

Потоки ввода/вывода

- Байтовые
- Символьные
- Форматные

Выбор подходящего класса:

- Анализ решаемой задачи
- Выбор потоков с подходящими методами
- Выбор потока с подходящим конструктором

Пакет `java.io`

Байтовые потоки вывода

Базовый класс OutputStream (абстрактный)

Конструктор

OutputStream()

Методы:

abstract void write(int b) throws IOException

void write(byte[] b) throws IOException

void write(byte[] b, int off, int len) throws
IOException

void flush()

void close()

Наследники OutputStream

`FilterOutputStream` – базовый класс для специализированных выходных потоков

- `BufferedOutputStream` (вывод через буфер)
- `PrintStream` (форматный вывод)

`FileOutputStream` – вывод данных в файл

`ObjectOutputStream` – запись объектов в поток вывода

Буферизованный вывод

Класс `BufferedOutputStream`

Конструкторы:

`BufferedOutputStream(OutputStream out)`

`BufferedOutputStream(OutputStream out, int size)`

Методы:

`public void write(int b) throws IOException`

`public void write(byte[] b, int off, int len)
throws IOException`

`public void flush() throws IOException`

`public void close() throws IOException`

Форматный вывод

Класс `PrintStream`

Конструкторы:

`PrintStream(OutputStream out)`

`PrintStream(OutputStream out, boolean autoFlush, String encoding)`

`PrintStream(String fileName)`

`PrintStream(File file)`

Методы:

`void print(double d)`

`void print(int i)`

`void println(String x)`

`PrintStream printf(String format, Object... args)`

Байтовый вывод в файл

Класс `FileOutputStream`

Конструкторы:

`FileOutputStream(File file) throws
FileNotFoundException`

`FileOutputStream(String name) throws
FileNotFoundException`

`FileOutputStream(File file, boolean append)
throws FileNotFoundException`

Методы:

`public void write(int b) throws IOException`

`public void write(byte[] b) throws
IOException`

`public void close() throws IOException`

Байтовый вывод объектов

Класс `ObjectOutputStream`

Конструктор:

`ObjectOutputStream(OutputStream out)` throws
`IOException`

Методы:

`public void writeDouble(double val)` throws
`IOException`

`public void writeInt(int val)` throws `IOException`

`public void writeChars(String str)` throws
`IOException`

`public final void writeObject(Object obj)`
throws `IOException`

Символьные потоки вывода

Базовый класс `Writer` (абстрактный)

Конструктор

`Writer()`

Методы:

`void write(int c) throws IOException`

`void write(char[] cbuf) throws IOException`

`void write(char[] cbuf, int off, int len) throws
IOException`

`void write(String str) throws IOException`

`abstract void flush() throws IOException`

`abstract void close() throws IOException`

Буферизованный символьный ВЫВОД

Класс `BufferedWriter`

Конструкторы:

`BufferedWriter(Writer out)`

`BufferedWriter(Writer out, int sz)`

Методы:

`public void write(int c) throws IOException`

`public void write(String s, int off, int len) throws
IOException`

`public void newLine() throws IOException`

`public void flush() throws IOException`

`public void close() throws IOException`

Вывод символов в заданной кодировке

Класс `OutputStreamWriter`

Конструкторы:

```
OutputStreamWriter(OutputStream out)
```

```
OutputStreamWriter(OutputStream out, String  
                    charsetName)
```

Методы:

```
public String getEncoding()
```

```
public void write(int c) throws IOException
```

```
public void write(char[] cbuf, int off, int len)  
                throws IOException
```

ФОРМАТНЫЙ СИМВОЛЬНЫЙ ВЫВОД

Класс `PrintWriter`

Конструкторы:

`PrintWriter(OutputStream out)`

`PrintWriter(String fileName)`

`PrintWriter(File file)`

`PrintWriter(String fileName, String csn)`

Методы:

`void print(double d)`

`void print(int i)`

`void println(String x)`

`PrintWriter printf(String format, Object... args)`

Байтовые потоки ввода

Базовый класс InputStream (абстрактный)

Конструктор

InputStream()

Методы:

`public abstract int read() throws IOException`

`public int read(byte[] b) throws IOException`

`public int read(byte[] b, int off, int len) throws IOException`

`void mark(int readlimit)` - помещает метку в текущую точку входного потока, которая остаётся корректной до тех пор, пока не будет прочитано `readlimit` байт

`reset()` - сбрасывает входной указатель в ранее установленную метку

`long skip(long byteCount)` - пропускает `byteCount` байт ввода, возвращая количество проигнорированных байтов

`void close() throws IOException`

Наследники `InputStream`

`FilterInputStream` – базовый класс для специализированных входных потоков

- `BufferedInputStream` (ввод через буфер)

`FileInputStream` – чтение данных из файла

`ObjectInputStream` – чтение записанного в виде потока байтов объекта

Буферизованный ввод

Класс `BufferedInputStream`

Конструкторы:

`BufferedInputStream(InputStream in)`

`BufferedInputStream(InputStream in)`

Методы:

`public int read() throws IOException`

`public int read(byte[] b, int off, int len)
throws IOException`

`public void close() throws IOException`

Чтение байтов из файла

Класс `FileInputStream`

Конструкторы:

`FileInputStream(File file)` throws
`FileNotFoundException`

`FileInputStream(String name)` throws
`FileNotFoundException`

Методы:

`public int read()` throws `IOException`

`public int read(byte[] b)` throws `IOException`

`public void close()` throws `IOException`

Байтовый ввод объектов

Класс `ObjectInputStream`

Конструктор:

`ObjectInputStream(InputStream in)` throws
`IOException`

Методы:

`double readDouble()` throws `IOException`

`int readInt()` throws `IOException`

`public String readUTF()` throws `IOException`

`public final Object readObject()` throws
`IOException, ClassNotFoundException`

Символьные потоки ввода

Базовый класс Reader (абстрактный)

Конструктор

```
protected Reader()
```

Методы:

```
public int read() throws IOException
```

```
public int read(char[] cbuf) throws IOException
```

```
public abstract int read(char[] cbuf, int off, int len)  
                                throws IOException
```

```
public void mark(int readAheadLimit) throws IOException
```

```
public void reset() throws IOException
```

```
abstract void close() throws IOException
```

Чтение символов в заданной кодировке

Класс `InputStreamReader`

Конструкторы:

```
InputStreamReader(InputStream in)
```

```
InputStreamReader(InputStream in, String  
                                charsetName)
```

Методы:

```
public String getEncoding()
```

```
public int read() throws IOException
```

```
public int read(char[] cbuf, int off, int len)  
              throws IOException
```

Буферизованный символьный ввод

Класс `BufferedReader`

Конструкторы:

`BufferedReader(Reader in)`

`BufferedReader(Reader in, int sz)`

Методы:

`public String readLine() throws IOException`