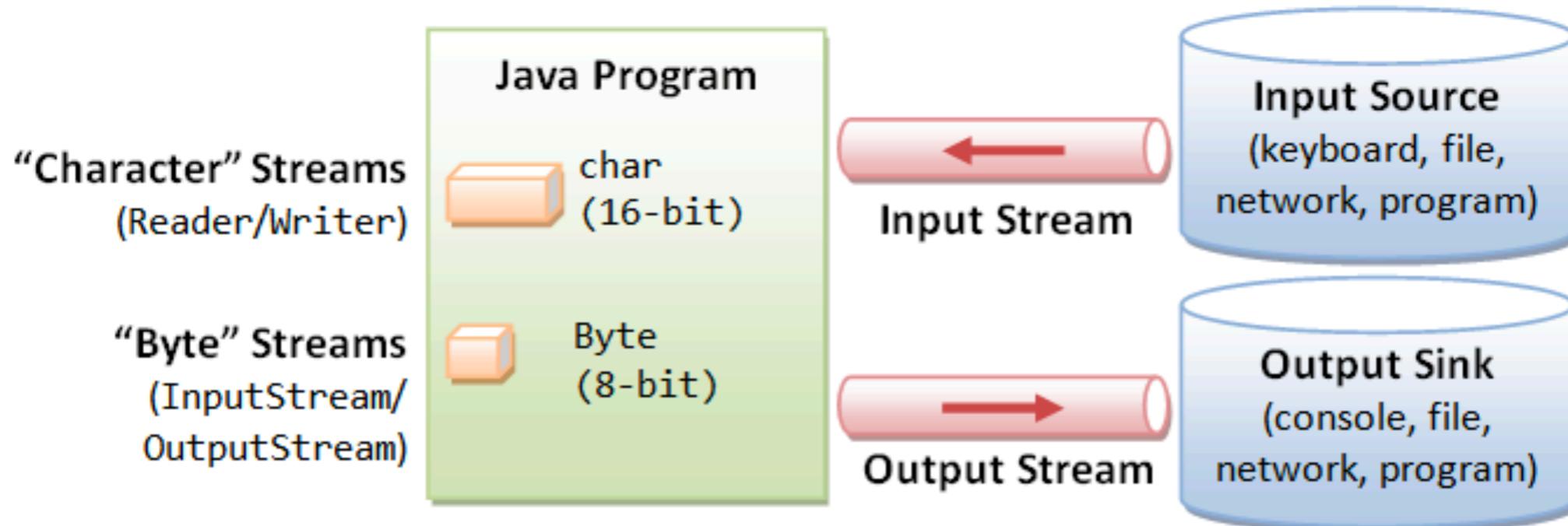
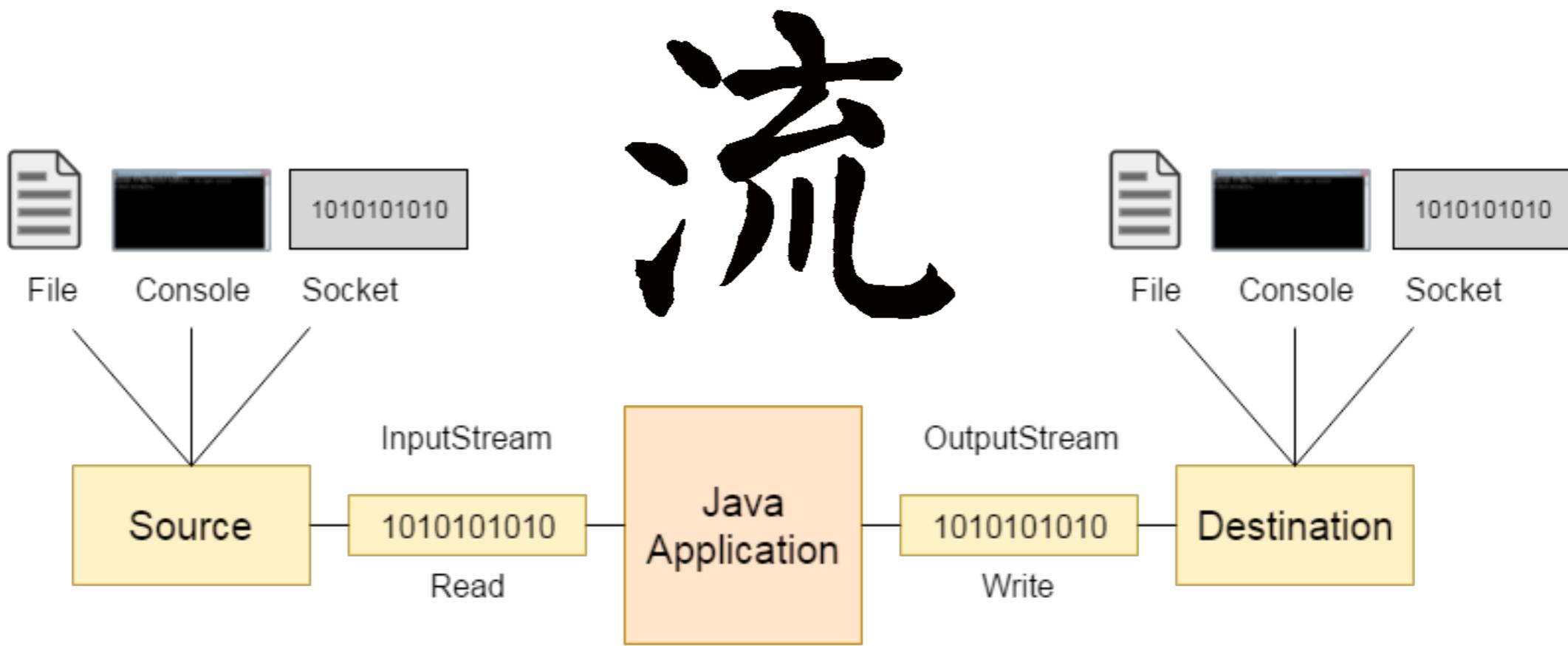


输入/输出

杨亮





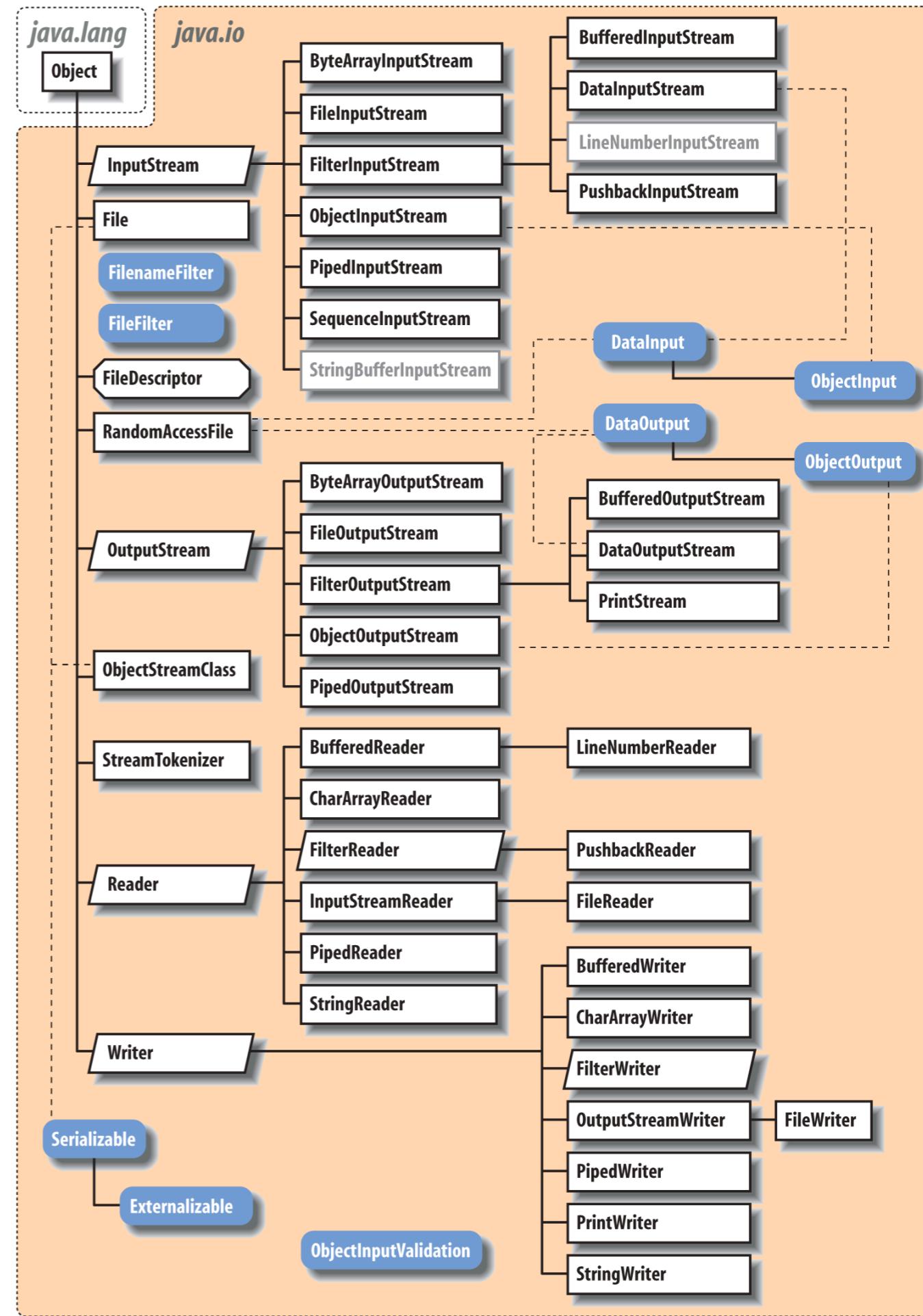
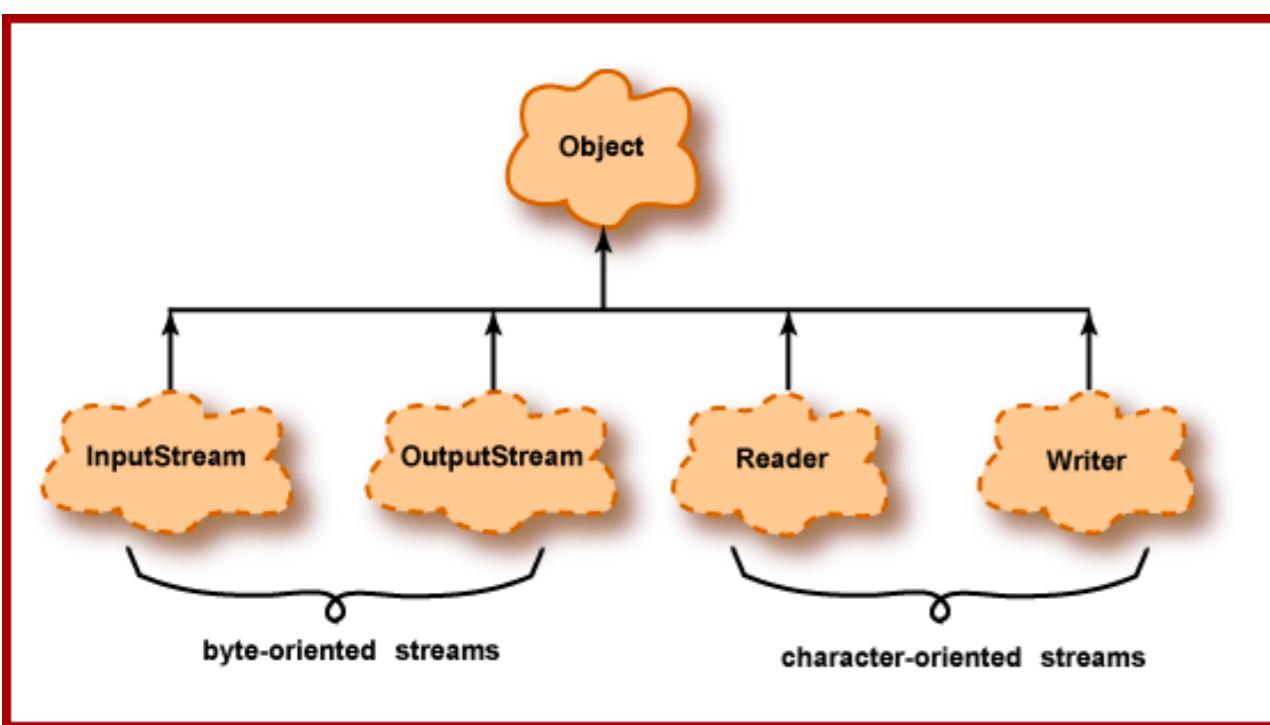
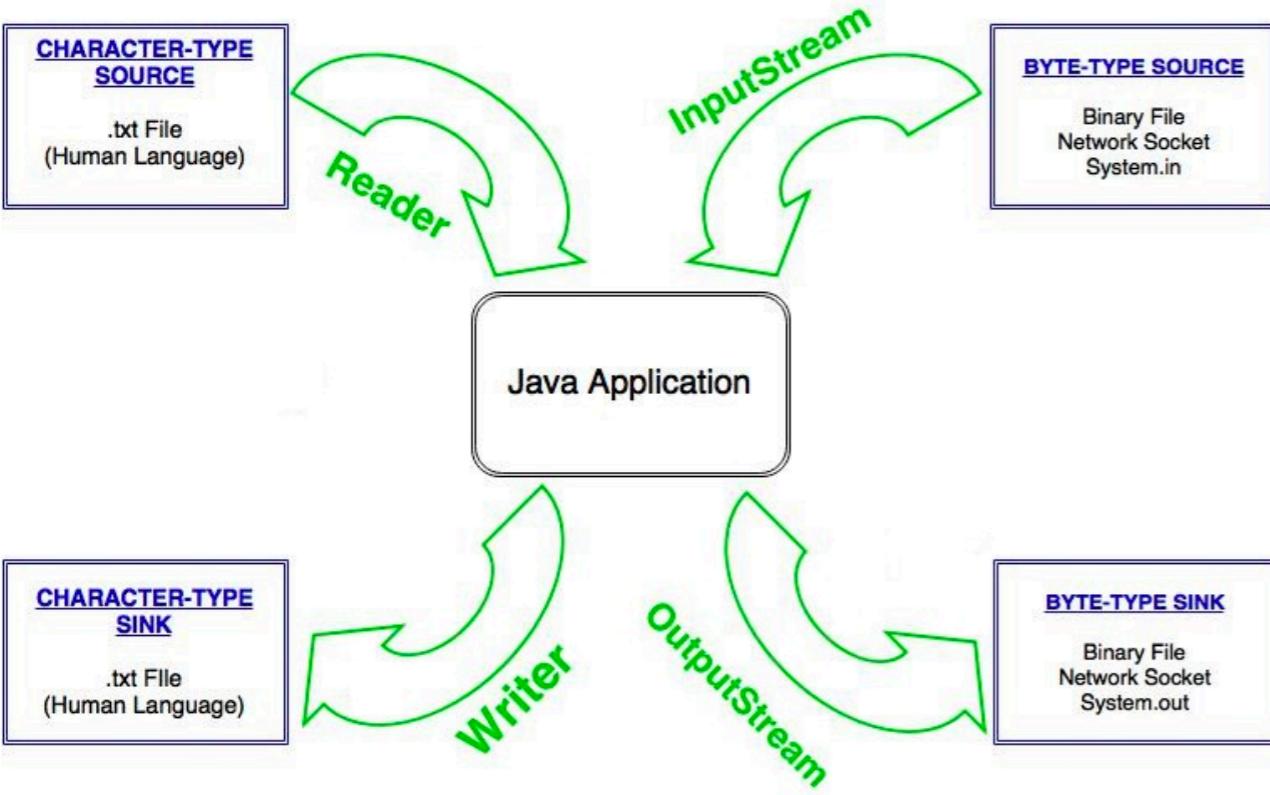
Internal Data Formats:

- Text (char): UCS-2
- int, float, double, etc.

External Data Formats:

- Text in various encodings (US-ASCII, ISO-8859-1, UCS-2, UTF-8, UTF-16, UTF-16BE, UTF16-LE, etc.)
- Binary (raw bytes)

流的分类



KEY

CLASS

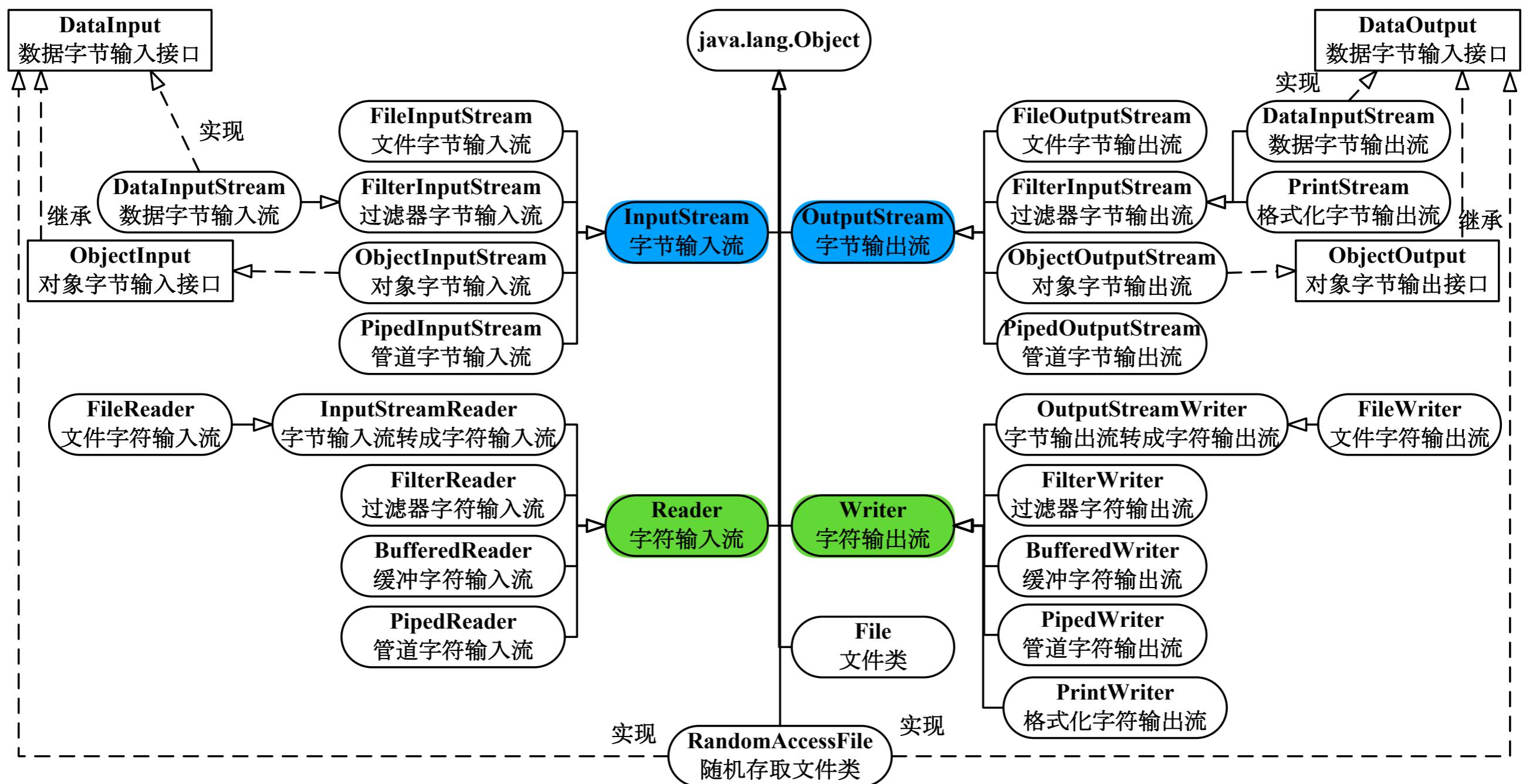
ABSTRACT CLASS

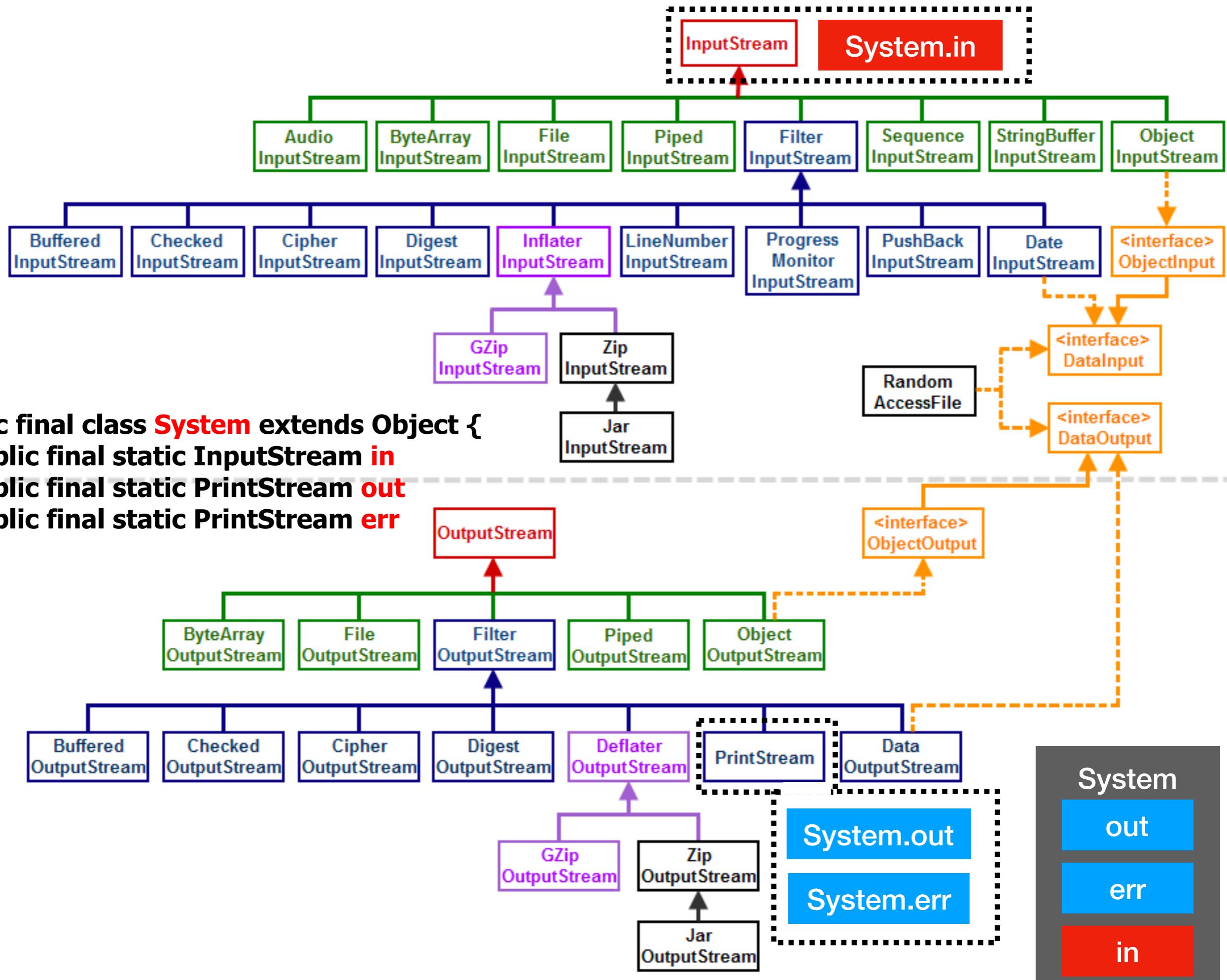
DEPRECATED CLASS

FINAL CLASS

INTERFACE

输入输出相关类图





抽象类

InputStream

```
public abstract int read() throws IOException  
public int read(byte[] bytes, int offset, int length) throws IOException  
public int read(byte[] bytes) throws IOException
```

OutputStream

```
public void abstract void write(int unsignedByte) throws IOException  
public void write(byte[] bytes, int offset, int length) throws IOException  
public void write(byte[] bytes) throws IOException
```

Reader

```
public abstract int read() throws IOException  
public int read(char[] chars, int offset, int length) throws IOException  
public int read(char[] chars) throws IOException
```

Writer

```
public void abstract void write(int aChar) throws IOException  
public void write(char[] chars, int offset, int length) throws IOException  
public void write(char[] chars) throws IOException
```

文件流

FileInputStream

FileReader

```
public class FileInputStream extends InputStream {  
    public FileInputStream(String filename) throws FileNotFoundException  
    public FileInputStream(File file) throws FileNotFoundException  
}
```

FileOutputStream

FileWriter

```
public class FileOutputStream extends OutputStream{  
    public FileOutputStream(String filename) throws FileNotFoundException  
    public FileOutputStream(File file) throws FileNotFoundException  
    public FileOutputStream(String filename, boolean append) throws  
        FileNotFoundException  
}
```

```
import java.io.*;

public class fileStreamTest {
    public static void main(String args[]) {
        try {
            byte bWrite[] = { 11, 21, 3, 40, 5 };
            OutputStream os = new FileOutputStream("test.txt");
            for (int x = 0; x < bWrite.length; x++) {
                os.write(bWrite[x]); // writes the bytes
            }
            os.close();

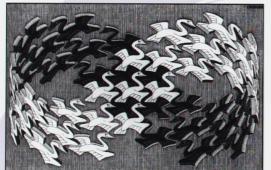
            InputStream is = new FileInputStream("test.txt");
            int size = is.available();

            for (int i = 0; i < size; i++) {
                System.out.print((char) is.read() + " ");
            }
            is.close();
        } catch (IOException e) {
            System.out.print("Exception");
        }
    }
}
```

Design Patterns

Elements of Reusable
Object-Oriented Software

Erich Gamma
Richard Helm
Ralph Johnson
John Vlissides



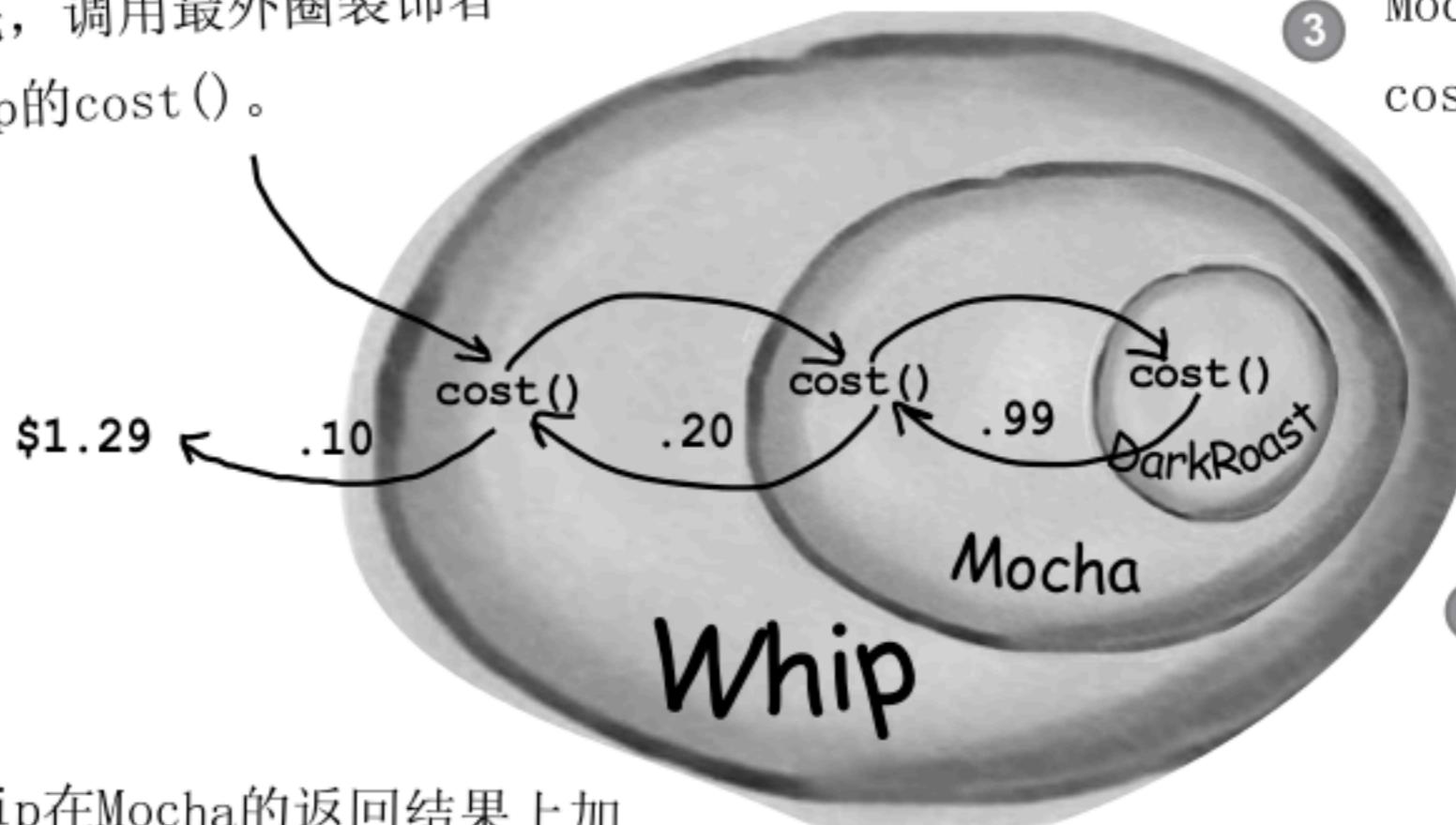
Cover art © 1994 M.C. Escher / Cordon Art - Baarn - Holland. All rights reserved.

Foreword by Grady Booch

* ADDISON-WESLEY PROFESSIONAL COMPUTING SERIES

装饰者模式

- ① 首先，调用最外圈装饰者
Whip的cost()。



- ⑥ Whip在Mocha的返回结果上加
上自己的价钱\$0.10，然后返回
最后结果\$1.29。

- ⑤ Mocha在DarkRoast的结果上，
加上自己的价钱\$0.20，返回新
的价钱\$1.19。

知道这是如何办
到的。

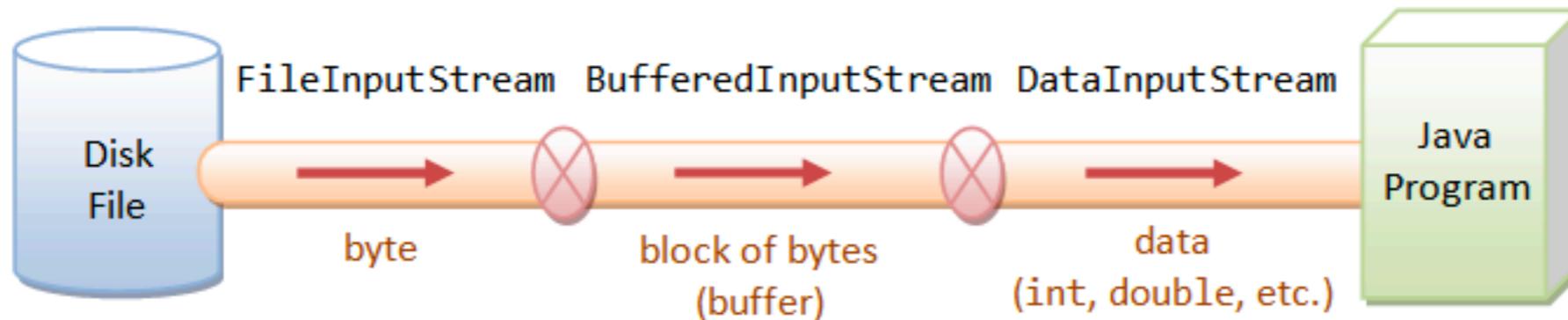
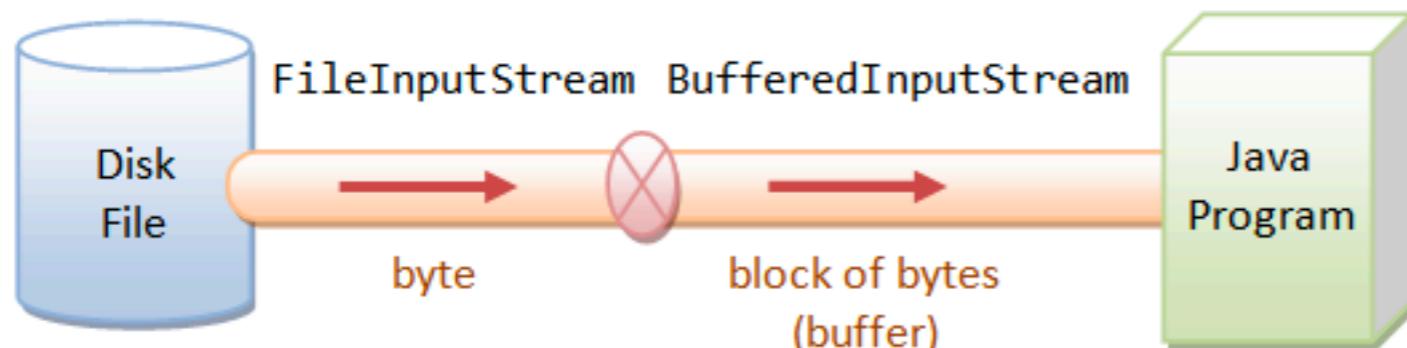
- ③ Mocha调用DarkRoast的
cost()。

- ④ DarkRoast返回它的价
钱\$0.99。

链式输入输出流

```
protected FilterInputStream(InputStream in) {  
    this.in = in;  
}
```

```
public int read() throws IOException {  
    return in.read();  
}
```

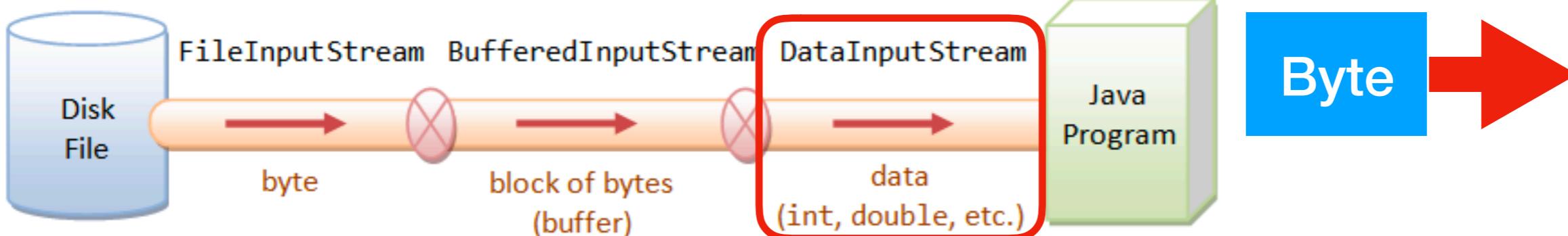
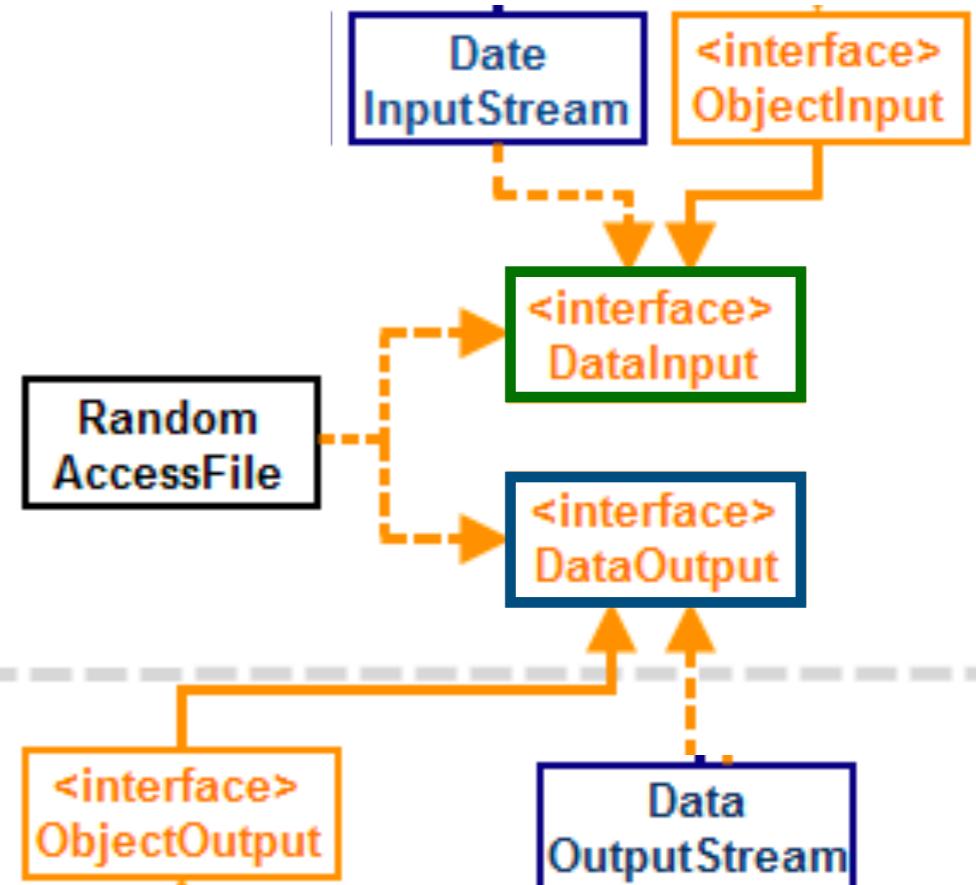


c FilterInputStream	
f in	InputStream
m FilterInputStream(InputStream)	
m read()	int
m read(byte[])	int
m read(byte[], int, int)	int
m skip(long)	long
m available()	int
m close()	void
m mark(int)	void
m reset()	void
m markSupported()	boolean

DataXXputStream

- public DataInputStream(InputStream in);
- readShort()、readByte()、readInt()、
- readLong()、readFloat()、readDouble()、
- readChar()、readBoolean()

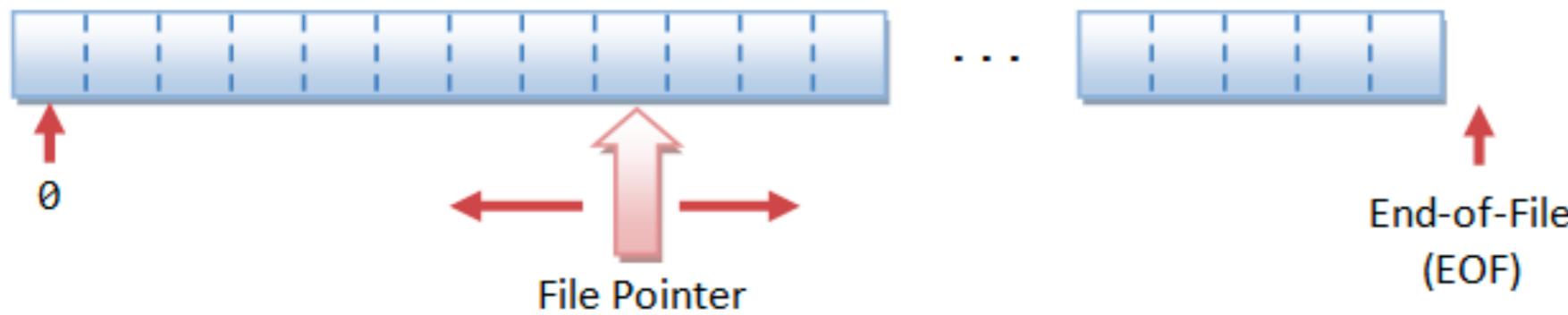
- public DataOutputStream(OutputStream out);
- writeShort()、writeByte()、writeInt()、
- writeLong()、writeFloat()、writeDouble()、
- writeChar()、writeBoolean()、writeChars()



Short
Int
Long
Float
Double
Char
Boolean

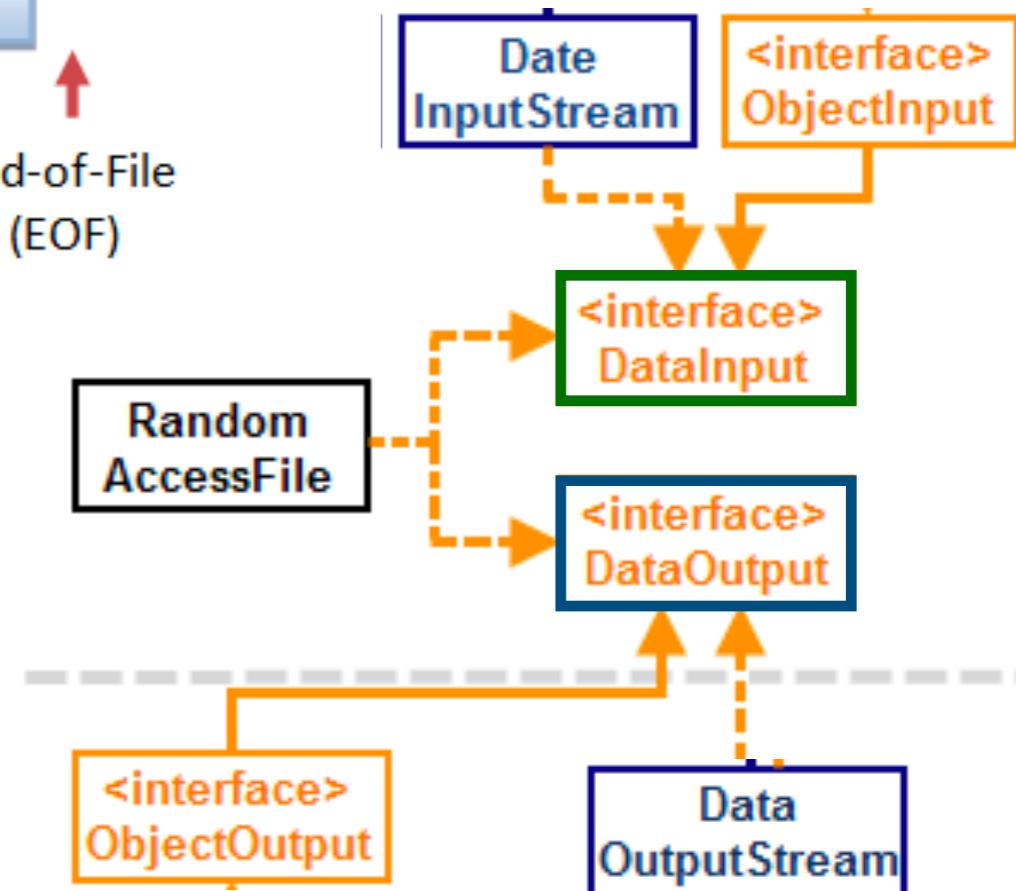
RandomAccessFile

```
public void seek(long pos) throws IOException;  
// Positions the file pointer for subsequent read/write operation.  
public int skipBytes(int numBytes) throws IOException;  
// Moves the file pointer forward by the specified number of bytes.  
public long getFilePointer() throws IOException;  
// Gets the position of the current file pointer, in bytes, from the beginning of  
the file.  
public long length() throws IOException;  
// Returns the length of this file.
```

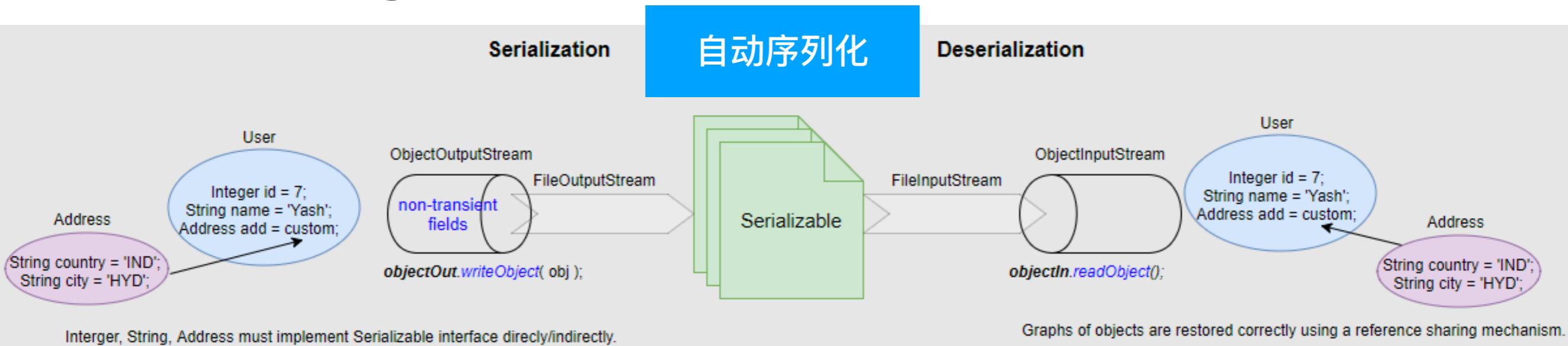


```
➤ public DataInputStream(InputStream in);  
➤ readShort( )、readByte( )、readInt( )、readLong( )、readFloat( )、  
readDouble( )、readChar( )、readBoolean( )
```

```
➤ public DataOutputStream(OutputStream out);  
➤ writeShort( )、writeByte( )、writeInt( )、writeLong( )、  
writeFloat( )、writeDouble()、writeChar( )、writeBoolean( )、  
writeChars()
```



ObjectOutputStream



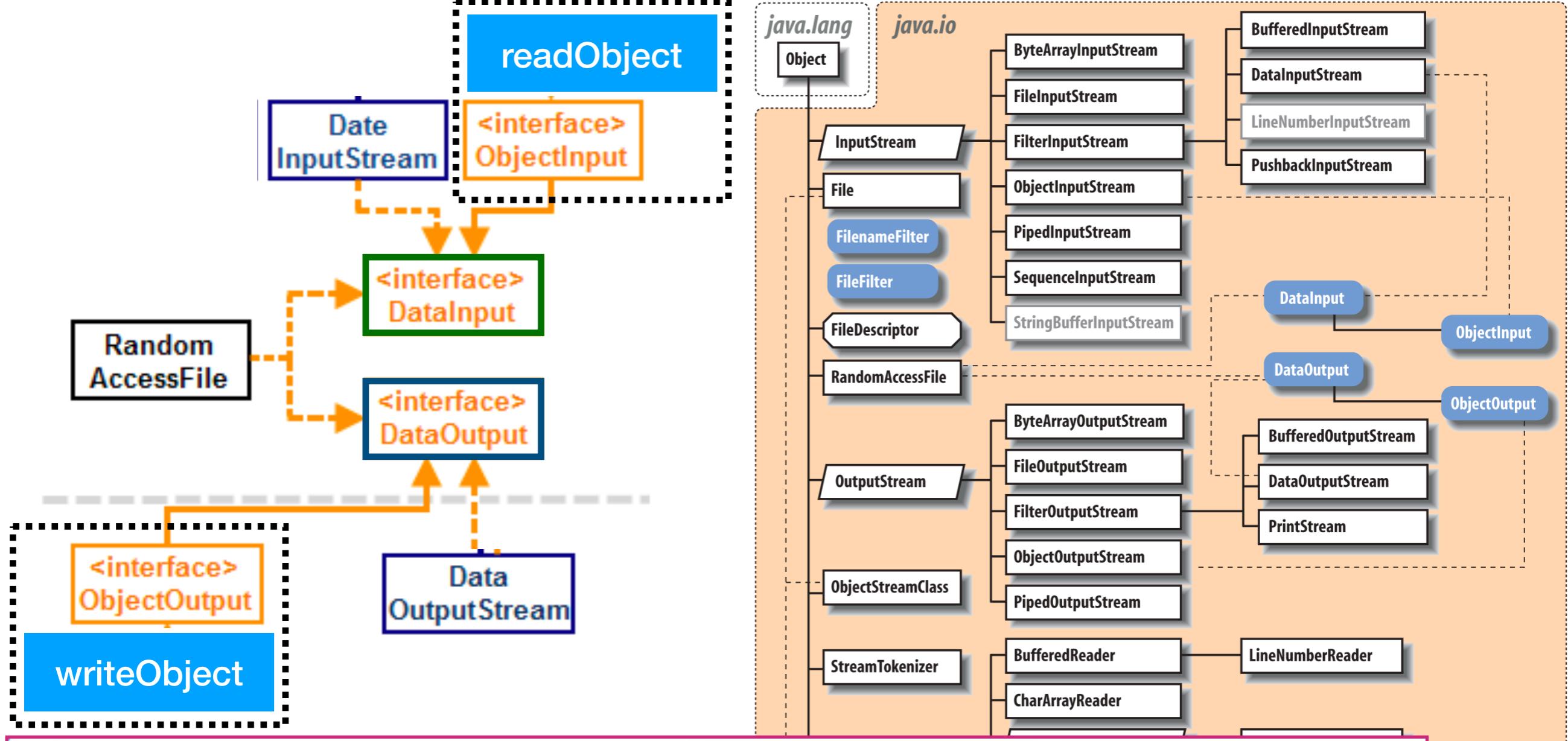
```
ObjectOutputStream out =
    new ObjectOutputStream(
        new BufferedOutputStream(
            new FileOutputStream("object.ser")));
out.writeObject("The current Date and Time is "); // write a String object
out.writeObject(new Date()); // write a Date object
out.flush();
out.close();
```

不序列化的属性
static
transient

public interface Serializable {};

```
public final Object readObject() throws IOException, ClassNotFoundException;
public final void writeObject(Object obj) throws IOException;
```

```
ObjectInputStream in =
    new ObjectInputStream(
        new BufferedInputStream(
            new FileInputStream("object.ser")));
String str = (String)in.readObject();
Date d = (Date)in.readObject(new Date()); // downcast
in.close();
```



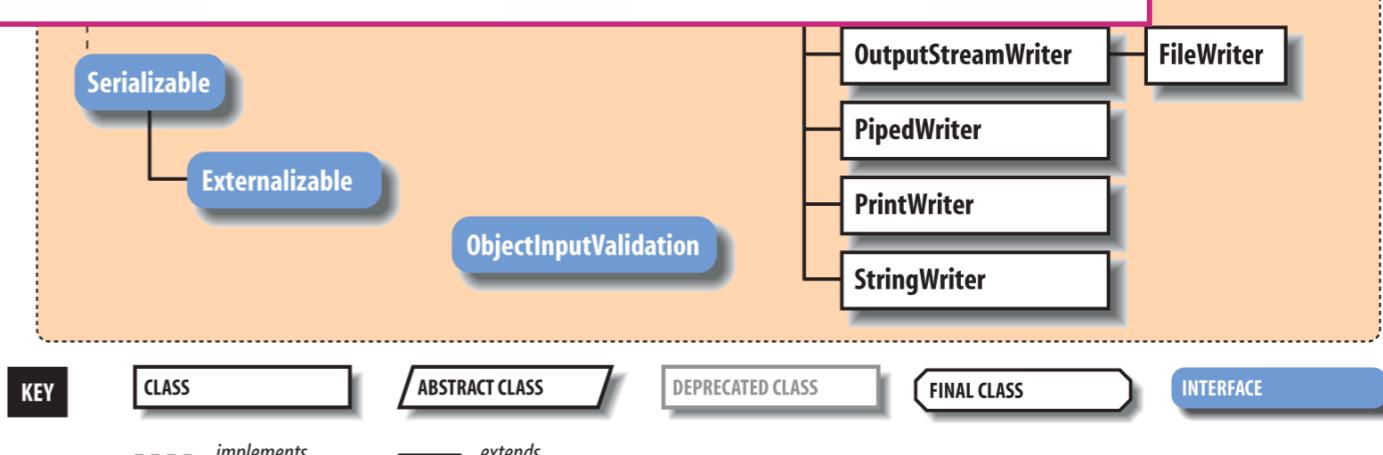
```

66 public interface Externalizable extends java.io.Serializable {
67     * The object implements the writeExternal method to save its contents.
68     void writeExternal(ObjectOutput out) throws IOException;
69
70     * The object implements the readExternal method to restore its.
71     void readExternal(ObjectInput in) throws IOException, ClassNotFoundException;
72 }

```

public interface Serializable {};

手动序列化



KEY

CLASS

ABSTRACT CLASS

DEPRECATED CLASS

FINAL CLASS

INTERFACE

--- implements

— extends

文件 File

```
public interface Serializable {};
```

```
public class File extends Object implements Comparable<File>{  
    public File(String pathname)  
    public File(String parent, String child)  
    public File(File parent, String child)  
}
```

```
interface Comparable<T> {  
    abstract int compareTo(T o);  
}
```

```
public boolean exists()          // Tests if this file/directory exists.  
public long length()            // Returns the length of this file.  
public boolean isDirectory()     // Tests if this instance is a directory.  
public boolean isFile()           // Tests if this instance is a file.  
public boolean canRead()         // Tests if this file is readable.  
public boolean canWrite()        // Tests if this file is writable.  
public boolean delete()          // Deletes this file/directory.  
public void deleteOnExit()       // Deletes this file/directory when the program terminates.  
public boolean renameTo(File dest) // Renames this file.  
public boolean mkdir()            // Makes (Creates) this directory.  
public String[] list()           // List the contents of this directory in a String-array  
public File[] listFiles()        // List the contents of this directory in a File-array
```

```
public String[] list()      // List the contents of this directory in a String-array  
public File[] listFiles() // List the contents of this directory in a File-array
```

```
public String[] list(FilenameFilter filter)  
public File[] listFiles(FilenameFilter filter)  
public File[] listFiles(FileFilter filter)
```

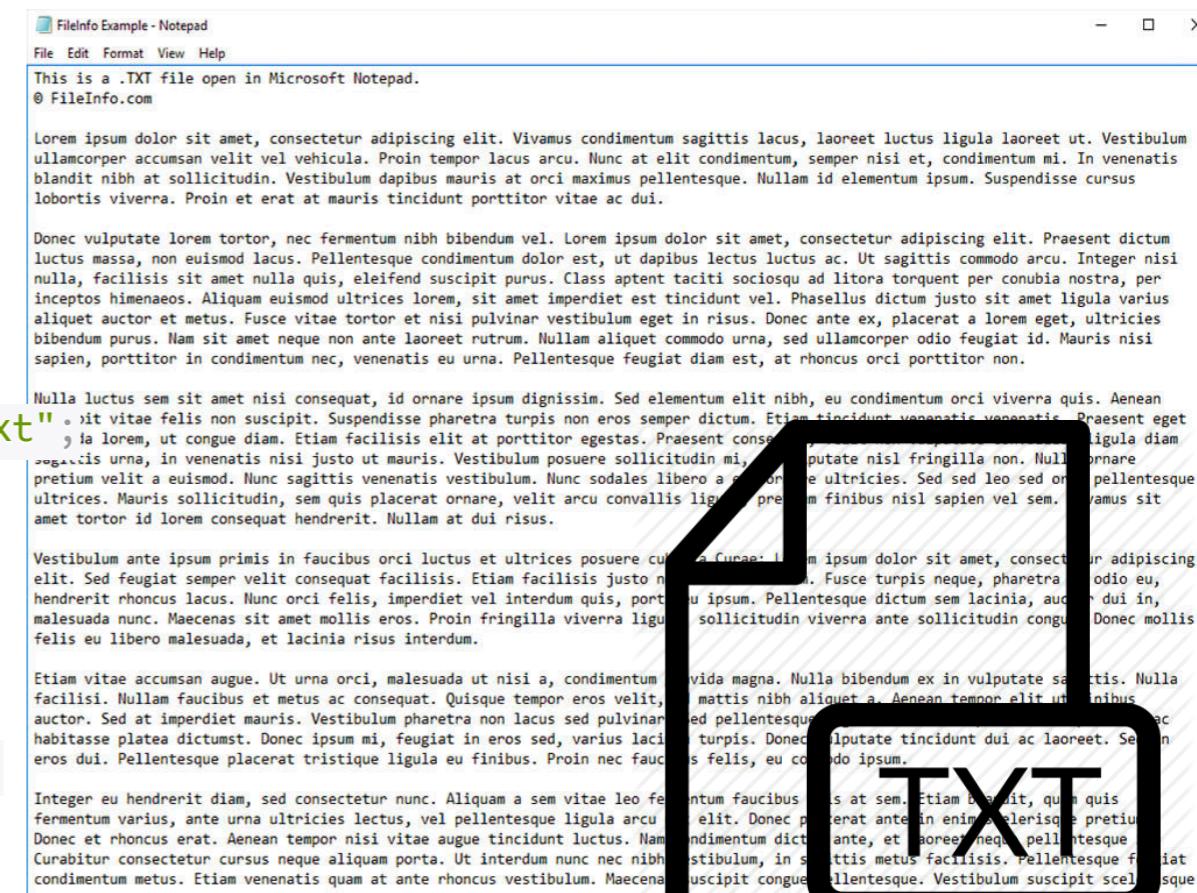
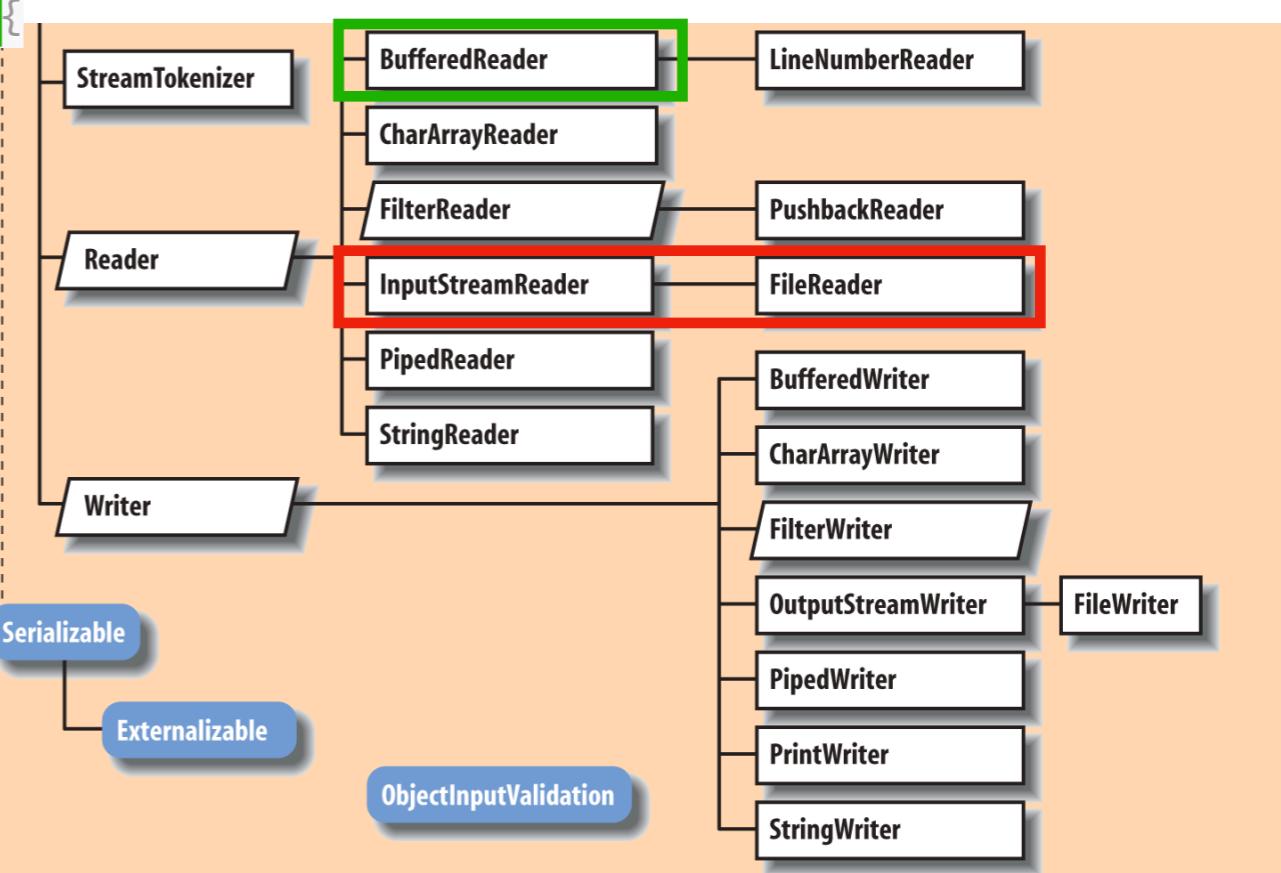
```
public interface FileFilter {  
    public boolean accept(File pathname)  
}
```

```
public interface FilenameFilter{  
    public boolean accept(File dir, String filename)  
}
```

```
public class ListDirectoryWithFilter {  
    public static void main(String[] args) {  
        File dir = new File("."); // current working directory  
        if (dir.isDirectory()) {  
            // List only files that meet the filtering criteria  
            // programmed in accept() method of FilenameFilter.  
            String[] files = dir.list(new FilenameFilter() {  
                public boolean accept(File dir, String file) {  
                    return file.endsWith(".java");  
                }  
            }); // an anonymous inner class as FilenameFilter  
            for (String file : files) {  
                System.out.println(file);  
            }  
        }  
    }  
}
```

文本文件读取

```
public class ReadFileExample1 {  
  
    private static final String FILENAME = "E:\\test\\filename.txt";  
    public static void main(String[] args) {  
        BufferedReader br = null;  
        FileReader fr = null;  
  
        try {  
            //br = new BufferedReader(new FileReader(FILENAME));  
            fr = new FileReader(FILENAME);  
            br = new BufferedReader(fr);  
        } catch (IOException e) {  
            e.printStackTrace();  
        } finally {  
            if (fr != null) {  
                try {  
                    fr.close();  
                } catch (IOException e) {  
                    e.printStackTrace();  
                }  
            }  
            if (br != null) {  
                try {  
                    br.close();  
                } catch (IOException e) {  
                    e.printStackTrace();  
                }  
            }  
        }  
    }  
}
```



结构化文本文件读取

SQL: Structured Query Language

Scanner(File source)

Constructs a new Scanner that produces values scanned from the specified file.

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

Scanner(InputStream source)

Constructs a new Scanner that produces values scanned from the specified input stream.

```
Scanner sc = new Scanner(System.in);
```

Scanner(String source)

Constructs a new Scanner that produces values scanned from the specified string.

```
Scanner s = new Scanner("123 asda bf 12 adafafa lda");
```

File Edit Format View Help

```
Supplier IDs, ID, Product Code, Product Name, Description, Standard Cost, List Price, Reorder Level, Target Level, Quantity Per Unit, Discontinued, Minimum Reorder Quantity, Category, Att.  
4,1,NWTB-1, Northwind Traders Chai,,,$13.50,$18.00,10,40,10 boxes x 20 bags, FALSE, 10, Beverages,  
10,3,NWTCO-3, Northwind Traders Syrup,,,$7.50,$10.00,25,100,12 - 550 ml bottles, FALSE, 25, Condiments,  
10,4,NWTCO-4, Northwind Traders Cajun Seasoning,,,$16.50,$22.00,10,40,48 - 6 oz jars, FALSE, 10, Condiments,  
10,5,NWTO-5, Northwind Traders Olive Oil,,,$16.01,$21.35,10,40,36 boxes, FALSE, 10, Oil,  
2;6,6,NWTJP-6, Northwind Traders Boysenberry Spread,,,$18.75,$25.00,25,100,12 - 8 oz jars, FALSE, 25, "Jams, Preserves",  
2,7,NWTDFN-7, Northwind Traders Dried Pears,,,$22.50,$30.00,10,40,12 - 1 lb pkgs., FALSE, 10, Dried Fruit & Nuts,  
8,8,NWTS-8, Northwind Traders Curry Sauce,,,$30.00,$40.00,10,40,12 - 12 oz jars, FALSE, 10, Sauces,  
2;6,14,NWTDFN-14, Northwind Traders Walnuts,,,$17.44,$23.25,10,40,40 - 100 g pkgs., FALSE, 10, Dried Fruit & Nuts,  
6,17,NWTCFV-17, Northwind Traders Fruit Cocktail,,,$29.25,$39.00,10,40,15.25 OZ, FALSE, 10, Canned Fruit & Vegetables,  
1,19,NWTBGM-19, Northwind Traders Chocolate Biscuits Mix,,,$6.90,$9.20,5,20,10 boxes x 12 pieces, FALSE, 5, Baked Goods & Mixes,  
2;6,20,NWTJP-6, Northwind Traders Marmalade,,,$60.75,$81.00,10,40,30 gift boxes, FALSE, 10, "Jams, Preserves",  
1,21,NWTBGM-21, Northwind Traders Scones,,,$7.50,$10.00,5,20,24 pkgs. x 4 pieces, FALSE, 5, Baked Goods & Mixes,  
4,34,NWTB-34, Northwind Traders Beer,,,$10.50,$14.00,15,60,24 - 12 oz bottles, FALSE, 15, Beverages,  
7,40,NWTCM-40, Northwind Traders Crab Meat,,,$13.80,$18.40,30,120,24 - 4 oz tins, FALSE, 30, Canned Meat,  
6,41,NWTSO-41, Northwind Traders Clam Chowder,,,$7.24,$9.65,10,40,12 - 12 oz cans, FALSE, 10, Soups,  
3;4,43,NWTB-43, Northwind Traders Coffee,,,$34.50,$46.00,25,100,16 - 500 g tins, FALSE, 25, Beverages,  
10,48,NWTCA-48, Northwind Traders Chocolate,,,$9.56,$12.75,25,100,10 pkgs, FALSE, 25, Candy,  
2,51,NWTDFN-51, Northwind Traders Dried Apples,,,$39.75,$53.00,10,40,50 - 300 g pkgs., FALSE, 10, Dried Fruit & Nuts,  
1,52,NWTG-52, Northwind Traders Long Grain Rice,,,$5.25,$7.00,25,100,16 - 2 kg boxes, FALSE, 25, Grains,  
1,56,NWTP-56, Northwind Traders Gnocchi,,,$28.50,$38.00,30,120,24 - 250 g pkgs., FALSE, 30, Pasta,  
1,57,NWTP-57, Northwind Traders Ravioli,,,$14.63,$19.50,20,80,24 - 250 g pkgs., FALSE, 20, Pasta,  
8,65,NWTS-65, Northwind Traders Hot Pepper Sauce,,,$15.79,$21.05,10,40,32 - 8 oz bottles, FALSE, 10, Sauces,  
8,66,NWTS-66, Northwind Traders Tomato Sauce,,,$12.75,$17.00,20,80,24 - 8 oz jars, FALSE, 20, Sauces,  
5,72,NWTD-72, Northwind Traders Mozzarella,,,$26.10,$34.80,10,40,24 - 200 g pkgs., FALSE, 10, Dairy Products,  
2;6,74,NWTDFN-74, Northwind Traders Almonds,,,$7.50,$10.00,5,20,5 kg pkg., FALSE, 5, Dried Fruit & Nuts,  
10,77,NWTCO-77, Northwind Traders Mustard,,,$9.75,$13.00,15,60,12 boxes, FALSE, 15, Condiments,  
2,80,NWTDFN-80, Northwind Traders Dried Plums,,,$3.00,$3.50,50,75,1 lb bag, FALSE, 25, Dried Fruit & Nuts,  
3,81,NWTB-81, Northwind Traders Green Tea,,,$2.00,$2.99,100,125,20 bags per box, FALSE, 25, Beverages,  
1,82,NWTC-82, Northwind Traders Granola,,,$2.00,$4.00,20,100,, FALSE,, Cereal,  
9,83,NWTCS-83, Northwind Traders Potato Chips,,,$0.50,$1.80,30,200,, FALSE,, "Chips, Snacks",  
1,85,NWTBGM-85, Northwind Traders Brownie Mix,,,$9.00,$12.49,10,20,3 boxes, FALSE, 5, Baked Goods & Mixes,  
1,86,NWTBGM-86, Northwind Traders Cake Mix,,,$10.50,$15.99,10,20,4 boxes, FALSE, 5, Baked Goods & Mixes,  
7,87,NWTB-87, Northwind Traders Tea,,,$2.00,$4.00,20,50,100 count per box, FALSE,, Beverages,  
6,88,NWTCFV-88, Northwind Traders Pears,,,$1.00,$1.30,10,40,15.25 OZ, FALSE,, Canned Fruit & Vegetables,  
6,89,NWTCFV-89, Northwind Traders Peaches,,,$1.00,$1.50,10,40,15.25 OZ, FALSE,, Canned Fruit & Vegetables,  
6,90,NWTCFV-90, Northwind Traders Pineapple,,,$1.00,$1.80,10,40,15.25 OZ, FALSE,, Canned Fruit & Vegetables,
```

```
import java.util.Scanner;
import java.io.File;
public class j05{
    public static void main(String[] args) throws Exception{
        //输入测试
        Scanner incmd1 = new Scanner(System.in);
        System.out.println("input a String");
        String xx = incmd1.next();
        System.out.println("xx = " + xx );

        //从文件中读取数据
        File f05 = new File("f005.txt");
        Scanner readit = new Scanner(f05);

        while(readit.hasNext()){
            String firstname = readit.next();
            String middlename = readit.next();
            String lastname = readit.next();
            int score = readit.nextInt();
            System.out.println(firstname + " " + middlename + " " + lastname + " " + score );
        }
        readit.close();
    }
}
```

```
public class readhuman {  
    private static void readfile(String filename) {  
        try {  
            Scanner scanner = new Scanner(new File(filename));  
            scanner.useDelimiter(System.getProperty("line.separator"));  
            while (scanner.hasNext()) {  
                parseline(scanner.next());  
            }  
            scanner.close();  
        } catch (FileNotFoundException e) {  
            System.out.println(e);  
        }  
    }  
  
    private static void parseline(String line) {  
        Scanner linescanner = new Scanner(line);  
        linescanner.useDelimiter(",");  
        //可以修改usedelimiter参数以读取不同分隔符分隔的内容  
        String name = linescanner.next();  
        int age = linescanner.nextInt();  
        String idate = linescanner.next();  
        boolean iscrtified = linescanner.nextBoolean();  
        System.out.println("姓名: "+name+", 年龄: "+ age+", 入司时间: "+ idate+", 验证标记: "+iscrtified );  
    }  
  
    public static void main(String[] args) {  
        if (args.length != 1) {  
            System.err.println("usage: java readhuman file location");  
            System.exit(0);  
        }  
        readfile(args[0]);  
    }  
}
```

老赵,28,feb-01,true
小竹,22,dec-03,false
阿波,21,dec-03,false
凯子,25,dec-03,true