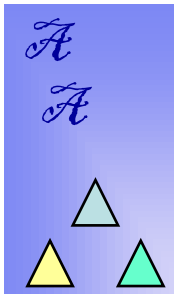


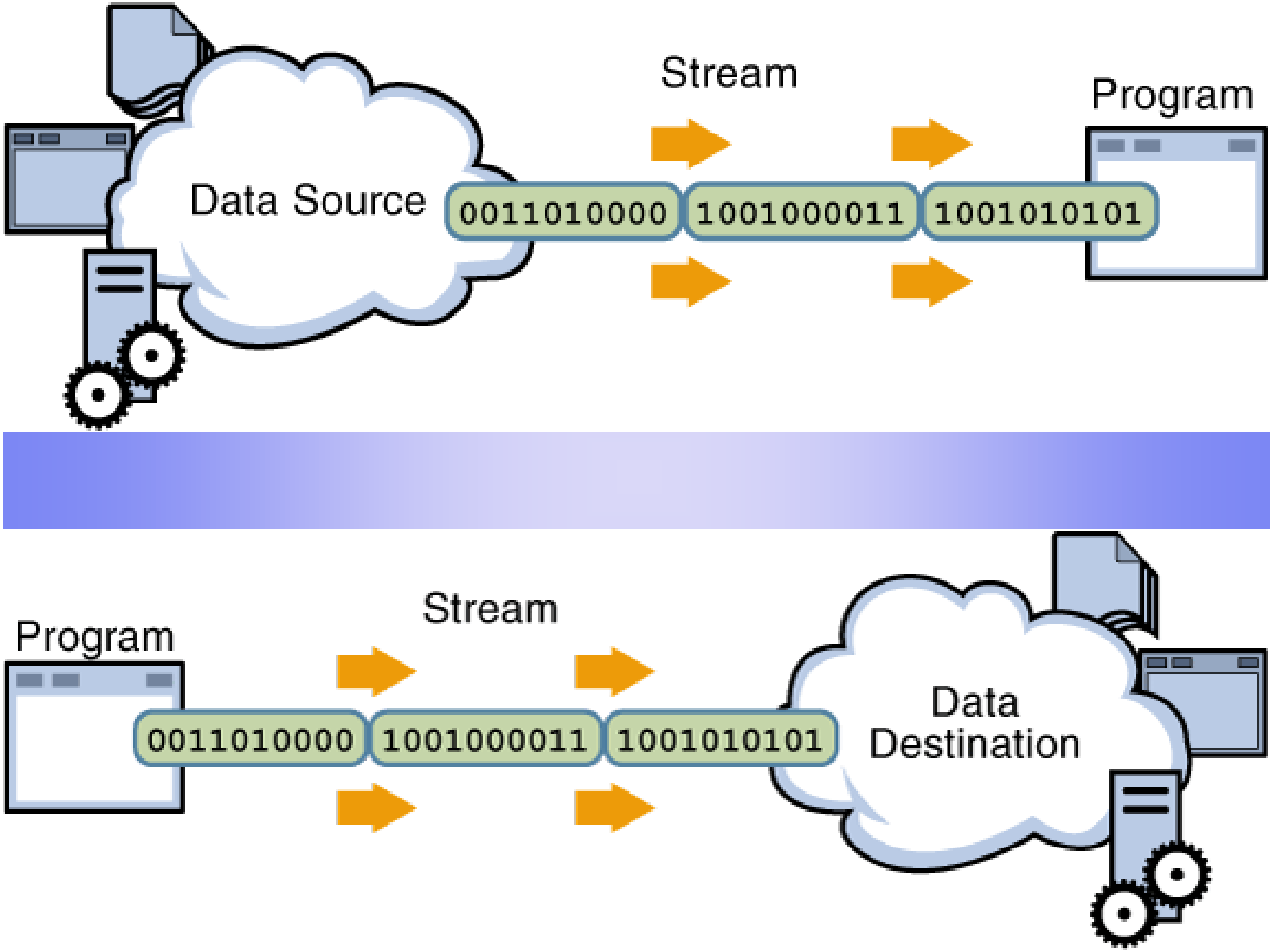
# Потоци (Streams)

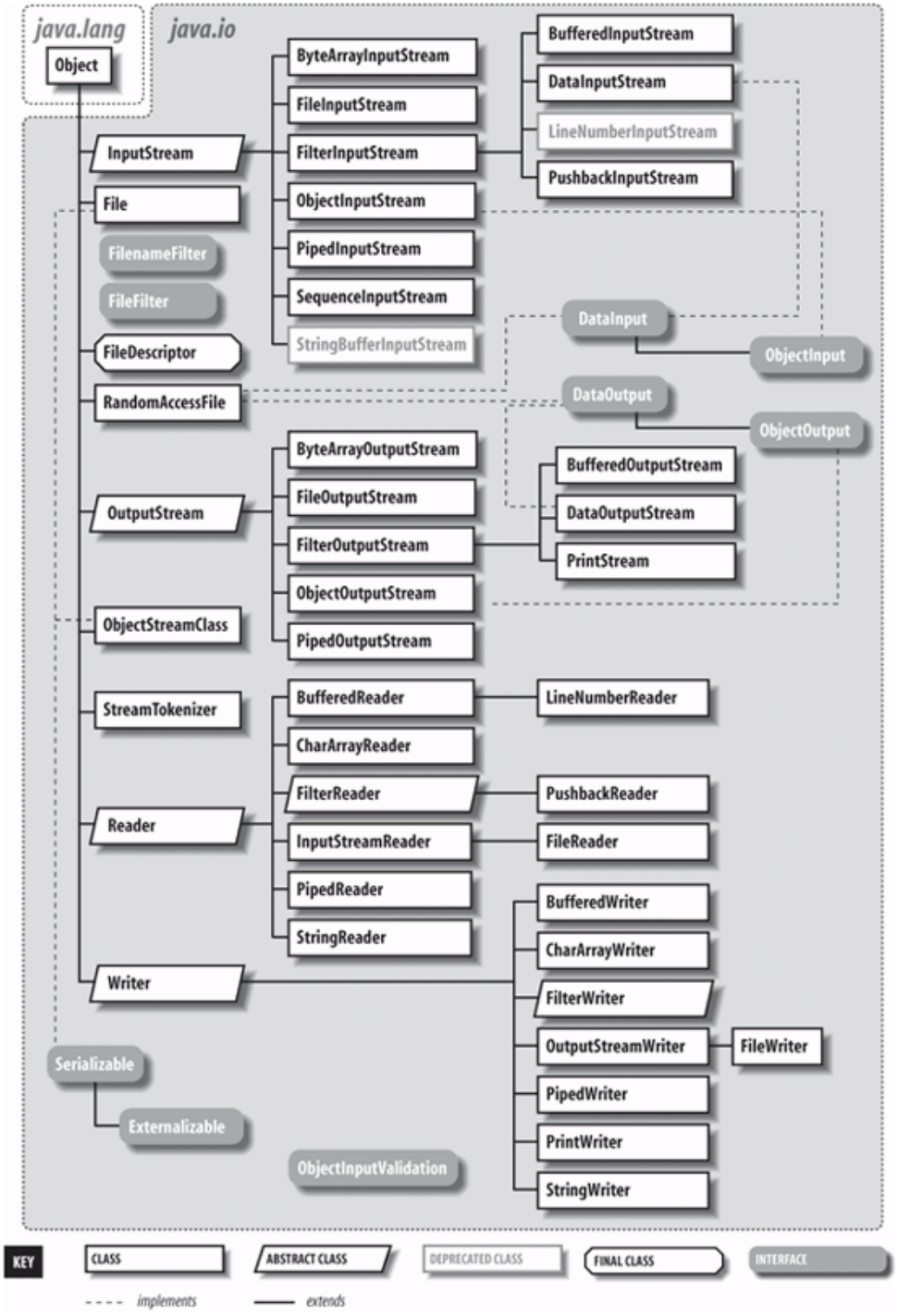
Аделина Алексиева

# Входно-изходни потоци

Абстрактна представа за последователност от данни. Програмата използва входните-изходните потоци, за да чете или записва данни от/на различни устройства.

Имплементирането на йерархия от класове за работа с потоците в Java е в пакета **java.io.**


Stream
Program
Data Source
0011010000
1001000011
1001010101
Program
Stream
0011010000
1001000011
1001010101
Data
Destination


java.lang
java.io
Object
InputStream
File
FilenameFilter
FileFilter
FileDescriptor
RandomAccessFile
ByteArrayInputStream
FileInputStream
FilterInputStream
ObjectInputStream
PipedInputStream
SequenceInputStream
StringBufferInputStream
BufferedInputStream
DataInputStream
LineNumberInputStream
PushbackInputStream
DataInput
ObjectInput
DataOutput
ObjectOutput
OutputStream
ObjectStreamClass
ByteArrayOutputStream
FileOutputStream
FilterOutputStream
ObjectOutputStream
PipedOutputStream
BufferedOutputStream
DataOutputStream
PrintStream
StreamTokenizer
Reader
BufferedReader
CharArrayReader
FilterReader
InputStreamReader
PipedReader
StringReader
LineNumberReader
PushbackReader
FileReader
Writer
BufferedWriter
CharArrayWriter
FilterWriter
OutputStreamWriter
PipedWriter
PrintWriter
StringWriter
FileWriter
Serializable
Externalizable
ObjectInputValidation
KEY
CLASS
ABSTRACT CLASS
DEPRECATED CLASS
FINAL CLASS
INTERFACE
implements
extends

# java.io

# Видове

- Бинарни потоци: **InputStream** и **OutputStream**
- Текстови потоци: **Reader** и **Writer**

# Класове за работа с бинарни потоци 1/2

- **InputStream** – абстрактен клас който описва входните потоци;
- **OutputStream** - абстрактен клас който описва изходните потоци;
- **BufferedInputStream**
- **BufferedOutputStream**
- **ByteArrayInputStream-** чете байтове от масив
- **ByteArrayOutputStream** – записва байтове в масив

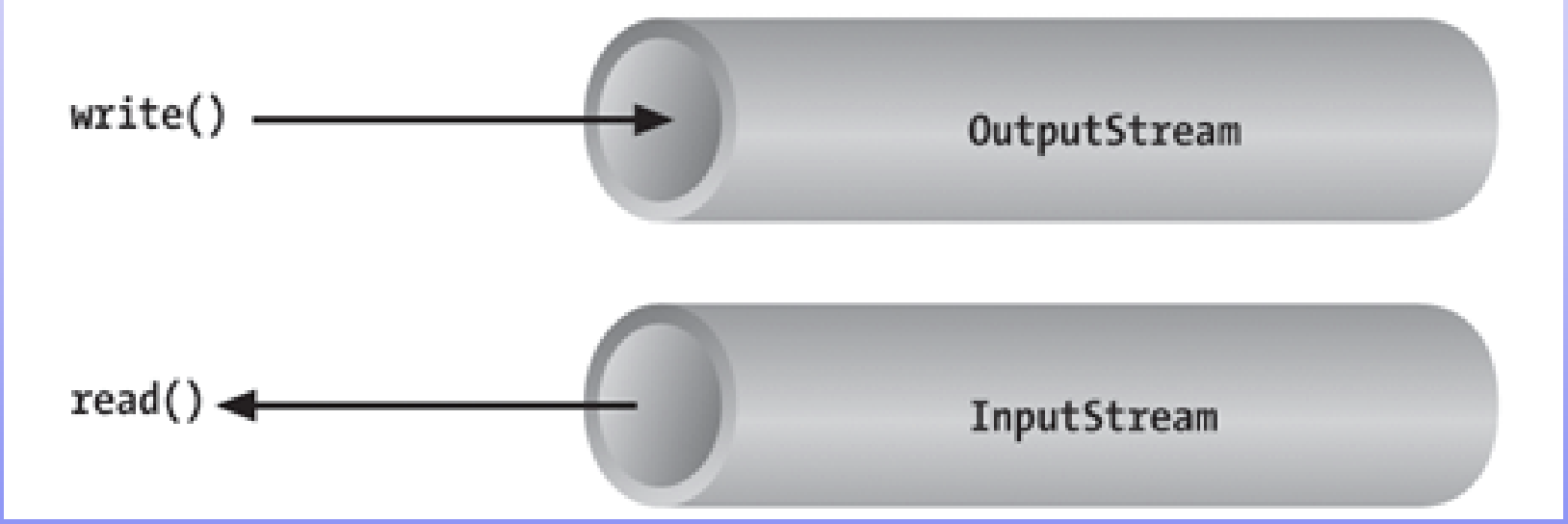
# Класове за работа с бинарни потоци 2/2

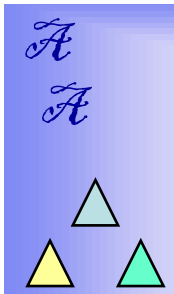
- **DataInputStream** – съдържа методи за четене на стандартни Java типове данни
- **DataOutputStream**
- **FileInputStream** – четене от файл;
- **FileOutputStream** – запис във файл;
- **FilterInputStream** – имплементира **InputStream;**
- **FilterOutputStream** – имплементира **OutputStream;**

# public abstract class OutputStream

- public abstract void write(int b) throws IOException
- public void write(byte[] data) throws IOException
- public void write(byte[] data, int offset, int length) throws IOException
- public void flush( ) throws IOException
- public void close( ) throws IOException

# Основна функционалност





# public abstract class InputStream

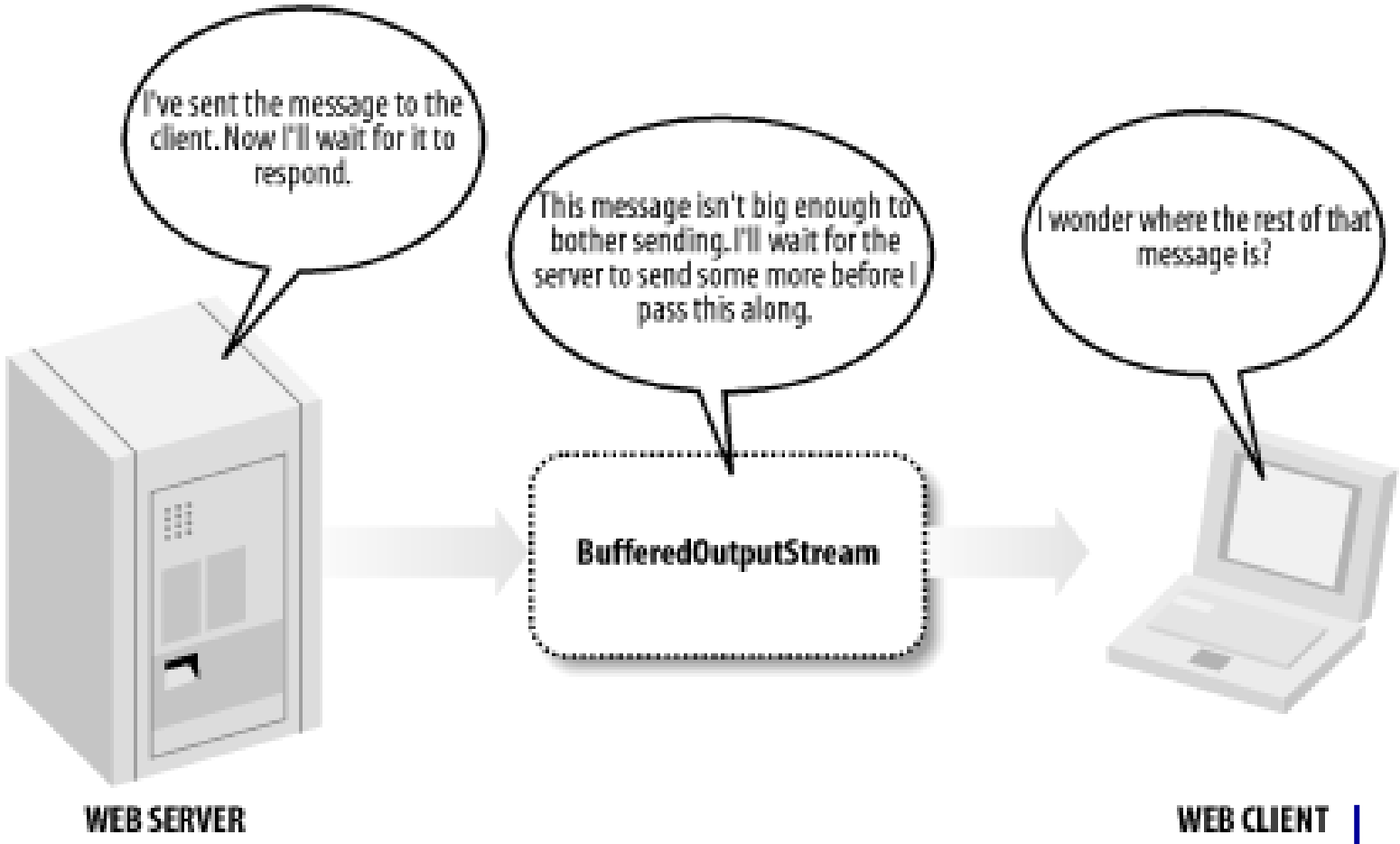
- public abstract int read( ) throws IOException
- public int read(byte[] input) throws IOException
- public int read(byte[] input, int offset, int length) throws IOException
- public long skip(long n) throws IOException
- public int available( ) throws IOException
- public void close( ) throws IOException

# Буфериране

- BufferedInputStream
- BufferedOutputStream

```
in = new BufferedInputStream(new
          FileInputStream("source.txt"));
out = new BufferedOutputStream(new
     FileOutputStream("result.txt"));
```

# Използване на flush( )



# Класове за работа с текстови потоци 1/2

- Reader – абстрактен клас който описва входни текстови потоци;
- Writer - абстрактен клас който описва изходни текстови потоци;
- BufferedReader – с буфериране;
- BufferedWriter – с буфериране;
- CharArrayReader – входен поток, който чете от масив;
- CharArrayWriter – записва в масив;

# Класове за работа с текстови потоци 2/2

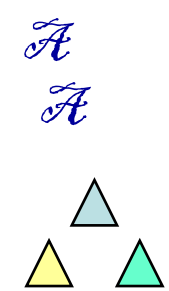
- FileReader, FileWriter– чете/ записва от/ във файл;
- FilterReader, FilterWriter– с филтриране;
- InputStreamReader, OutputStreamWriter– превръща байтовете в символи;
- LineNumberReader – входен поток, който брои линиите;
- PrintWriter – изходящ текстови поток, който има методи **print( )** и **println( )**
- StringReader, StringWriter – чете/ записва от/в низ;

# java.io.Writer

- protected Writer( )
- protected Writer(Object lock)
- public abstract void write(char[] text, int offset, int length) throws IOException
- public void write(int c) throws IOException
- public void write(char[] text) throws IOException
- public void write(String s) throws IOException
- public void write(String s, int offset, int length) throws IOException
- public abstract void flush( ) throws IOException
- public abstract void close( ) throws IOException

# java.io.Reader

- protected Reader( )
- protected Reader(Object lock)
- public int read( )
- public int read(char[] text)
- public long skip(long n)
- public boolean ready( )
- public void reset( )
- public abstract void close( )



# OutputStreamWriter

```
OutputStreamWriter w = new OutputStreamWriter(
        new FileOutputStream("OdysseyB.txt"),
                             "Cp1253");
    w.write(“ ἦμος δ’ἠριγένεια φάνη ῥοδοδάκτυλος Ἠώς ");
```

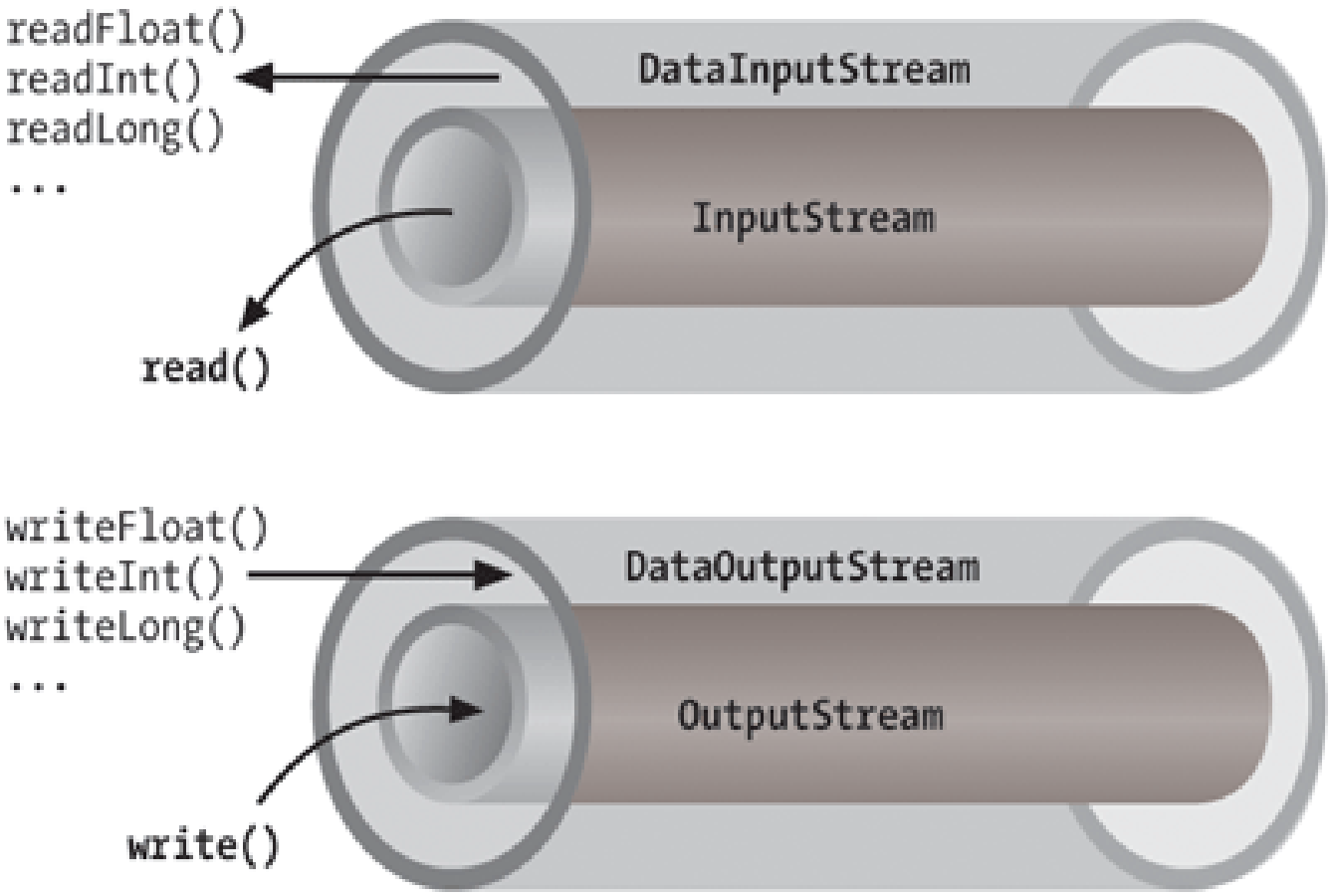
http://java.sun.com/j2se/1.4.2/docs/guide/intl/encoding.doc.html

# Задача:

Да се копира съдържанието от един файл в друг:

- source.txt – файл от който ще се чете;
- result.txt – файл в който се записва.

# Data streams



# Пример

```
DataInputStream dis = new
    DataInputStream( System.in );
double d = dis.readDouble( );
```

# Работа с файлове

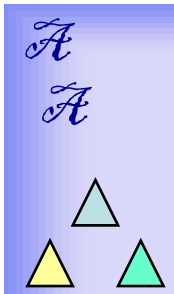
- Клас java.io.File
- Конструктори:

```
File fooFile = new File( "/tmp/foo.txt" );

File fooFile = new File( "/tmp", "foo.txt" );
```

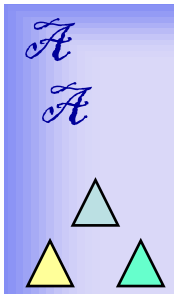
# Методи 1/2

| Method | Return type | Description |
|---|---|---|
| `canRead( )` | `Boolean` | Is the file (or directory) readable? |
| `canWrite( )` | `Boolean` | Is the file (or directory) writable? |
| `createNewFile( )` | `Boolean` | Creates a new file. |
| `createTempFile (String pfx, String sfx)` | `File` | Static method to create a new file, with the specified prefix and suffix, in the default temp file directory. |
| `delete( )` | `Boolean` | Deletes the file (or directory). |
| `deleteOnExit( )` | `Void` | When it exits, Java runtime system deletes the file. |
| `exists( )` | `boolean` | Does the file (or directory) exist? |
| `getAbsolutePath( )` | `String` | Returns the absolute path of the file (or directory). |
| `getCanonicalPath( )` | `String` | Returns the absolute, case-correct path of the file (or directory). |
| `getName( )` | `String` | Returns the name of the file (or directory). |
| `getParent( )` | `String` | Returns the name of the parent directory of the file (or directory). |
| `getPath( )` | `String` | Returns the path of the file (or directory). |
| `isAbsolute( )` | `boolean` | Is the filename (or directory name) absolute? |
| `isDirectory( )` | `boolean` | Is the item a directory? |



# Методи 2/2

| | | |
|---|---|---|
| `isFile( )` | `boolean` | Is the item a file? |
| `isHidden( )` | `boolean` | Is the item hidden? (System-dependent.) |
| `lastModified( )` | `long` | Returns the last modification time of the file (or directory). |
| `length( )` | `long` | Returns the length of the file. |
| `list( )` | `String []` | Returns a list of files in the directory. |
| `listFiles( )` | `File[]` | Returns the contents of the directory as an array of `File` objects. |
| `listRoots( )` | `File[]` | Returns array of root filesystems if any (e.g., C:/, D:/). |
| `mkdir( )` | `boolean` | Creates the directory. |
| `mkdirs( )` | `boolean` | Creates all directories in the path. |
| `renameTo(File dest)` | `boolean` | Renames the file (or directory). |
| `setLastModified( )` | `boolean` | Sets the last-modified time of the file (or directory). |
| `setReadOnly( )` | `boolean` | Sets the file to read-only status. |
| `toURL( )` | `java.net.URL` | Generates a URL object for the file (or directory). |



# Задача

ListIt.java

Извежда съдържанието на файл или директория на стандартния изход

# Компресиране на данни

- java.util.zip
- GZIP формат за компресия
  - клас ZIPOutputStream
  - клас ZipInputStream
- ZIP формат за компресия
  - клас GZIPOutputStream
  - клас GZIPInputStream

# Пример за създаване на ZipOutputStream

```
ZipOutputStream zipout;
  try {
     FileOutputStream out = new
        FileOutputStream("archive.zip");
      zipout = new ZipOutputStream(out);
   } catch (IOException e) {}
```

# Пример за добавяне в архива

```
try {
   ZipEntry entry = new ZipEntry("First");
   zipout.putNextEntry(entry);
   ZipEntry entry = new ZipEntry("Second");
   zipout.putNextEntry(entry);
   . . .
} catch (IOException e) {}
```

# Пример за създаване на ZipInputStream

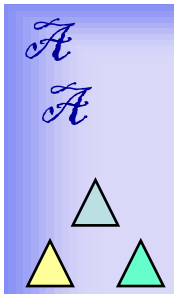
```
ZipInputStream zipin;
try {
      FileInputStream in = new
          FileInputStream("archive.zip");
      zipin = new ZipInputStream(in);
} catch (IOException e) {}
```

# Пример за извличане от архив

```
try {
 ZipEntry first = zipin.getNextEntry( );
} catch (IOException e) {}
```

# Задачи

- Пример за компресиране чрез GZIP формат:
  - GZip.java
- Пример за декомпресиране на GZIP формат:
  - GUnzip.java



# Ресурси

***Ресурси към темата:***

*http://java.sun.com/docs/books/tutorial/essential/io/index.html*