(cc) BY



***Коновалова Виктория Александровна***
учитель обществознания
МАОУ – СОШ №7 им. Г.К. Жукова
г. Армавир, Краснодарский край
***Макаров Дмитрий Александрович***
магистрант
ФГБОУ ВО «Армавирский государственный
педагогический университет»
г. Армавир, Краснодарский край

# СИСТЕМАТИЗАЦИЯ ПАКЕТА JAVA IO

***Аннотация**: в статье рассматривается систематизация пакета Java IO.*

***Ключевые слова**: Java, буфер памяти, дисковый файл, ввод-вывод.*

Все приложения разрабатываются для обработки, хранения или передачи информации. Данные извлекаются из источника ввода, а результат выполнения программы направляется адресату вывода. В Java эти понятия определяются очень широко. Например, источником ввода или адресатом вывода может служить сетевое соединение, буфер памяти или дисковый файл. Данными ресурсами можно манипулировать с помощью классов ввода-вывода в Java. И хотя все устройства ввода-вывода отличаются физически, все они описываются единой абстракцией – потоком ввода-вывода. Поток ввода-вывода – это логический объект, который поставляет или потребляет информацию. Поток ввода-вывода присоединяется к физическому устройству системой ввода-вывода в Java. Общий принцип работы изброжен ниже на рисунке 1.

Рис. 1

Из-за большого количества классов предоставляемых платформой Java можно запутаться и не разобраться как функционирует тот или иной класс с какими данными он работает. Во избежание недопонимания и разного вида путаницы сгруппируем классы по источникам, с которыми они работают:

– файлы;

– именованный канал;

– сетевые соединения;

– буфер памяти (например, массивы);

– System.in, System.out, System.error.

IO Потоки являются ключевым понятием в Java IO. Поток представляет собой концептуально бесконечный поток данных. Вы можете либо прочитать из потока или записать в поток. Поток подключен к источнику данных или адресату вывода. Потоки в Java IO работают с байтами (чтение и запись байтов) или с символами (чтение и запись символов).

Программа, которая считывает данные из какого-либо источника, должна реализовать классы InputStream или Reader. Программа, которая записывает данные в адресат вывода реализует OutputStream или Writer. Это проиллюстрировано ниже на рисунке 2.

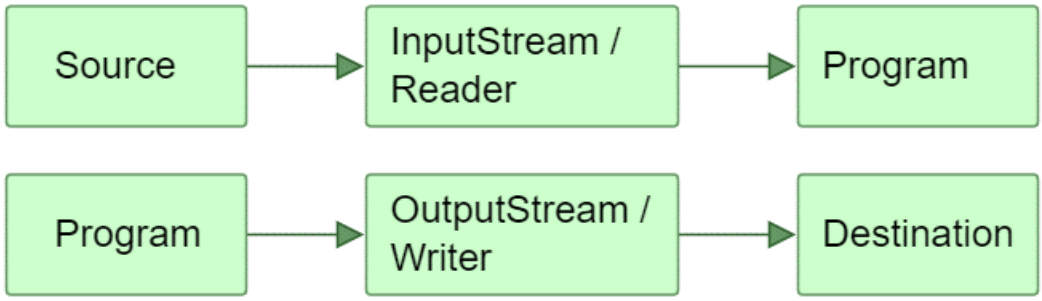


Рис. 2

InputStream и Reader связаны с источником данных. В свою очередь OutputStream и Writer связанны с адресатом вывода.

Java IO содержит множество подклассов InputStream, OutputStream, Reader, и Writer классов.

Цели, которые решают подклассы: доступ к файлам, доступ к сети, буфер обмена, межпотоковое взаимодействие (Pipes), буферизация, фильтрация, парсинг, чтение и запись текста (чтение / запись), чтение и запись данных примитивных данных (long, int и т. д.), чтение и запись объектов.

Эти цели позволяют быстрее ориентироваться при работе с Java IO. Они упрощают понимания данного пакета и в частности каждого класса и понять на что они ориентированы.

На основе выше изложенного материала отобразим в таблице практически полный перечень классов из пакета Java IO, разделив на ввод-вывод и основываясь с каким типом данных работаем (символы или байты), а также учитывая преследуемые конечные цели.

Такблица

| | Byte Based Input | Byte Based Output | Character Based Input | Character Based Output |
|---|---|---|---|---|
| Basic | InputStream | OutputStream | Reader InputStreamReader | Writer OutputStreamWriter |
| Arrays | ByteArrayInputStream | ByteArrayOutputStream | CharArrayReader | CharArrayWriter |
| Files | FilelnputStream RandomAccessFile | FileOutputStream RandomAccessFile | FileReader | FileWriter |
| Pipes | PipedlnputStream | PipedOutputStream | PipedReader | PipedWriter |
| Buffering | BufferedlnputStream | BufferedOutputStream | BufferedReader | BufferedWriter |
| Filtering | FilterlnputStream | FilterOutputStream | FilterReader | FilterWriter |
| Parsing | PushbacklnputStream StreamTokenizer | | PushbackReader LineNumberReader | |
| Strings | | | StringReader | StringWriter |
| Data | DatalnputStream | DataOutputStream | | |
| Data – Formatted | | PrintStream | | PrintWriter |
| Objects | ObjectlnputStream | ObjectOutputStream | | |
| Utilities | SequencelnputStream | | | |

Используя данную сводную таблицу, программист сможет мгновенно ориентироваться в системе ввода-вывода, а также сделать быстрый старт в изучении

данной области разработки, поскольку выше изложенная концепция является фундаментом для дальнейших исследований.

***Список литературы***

1. Package java.io [Электронный ресурс]. – Режим доступа: https://docs.oracle.com/javase/8/docs/api/java/io/package-summary.html