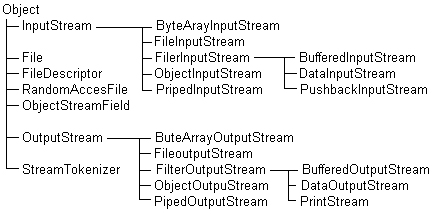
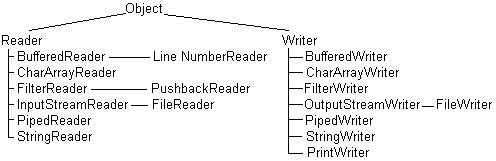
Потоки ввода / вывода в Java

Обобщенное понятие источника ввода относится к различным способам получения информации: к чтению дискового файла, символов с клавиатуры, либо получению данных из сети. Аналогично, под обобщенным понятием вывода также могут пониматься дисковые файлы, сетевое соединение и т.п.

Эти абстракции дают удобную возможность для работы с вводом-выводом (I/O), не требуя при этом, чтобы каждая часть кода понимала разницу между, скажем, клавиатурой и сетью. В Java эта абстракция называется потоком (stream) и реализована в нескольких классах пакета java.io.

Все программы Java автоматически импортируют пакет java.lang. Поэтому всегда можно воспользоваться предопределенными потоками: System.in, System.out, System.err.

Потоки в Java делятся на:

* **Байтовые**. Эти потоки предоставляют средства для ввода/вывода байтов. Используются при работе с бинарными данными. На рисунке ниже показана иерархия классов байтовых потоков  
  
* **Символьные**. Эти потоки предоставляют средства для ввода/вывода символов. Символьные потоки используют кодировку Unicode. На рисунке ниже показана иерархия классов символьных потоков.  
  

Ввод/вывод в Java можно разделить на

* **Буферизированный**.При буферизированном вводе/выводе данные сначала помещаются в буфер в оперативной памяти. Родное API **ввода** вызывается только тогда, когда буфер пустой. Родное API **вывода** вызывается только тогда, когда буфер полный.  
    
  Часто возникает потребность записать содержимое буфера в какие-то критические моменты, не ожидая, когда буфер заполнится. Этот процесс известен как **flushing** (сброс) буфера.  
    
  Некоторые буферизированные классы для вывода поддерживаю автоматический сброс буфера, определяемый опциональным аргументом в конструкторе. Например, класс PrintWriter сбрасывает буфер при каждом вызове println() или format().

Чтобы сбросить буфер вручную, нужно вызвать метод flush(). Этот метод определен для любого выходного потока, но выполняет работу только для буферизированного потока.

* **Не буферизированным**. При не буферизированном вводе выводе каждый запрос на чтение или запись означает взаимодействие с оборудованием (дисками, сетью, и т.п.) через соответствующую операционную систему. Это может привести к снижению производительности;

1. Thinking in Java Fourth Edition, Bruce Eckel, с. 324
2. Java 7- The Complete Reference - 8th Edition, Herbert Schildt, с. 318-320.
3. Buffered Streams, <https://docs.oracle.com/javase/tutorial/essential/io/buffers.html>