Java IO

|  |  |  |  |
| --- | --- | --- | --- |
| № урока: | 2 | Курс: | Java Advanced |
|  | | |  |
| Средства обучения: | | | Компьютер с установленной IntelliJ IDEA. |

# Обзор, цель и назначение урока

На уроке рассматривается работа с вводом-выводом в Java:

* Создание файлов и папок
* Создание путей к файлам в зависимости от операционной системы
* Работа с файлами и атрибутами файлов
* Чтение содержимого файла
* Запись файла
* Перемещение файлов
* Работа с java.util.Scanner

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

* Создавать файлы и папки
* Перемещать, копировать, изменять и удалять файлы и папки
* Создавать потоки чтения и записи файлов
* Работать с потоками чтения и записи

# Содержание урока

1. Java Input / Output, структура пакета java.io.
2. Входящие и исходящие потоки.
3. Создание файлов и директорий.
4. Задание пути к файлу.
5. Удаление файлов.
6. Выборка файлов с применением фильтра.
7. Переименование файлов.
8. Потоки для чтения DataInputStream и BufferedReader.
9. Потоки для записи FileOutputStream и BufferedWriter.
10. Копирование файлов.
11. Работа со Scanner.

# Резюме

* I/O Stream представляет собой входящий или исходящий поток данных. Источником или назначением потока может быть файл на диске, устройство, другая программа или оперативная память.
* Потоки поддерживают различные типы данных, такие как единичный байт, примитивные типы, локализованные символы и объекты.
* Byte streams (Байтовые потоки) - предназначены для принятия и передачи 8-битных значений. Все байтовые потоки наследуются от InputStream и OutputStream.
* Character Streams (Символьные потоки) – предназначены для принятия и передачи символьных значений в кодировке Unicode. Символьные потоки автоматически транслируются в родной формат из и в локальные настройки. Традиционно, используется формат UTF-16.
* Buffered Streams (буферизированные потоки) – предназначены для чтения / записи данных из / в области памяти, именуемой буфером. Буферизированный входящий поток считывает данные из буфера. Когда буфер пустой, вызывается нативный метод для заполнения буфера. Аналогично, буферизированный исходящий поток записывает данные в буфер, и когда он переполнен – вызывается нативный метод для опустошения буфера.
* Часто есть необходимость передать содержимое буфера до его заполнения. Для этого его необходимо «слить» (flush), то есть вызвать метод flush().
* Объекты класса Scanner позволяют разделить входящие значения на лексемы (tokens) и транслировать отдельные лексемы в соответствии с их типом данных.
* По умолчанию Scanner использует как разделитель для лексем символ «пробел».
* Data Streams (потоки данных) – поддерживают бинарный ввод/вывод из примитивных типов данных и объектов класса String. Все потоки данных наследуют интерфейсы DataInput или DataOutput.
* Object Streams (потоки объектов) – являются расширением Data Streams для передачи объектов. Большинство стандартных классов поддерживают сериализацию их объектов.
* При получении входящего потока мы должны конкретно указать, объект какого класса мы получаем. Если метод readObject() возвращает объект не того класса, бросается ClassNotFoundException.
* Класс File пакета java.io предназначен для работы с файлами.

# Закрепление материала

* Какие есть виды I/O потоков в Java?
* Для чего используется буферизированные потоки?
* Каков алгоритм записи информации в файл?

# Дополнительное задание

Создайте на диске 100 директорий с именами от Folder\_0 до Folder\_99, затем удалите их.

# Самостоятельная деятельность учащегося

Задание 1

Выучите основные конструкции, классы и понятия, рассмотренные на уроке.

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

Посчитайте количество вхождений каждого символа в файле из задания 2. Выведите в консоль массив символов с количеством каждого символа в документе.

Задание 4

Напишите приложение для поиска заданного текстового файла на диске.

# Рекомендуемые ресурсы

Package Java IO

<http://docs.oracle.com/javase/7/docs/api/java/io/package-summary.html>

Java IO tutorial

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

Java.util.Scanner

<http://docs.oracle.com/javase/7/docs/api/java/util/Scanner.html>