

# Вложенные классы

- класс определенный в блоке другого класса или метода
- Статические вложенные классы – не привязаны к экземпляру, имеют доступ только к static полям и методам класса
- Внутренние классы – привязаны к экземпляру и имеют доступ к его полям и методам. Не имеют доступа к static. Нужно указывать экземпляр при определении.
  - Локальные классы – определены внутри метода
  - Анонимные классы – одновременно определяются и создаются от интерфейса или базового класса. Применяются, если нужен локальный класс для одноразового использования

# Библиотека классов

- основные: java.math, java.time, java.text
- System:
  - `System.in`
  - `System.out`
  - `System.err`
- Завершение процесса: `System.exit(0)`
- Время:
  - `System.currentTimeMillis()`; - миллисекунды с 01.01.1970
  - `System.nanoTime()`; - относительно JVM точки отсчета
- `System.arraycopy(src, srcPos, dest, destPos, len);`
- `sout(Arrays.toString)`, иначе выведется ссылка
- `Arrays.binarySearch(array, 4)` – возвращает индекс элемента. Иначе - `insertIndex - 1` → `insertIndex = -returnedIndex - 1`
- `Arrays.copyOf(array, array.length + 1);` - копирует по значению и расширяет нулями, если надо
- Вставка элемента с сохранением сортировки: `binarySearch` → `copyOf`(расширяем массив) → `arraycopy` → передвигаем часть массива → `array[i] = key` (вставляем на месте образовавшегося пропуска наш ключ)

- Random
- Обертки (Integer, ...) → позволяют использовать методы базового класса, передаются по значению, как ссылочные типы. Это будут неизменяемые типы, также как String, и при изменении каждый раз будет создаваться новый объект.
- final
- кэширование значений → valueOf
- parseX
- метод equals позволяет определить сравнение по значению полей
- BigInteger – неизменяемый тип для больших целых чисел
- BigDecimal
- ~~double для денег~~

## Строки

- Интернирование – использование общего пула при совпадении значения
  - `String vegetable = new String("Carrot"); // not interned`
  - `String vegetable = "Carrot" // interned`
- StringBuilder – для оптимальной конкатенации строк
- split (по паттерну regex), join (как в питоне)
- регулярки – найти и заменить по шаблону

## Даты

- LocalDate, LocalTime, LocalDateTime
- DateTimeFormatter
- instant: - абсолютный момент во времени, LocalDateTime – локальный момент во времени

## Обработка ошибок

- `void foo() throws Exception` → `throw new Exception("Error");`
- обработка исключений → try catch

- finally – выполняется в конце
- interface AutoClosable → override close()
- try-with-resources – автоматически добавляет finally → close
- Типы исключений:
  - Throwable
    - Error – необрабатываемая ошибка виртуальной машины
    - Exception
      - RuntimeException – unchecked – облегченная версия обычных исключений
      - остальные – checked – обязательно throws
- handle or declare
- Stack trace
- ~~пустой catch-блок~~
- Кастомные исключения – можно написать свой класс extends Exception