# 0. Intro

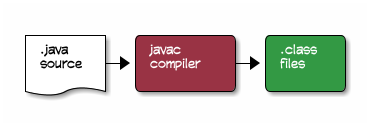
С третьего урока начнутся более практические задачи, первые 2 - основа и по большей части теория.

# 1. Java, Android, Git.

**Все перед уроком должны прочитать** <http://developer.alexanderklimov.ru/android/java/java.php>

## Java VM

Java - язык для кроссплатформенной разработки. Код на языке java компилируется в файлы \*.class, которые содержат в себе Java байт-код.



Данный байт-код выполняется в виртуальной машине JAVA. Именно с помощью виртуальной машины программы, написанные на Java являются кроссплатформенными. Это было одним из факторов, почему для основного языка программирования для Android была выбрана именно Java. Это позволило Android не зависеть от архитектуры процессора и работать на огромном количестве устройств - от телефонов до холодильников.

## Основы Java и программирования в целом

Объекты создаются динамически с помощью конструкции new, среда исполнения отслеживает наличие ссылок на них, а [сборщик мусора](https://ru.wikipedia.org/wiki/%D0%A1%D0%B1%D0%BE%D1%80%D0%BA%D0%B0_%D0%BC%D1%83%D1%81%D0%BE%D1%80%D0%B0_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)) периодически очищает память от объектов, ссылок на которые нет. Важно помнить о том, что стоит избегать создания большого количества объектов, иногда это может заметно повлиять на быстродействие программы.

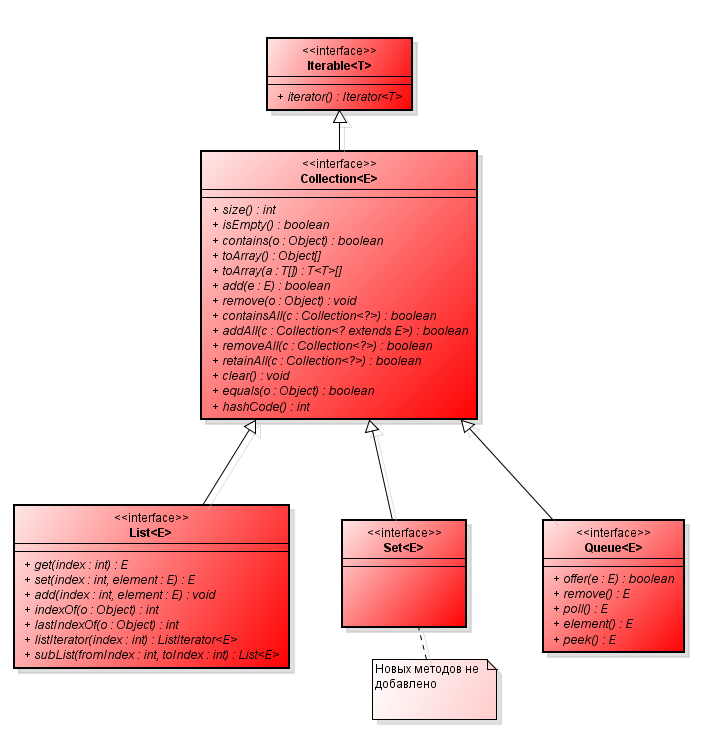
**Все перед уроком должны прочитать** <http://developer.alexanderklimov.ru/android/java/java.php>

Рассказать, почему важно соблюдать стиль кода:

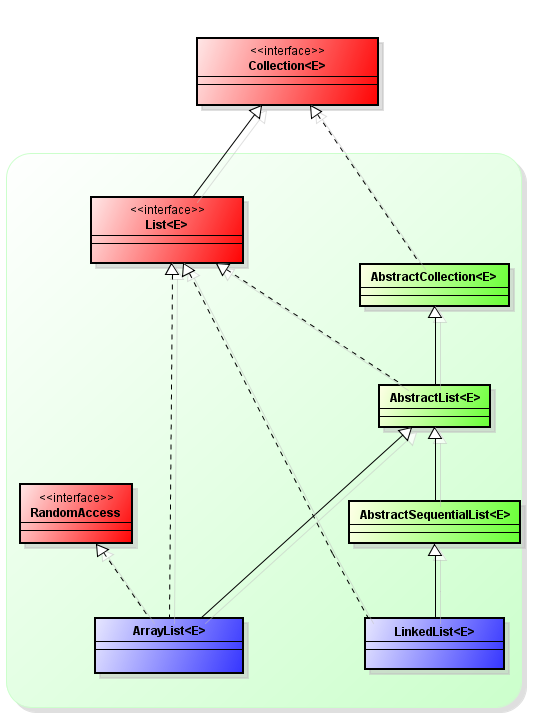
1. Google java code style <https://google.github.io/styleguide/javaguide.html>
2. проще читать
3. важнейшее значение имеет сопровождение кода, т.к. код чаще всего пишется не раз и навсегда.
4. Важность комментариев

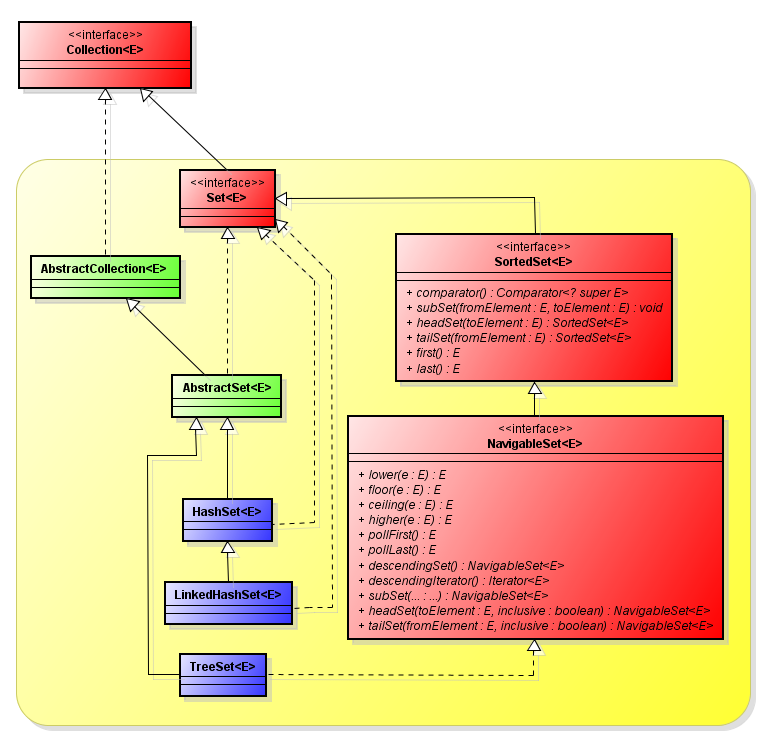
## Коллекции

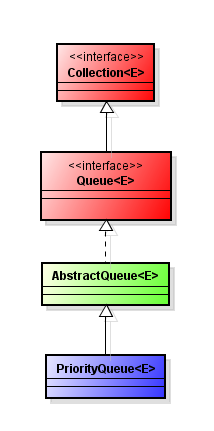
<http://www.quizful.net/post/Java-Collections>

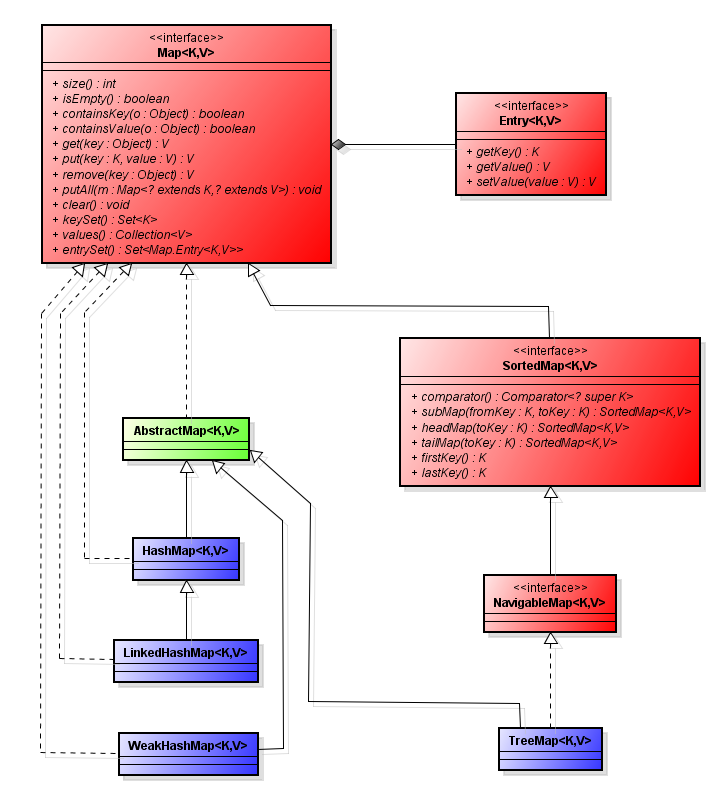


**Итератор** - это паттерн позволяющий получить доступ к элементам любой коллекции без вникания в суть ее реализации.

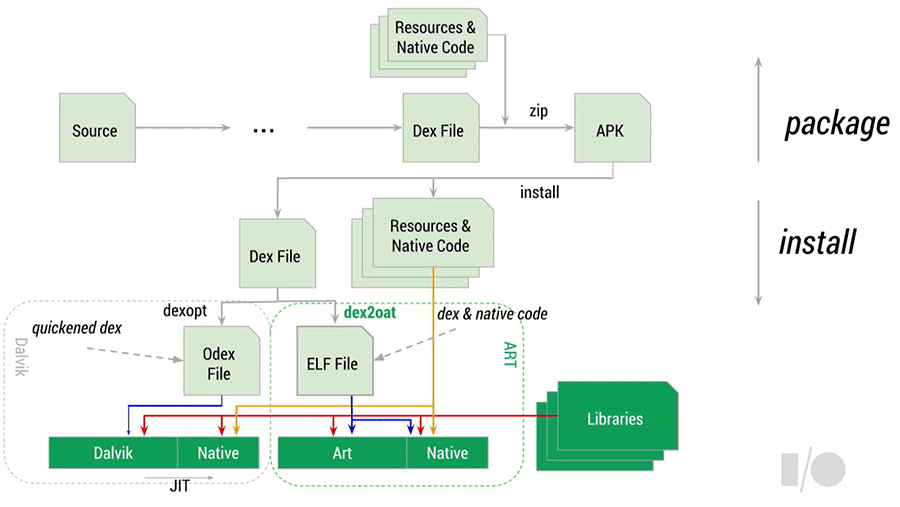








## Dalvik vs ART (c API 19 aka Kitkat)



JIT

AOT

Далвик меньше весит, но медленнее выполняется.

## Android studio

Запуск.  
Рассказ про минимальную версию сдк, с какой мы работаем. 4.1 - боль  
Gradle, groovy, основные блоки.

Создание пустого проекта.  
Структура проекта, рассказать про основные типы ресурсов.

Кнопка, текстовое поле, поле ввода.

Запуск нового активити

## Git

Про .gitignore

/captures

.externalNativeBuild

.gradle

/local.properties

/.idea/workspace.xml

.DS\_Store

/build

.idea/

\*iml

\*.iml

\*/build

release\_keystore.properties

gp\_publisher\_keys.json

Про ssh <https://help.github.com/articles/connecting-to-github-with-ssh/>

git init

git add --all

git commit -m "Initial commit"

git remote add origin git@github.com:Tooto/android-school.git  
git push -u origin master

<https://git-scm.com/book/en/v2>

<https://githowto.com/ru>

Source Tree

Про правильный порядок ведения проекта, git glow <https://danielkummer.github.io/git-flow-cheatsheet/index.ru_RU.html>

##### Полезные ссылки:

1. Минимум, который необходимо знать - <http://developer.alexanderklimov.ru/android/java/java.php>
2. <https://developer.android.com/guide/components/fundamentals.html>
3. Философия Java (Thinking Java) от Брюса Эккеля
4. Java. Полное руководство автора Герберта Шилдта
5. Google java code style <https://google.github.io/styleguide/javaguide.html>
6. Коллекции <http://www.quizful.net/post/Java-Collections>
7. Git <https://git-scm.com/book/en/v2>, <https://githowto.com/ru>
8. Про ssh <https://help.github.com/articles/connecting-to-github-with-ssh/>
9. Gitflow <https://danielkummer.github.io/git-flow-cheatsheet/index.ru_RU.html>

##### Задание:

**Задание 1 (обязательное)**

Создать активити, содержащую 2 кнопки, открывающие два разных экрана.

Активити 1 будет содержать:  
1. Поле ввода на всю ширину экрана (EditText)

2. 2 кнопки действия (Button)  
3. Текстовое поле, в которое будет выводиться информация. (textView)  
  
Что сделать?  
1. Нужно написать журнал учеников.

2. В поле ввода (EditText) вводятся имена людей, при нажатии на кнопку 1 (Button) они сохраняются в массив.  
3. Кнопка 2 выводит список учеников по алфавиту в поле вывода (TextView). Каждое имя должно выводиться с новой строки.  
4. Модифицируем так, чтобы повторяющиеся имена не допускались (TreeSet).

**Задание 2 (необязательное)**

Экран должен содержать:  
1. EditText для ввода информации по ученикам

2. Button - одну кнопку  
3.TextView - для вывода информации по ученикам

В поле ввода через пробел вводятся:

Имя Фамилия Класс Год\_рождения  
Пример: Игорь Глушков 1Г 1991  
  
**При нажатии на enter на цифровой клавиатуре** из строки ввода формируется объект Student c полями (<https://stackoverflow.com/questions/3481828/how-to-split-a-string-in-java> )

name

surname

grade

birthdayYear.  
  
Пример обработки нажатия на enter:

final EditText edittext = (EditText) findViewById(R.id.edittext); edittext.setOnKeyListener(new OnKeyListener()

{ public boolean onKey(View v, int keyCode, KeyEvent event) { // If the event is a key-down event on the "enter" button if ((event.getAction() == KeyEvent.ACTION\_DOWN) && (keyCode == KeyEvent.KEYCODE\_ENTER))

{ // Perform action on key press Toast.makeText(HelloFormStuff.this, edittext.getText(), Toast.LENGTH\_SHORT).show();

return true; }

return false; } });

Айди студента формируется случайным образом (можно использовать System.currentTimeMillis() )  
Далее студент помещается в HashMap.

Кнопка действия печатает все данные учеников, включая айди в текстовое поле вывода информации