

Платформа: история и окружение

Иван Игоревич Овчинников

GeekBrains. Java Core.

2022

Введение и знакомство (о себе)

Иван Овчинников. НПО ИТ, РКС, GB.

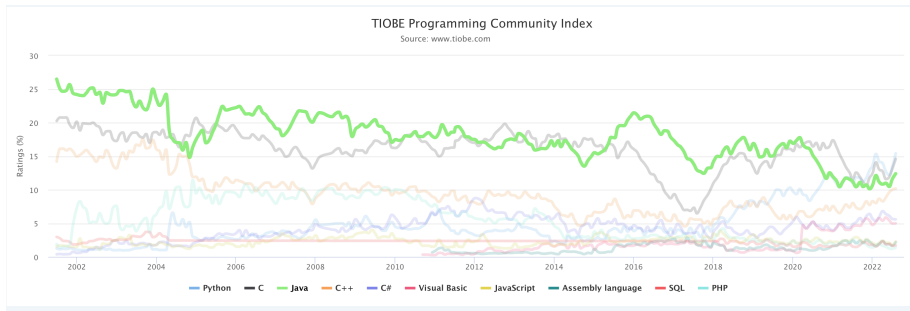
Используемые технологии

Много и с удовольствием C, C++, Java, Verilog. C# и Python даже не считаем.

Для GB более 20 потоков по разным направлениям, более двух тысяч студентов.

и об уроке

Краткая история



Почему Java?

Написано однажды, работает везде.

Базовый инструментарий

- 1 Eclipse
- 2 NetBeans
- 3 IntelliJ IDEA
- 4 BlueJ
- 5 Oracle JDeveloper
- 6 MyEclipse
- 7 Greenfoot
- 8 jGRASP
- 9 JCreator
- 10 DrJava

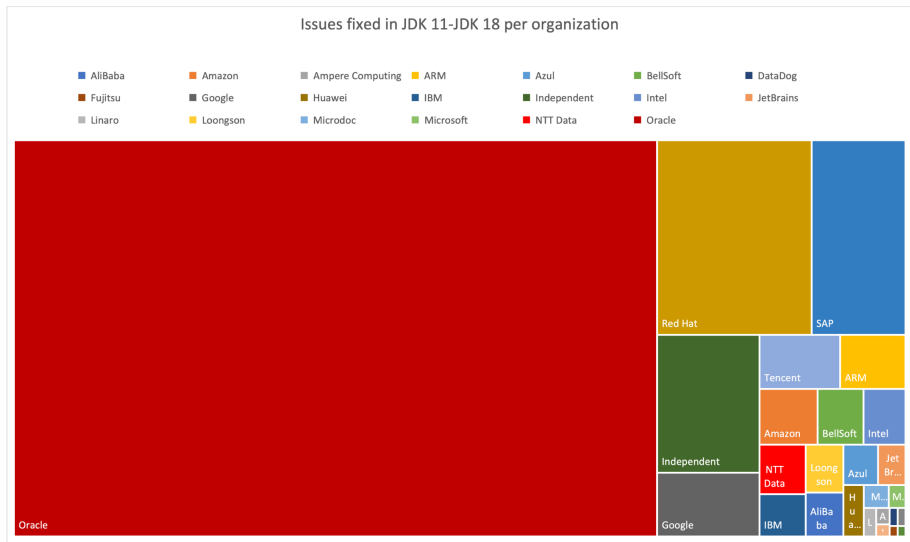
Что нужно скачать

- 1 ждк
- 2 среду

Что нужно скачать

- 1 Oracle JDK
- 2 OpenJDK by Oracle
- 3 Liberica JDK
- 4 экзотические
 - GOST Java
 - AdoptOpenJDK
 - Red Hat OpenJDK
 - Azul Zulu
 - Amazon Corretto

Кто делает для нас JDK



Когда нужно по-быстрому



Переменные среды

- PATH
- JAVA_HOME
- JRE_HOME
- J2SDKDIR
- J2REDIR

TL;DR:

- $JDK = JRE + \text{инструменты разработчика};$
- $JRE = JVM + \text{библиотеки классов};$
- $JVM = \text{Native API} + \text{механизм исполнения} + \text{управление памятью}.$

JVM и что в нём происходит

рис 1.2 (стр 4) конспекта

Структура проекта

- простейшие (один файл)
- обычные (несколько пакетов)
- шаблонные (формируются сборщиками)
- скриптовые (jupyter notebook)

Простейший проект

Думаю, тут надо вживую покодить, благо немного

```
public class Main
public static void main(String[] args)
System.out.println("Hello, world!");
```

```
ivan@gb src > javac Main.java
ivan@gb src > java Main
Hello, world!
```

также вживую открыть юпитер ноутбук, показать, что возможно запускать код в скриптовом формате

- пакеты,
- классы,
- метод main,
- комментарии
- ресурсы

разделить на два слайда сначала пакет с пакетами потом стоящие
стройными рядами коробки или контейнеры

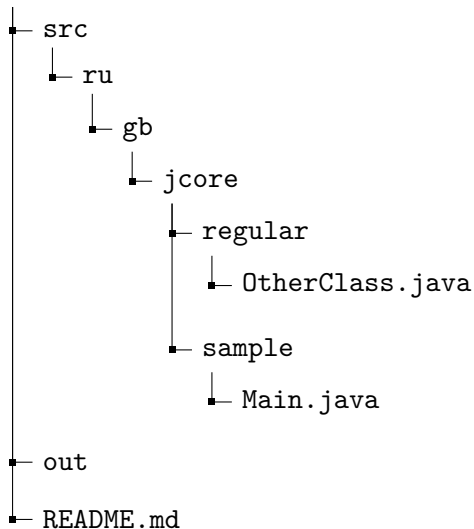
разделить на два слайда сначала фото пустого школьного класса, потом

разделить на два слайда псвм показать точку входа как вход в пещеру

разделить на два слайда можно как в редакции «комменты и лайки туда»
и слайд с типами комментариев

Структура простого проекта

Sample



показать папку с простейшим проектом из одного файла,
скомпилировать, запустить.

тут, скорее всего, без лайв кода не обойтись, нужно открыть терминал и показать папку проекта, структуру проекта и содержимое файлов

Пример автоматически сгенерированной документации

OVERVIEW PACKAGE **CLASS** TREE DEPRECATED INDEX HELP

ALL CLASSES

SEARCH:

SUMMARY: NESTED | FIELD | CONSTR | METHOD DETAIL: FIELD | CONSTR | METHOD

Package ru.gb.jcore.regular

Class OtherClass

java.lang.Object
ru.gb.jcore.regular.OtherClass

```
public class OtherClass
extends java.lang.Object
```

Другой, очень полезный класс приложения. Здесь мы можем описать его основное назначение и способы взаимодействия с ним.

основные ключи и аргументы

- `-d docs`
- `-sourcepath src`
- `-cp out`
- `-subpackages`
- `ru`

куда же без особенностей работы с Windows?

- `-locale ru_RU`
- `-encoding utf-8`
- `-docencoding cp1251`

Автоматизация сборки в CLI

makefile

Автоматизация сборки в CLI

```
SRCDIR := src OUTDIR := out
JC := javac JCFLAGS := -sourcepath .(SRCDIR)/ -d (OUTDIR)
MAINSOURCE := ru/gb/dj/Main MAINCLASS := ru.gb.dj.Main
all: (JC) (JCFLAGS) SRCDIR/MAINSOURCE.java
clean: rm -R (OUTDIR)
run: cd out && java MAINCLASS
```

docker

Автоматизация сборки в CLI, контейнеризация

```
# syntax=docker/dockerfile:1
FROM bellsoft/liberica-openjdk-alpine:11.0.16.1-1
# from where (related to Dockerfile) to where COPY ./src ./src
# remember that you are root and in / directory RUN mkdir ./out
# be sure to use relative paths RUN javac -sourcepath ./src -d out
./src/ru/gb/dj/Main.java
# what will we do on a container start CMD java -classpath ./out ru.gb.dj.Main
```

что изучили и домашка