I/O, Strings

|  |  |  |  |
| --- | --- | --- | --- |
| № урока: | 3 | Курс: | Java Professional |
|  | | |  |
| Средства обучения: | | | Компьютер с установленной IntelliJ IDEA. |

# Обзор, цель и назначение урока

Потоки ввода/вывода, абстрактные классы «Reader», «Writer», «InputStream», «OutputStream».

**Изучив материал данного занятия, учащийся сможет:**

* Понимать, как осуществляется запись в файл.
* Понимать, как осуществляется чтение с файла.
* Разбирать символьные потоки от байтовых.
* Понимать работу пула.
* Обрабатывать строки.

# Содержание урока

1. InputStream, OutputStream.
2. FileInputStream, BufferedReader
3. Абстрактные классы «Reader», «Writer», «InputStream», «OutputStream»
4. Основные методы вставка/удаление/добавление/изменение файла
5. Классы «String», «StringBuilder», «StringBuffer»,

# Резюме

**Поток** – это абстрактное значение источника или приёмника данных, которые способны обрабатывать информацию.

В программу идет **входной** поток (input stream) символов Unicode или просто байтов, воспринимаемый в программе методами read().

Из программы методами write() или print(), println() выводится **выходной** поток (output stream) **символов** или **байтов**.

При этом неважно, куда направлен поток: на консоль, на принтер, в файл или в сеть, методы write() и print() ничего об этом не знают.

В классе System определенны 3 потока статическими полями

* in (называется стандартный ввод stdin);
* out (называется стандартный ввод stdiout);
* err (называется стандартный ввод stderr);

**Reader** – абстрактный класс, в котором собраны самые общие методы символьного ввода;

**Writer** – абстрактный класс, в котором собраны самые общие методы символьного вывода;

**InputStream** – абстрактный класс с общими методами байтового ввода;

**OutputStream** – абстрактный класс с общими методами байтового вывода.

Каждая строка, создаваемая с помощью оператора **new** или с помощью литерала (заключённая в двойные апострофы), является объектом класса **String**.

Особенностью объекта класса **String** является то, что его значение не может быть изменено после создания объекта при помощи какого-либо метода класса, так как любое изменение строки приводит к созданию нового объекта.

Классы **StringBuilder** и **StringBuffer** являются “близнецами” и по своему предназначению близки к классу **String**, но, в отличие от последнего, содержимое и размеры объектов классов **StringBuilder** и **StringBuffer** можно изменять.

Основным и единственным отличием **StringBuilder** от **StringBuffer** является **потокобезопасность** последнего.

С помощью соответствующих методов и конструкторов объекты классов **StringBuffer**, **StringBuilder** и **String** можно преобразовывать друг в друга.

Конструктор класса **StringBuffer** (также, как и **StringBuilder**) может принимать в качестве параметра объект String.

# Закрепление материала

* Что такое File?
* Что такое Stream?
* В чем разница между символьным и байтовым потоком?
* Какие потоки Вы знаете?
* Назовите все абстрактные классы ввода/вывода?
* Основные особенности класса String?
* В чем разница между String, StringBuilder, StringBuffer?

# Дополнительное задание

**Задание**

Создайте в package текстовый файл и клас, внесите в файл некий текст.

Выведите на экран содержимое файла.

# Самостоятельная деятельность учащегося

Задание 1

В любой из профильных **книг (Хорстман, Эккель)** найти соответствующие темы и закрепить материал. Использование **YouTube, Quizful** приветствуется.

Задание 2

Создайте файл, запишите в него произвольные данные и закройте файл. Затем снова откройте этот файл, прочитайте из него данные и выведете их на консоль.

Задание 3

Необходимо создать строку с текстом (текст взять любой из интернета). Разбить эту строку на 2 подстроки равной длине и вывести на экран каждое предложения с новой строки.

# Рекомендуемые ресурсы

Oracle: I/O

<https://docs.oracle.com/javase/tutorial/essential/io/index.html>