

Java I

Урок X

Java IO

Java IO

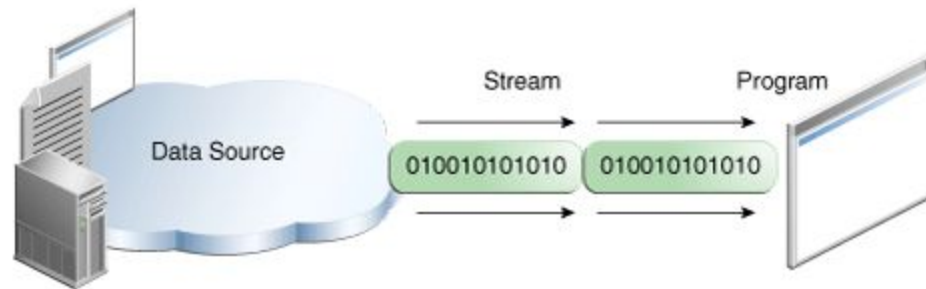
- Input/Output
- Все, что нам необходимо для работы с вводом/выводом находится в пакете `java.io`

Streams

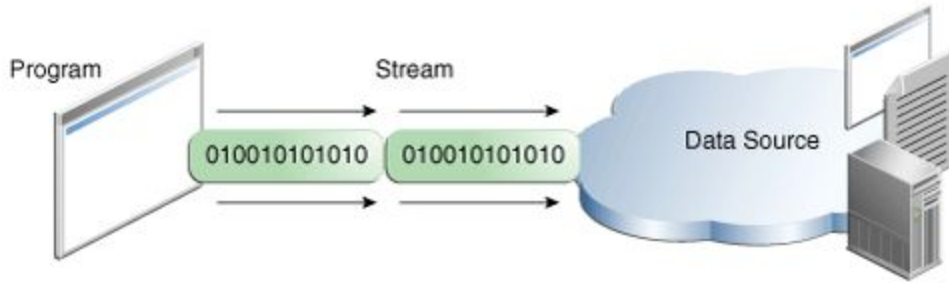
Stream (Поток)

- Поток представляет из себя последовательность данных идущих в программу или из неё

InputStream



OutputStream



Работа с файлами, директориями

File

```
String path = "src/lv/akurss/lesson10/lecture/file/test.txt";  
File file = new File(path);
```

```
System.out.println(file.exists());
```

```
if (!file.exists()) {  
    file.createNewFile();  
}
```

File

```
System.out.println(file.getName());
```

```
System.out.println(file.getPath());
```

```
System.out.println(file.getAbsolutePath());
```

```
System.out.println(file.getParent());
```

File

```
System.out.println(file.length());  
System.out.println(file.getTotalSpace() / Math.pow(2, 30) + " GB");  
System.out.println(file.getFreeSpace() / Math.pow(2, 30) + " GB");
```

File

```
System.out.println(file.isDirectory());
```

```
System.out.println(file.isHidden());
```

```
System.out.println(file.canRead());
```

```
System.out.println(file.canWrite());
```

```
System.out.println(file.canExecute());
```

File

```
File renamedFile = new File("src/lv/akurss/lesson10/lecture/file/test2.txt");
```

```
System.out.println(file.renameTo(renamedFile));
```

```
System.out.println(renamedFile.delete());
```

Directory

```
String path = "src/lv/akurss/lesson10/lecture/file/folder";  
File folder = new File(path);
```

```
System.out.println(folder.exists());
```

```
if (!folder.exists()) {  
    folder.mkdir();  
}
```

Directory

```
listFiles(new File("."));  
...  
private static void listFiles(File file) {  
    File[] files = file.listFiles();  
    if (files == null) {  
        return;  
    }  
    for (File f : files) {  
        if (f.isDirectory()) {  
            System.out.println("Directory: " + f.getName());  
            listFiles(f);  
        } else {  
            System.out.println("- " + f.getName());  
        }  
    }  
}
```

Заккрытие потоков

Старый подход

```
OutputStream outputStream = null;
```

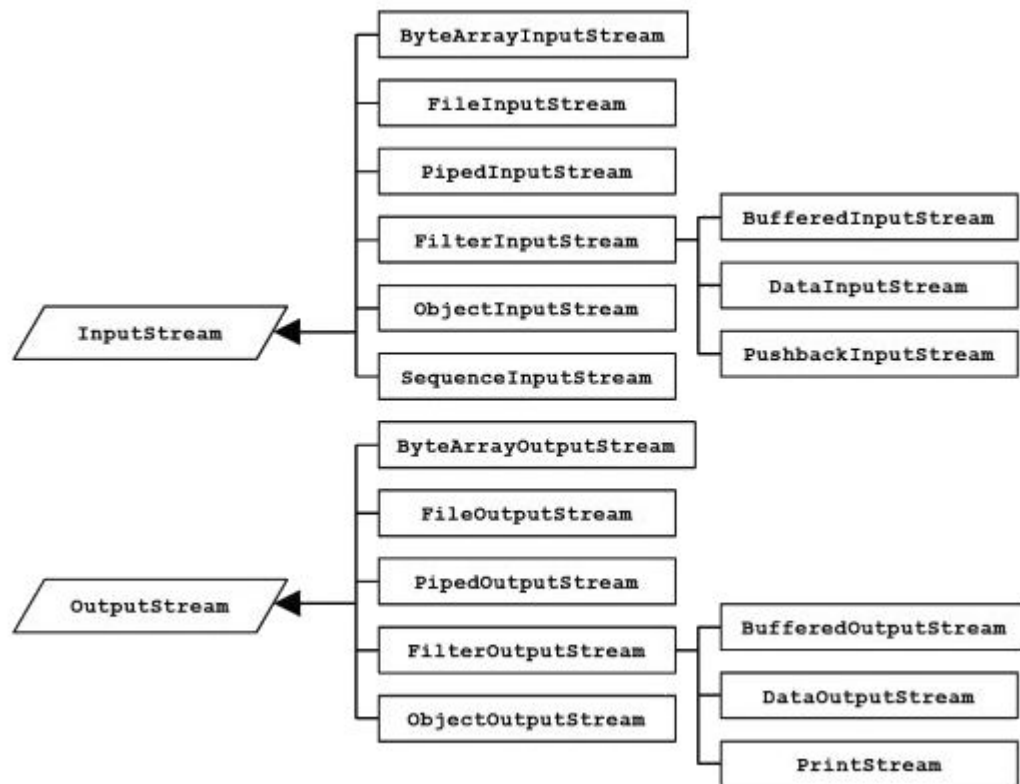
```
try {  
    fileOutputStream = ...;  
    ...  
} catch (IOException e) {  
    ...  
} finally {  
    try {  
        if (outputStream != null) {  
            outputStream.close();  
        }  
    } catch (IOException e) {  
        ...  
    }  
}
```

НОВЫЙ ПОДХОД

```
try (OutputStream outputStream = ...) {  
  
    ...  
  
} catch (IOException e) {  
    ...  
}
```

Байтовые потоки

Байтовые потоки



Запись в файл

```
byte[] bytes = ...;
```

```
File output = new File(...);
```

```
try (OutputStream fileOutputStream = new FileOutputStream(output)) {
```

```
    for (byte eachByte : bytes) {  
        fileOutputStream.write(eachByte);  
    }
```

```
} catch (IOException e) {  
    ...  
}
```

Чтение из файла

```
File input = new File(...);

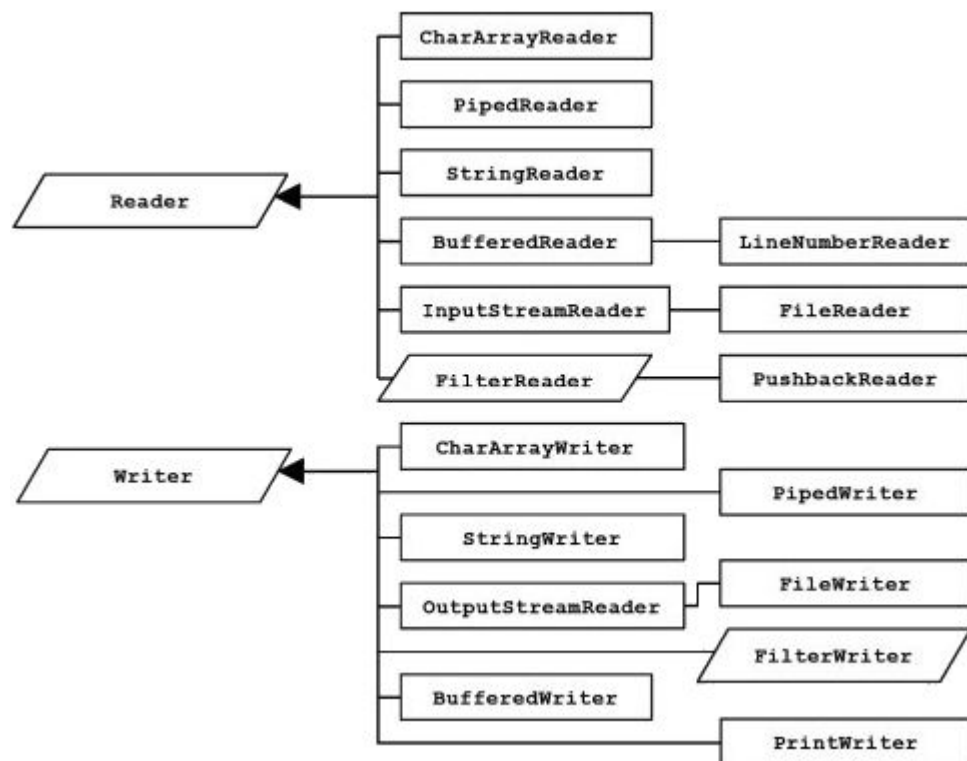
try (InputStream fileInputStream = new FileInputStream(input)) {

    int symbol;
    while ((symbol = fileInputStream.read()) != -1) {
        System.out.print(symbol);
    }

} catch (IOException e) {
    ...
}
```

Символьные потоки

Символьные потоки



Запись в файл

```
File output = new File(...);
```

```
try (Writer fileWriter = new FileWriter(output)) {  
    fileWriter.write("Hello, world!!!");  
} catch (IOException e) {  
    ...  
}
```

Чтение из файла (по символу)

```
File input = new File(...);

try (Reader reader = new FileReader(input)) {
    int symbol;
    while ((symbol = reader.read()) != -1) {
        System.out.print((char) symbol);
    }
} catch (IOException e) {
    ...
}
```

Чтение из файла (по строке)

```
File input = new File(...);
```

```
try (BufferedReader reader = new BufferedReader(new FileReader(input))) {  
    String line;  
    while ((line = reader.readLine()) != null) {  
        System.out.print(line);  
    }  
} catch (IOException e) {  
    ...  
}
```