Java IO

- Байтовые:
  - InputStream
  - OutputStream
- Символьные:
  - Reader
  - Writer
- Потоки необходимо закрывать после использования
- try-with-resources закрывает поток автоматически
- После закрытия поток больше нельзя использовать
- `available()` показывает <= кол-во байт на чтение
- `flush()` - принудительная запись накопленных в буфере данных на физ. носитель
- Фильтрующие потоки
- BufferedInputStream – буферизация потоков. Ускоряет работу с файлами
- Конкатенация потоков
- Символьные потоки трансформируют байты в символы в зависимости от указанной кодировки
- Доступ по проиозвольному смещение `f.seek()`
- Java.io.File – доступ ко многим функциям файловой системы
- Удаление файла:
  - old way – `if file.exists() → file.delete();`
  - new way – через класс Path – `deleteIfExists(path);`
- FileVisitor – рекурсивный обход директории

Нетворкинг

- Сокеты – программная абстракция для:
  - установки соединения
  - отправки данных
  - получения данных
- Методы сокетов блокируют текущий поток исполнения
- Каналы и селекторы: → java.nio.ByteBuffer
  - Каналы позволяют работать с буфурами в асинхронном режиме
  - Селекторы позволяют проверять наличие событий в разных каналах