

Язык Java

Глава I. Введение в Java

Виталий Витальевич Перевощиков

Осенний семестр 2021

Содержание

1. Кратко об истории Java
2. Рейтинги популярности
3. Области применения Java
4. Достоинства и недостатки Java
5. Java Development Kit (JDK)
6. Интегрированные среды разработки (IDE)
7. Проект Hello, World
8. Компиляция и выполнение кода на Java
9. Управление памятью в Java
10. Пакеты Java

Кратко об истории

- 1991 г.: начало создания языка
- 1996 г.: Sun Microsystems выпустила релиз Java 1.0
- 2007 г.: Релиз исходного кода виртуальной машины Java
- 2010 г.: Новый владелец: корпорация Oracle














Индекс TIOBE

- **Индекс TIOBE** определяется на основе подсчета результатов поисковых запросов на популярных порталах:
 - Google, Wikipedia, Yahoo!, Youtube, ...
- Данные на август 2021

Aug 2021	Aug 2020	Change	Programming Language		Ratings	Change
1	1		 C		12.57%	-4.41%
2	3	▲	 Python		11.86%	+2.17%
3	2	▼	 Java		10.43%	-4.00%
4	4		 C++		7.36%	+0.52%
5	5		 C#		5.14%	+0.46%

Рейтинг IEEE Spectrum

- Рейтинг журнала **IEEE Spectrum** подсчитывается на основе 11 метрик из 8 источников:
 - Google Search, Google Trends, Github, Twitter, Stack Overflow, Reddit, Hacker News, CareerBuilder
- Данные на август 2021

Language Ranking: IEEE Spectrum						
Rank	Language	Type		Score		
1	Python				100.0	
2	Java				95.3	
3	C					94.6
4	C++					87.0
5	JavaScript					79.5

Примеры компаний

Источник stackshare.io



Uber



Airbnb



Google



Netflix



Pinterest



Instagram



Spotify



Amazon



Slack

Области применения Java

- Веб-приложения
 - Технологии: [Spring](#), Jakarta Servlet, JSP (Java Server Pages), JSF (Java Server Faces)
- Мобильные приложения
 - Технологии: Java ME (Java Micro Edition), Android SDK
- Настольные приложения
 - Технологии: AWT (Abstract Window Toolkit), Swing, JavaFX
- Научные приложения (Matlab)
- Веб-сервера (Apache Tomcat, JBOSS, Websphere)
- ...

Достоинства Java

- Простота в использовании
- Объектная ориентированность
- Платформенная независимость
 - "Write once, run anywhere"
- Безопасность
- Автоматическое управление памятью
 - Сборщик мусора
- Большое количество библиотек

Недостатки Java

- Более низкая производительность
- Высокое потребление памяти
- Затраты на "железо"
- Отсутствие контроля над памятью

Java Development Kit (JDK)

- JDK - комплект разработчика приложений
- Компоненты JDK:
 - Компилятор `javac`
 - Среда выполнения JRE (Java Runtime Environment):
 - Виртуальная машина JVM (Java Virtual Machine)
 - Стандартная библиотека классов (Java API)

- **OpenJDK** - версия JDK, состоящая из исключительно свободного и открытого исходного кода (open source)
- Последняя версия: **16**
- Ранние версии:
 - 14
 - 11 (LTS = Long Term Support, долгосрочная поддержка ПО)
 - 8

Интегрированная среда разработки (IDE)

Основные функции

- Редактор исходного кода
 - Поиск кода
 - Автогенерация кода
 - Автодополнение кода(IntelliSense)
 - Рефакторинг
- Автоматизация сборки проекта
- Отладка (debugging)
- Поддержка контроля версий (Git, Subversion, ...)

Популярные IDE

- IntelliJ IDEA
- Eclipse
- NetBeans

В этом курсе мы будем работать с IntelliJ IDEA.

- Страница для скачивания
- Скачать и установить бесплатную Community-версию (CE)

Скачать IntelliJ IDEA

Windows

macOS

Linux

Ultimate

Для корпоративной и веб-разработки

Скачать

.dmg (Intel) ▼

Бесплатная пробная версия на 30 дней

Community

Для JVM и Android-разработки

Скачать

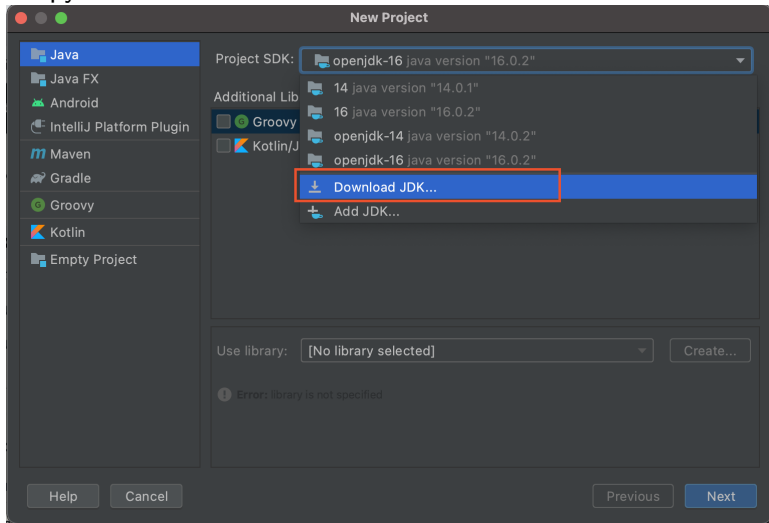
.dmg (Intel) ▼

Бесплатная, на базе открытого исходного кода

- **Важно:** IntelliJ IDEA не включает в себя JDK

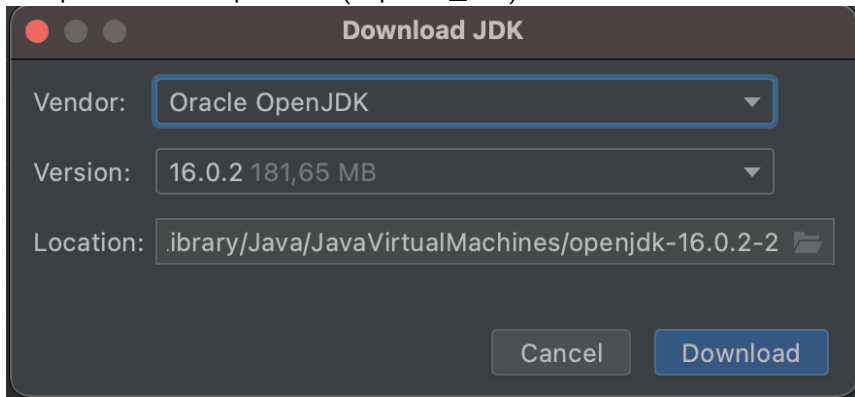
Новый проект в IntelliJ IDEA: часть 1

- Создать новый проект в IntelliJ IDEA
- Загрузить JDK



Новый проект в IntelliJ IDEA: часть 2

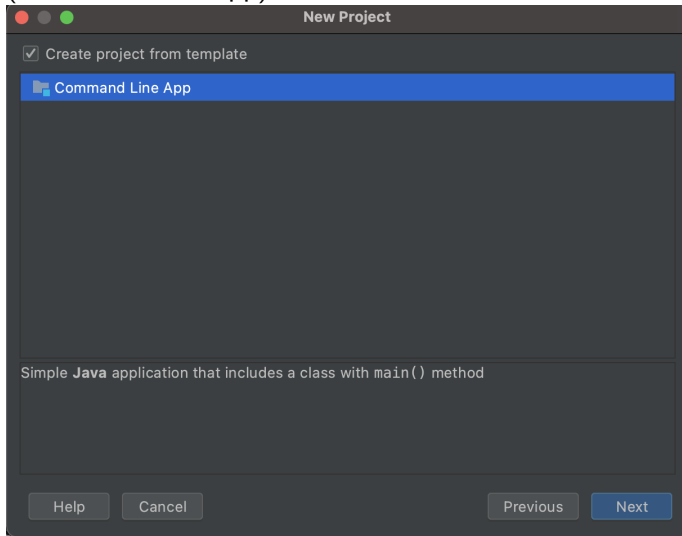
- Выбрать Oracle OpenJDK (версия ≥ 16) и нажать "Download"



- Выбрать загруженную версию OpenJDK и нажать "Next"

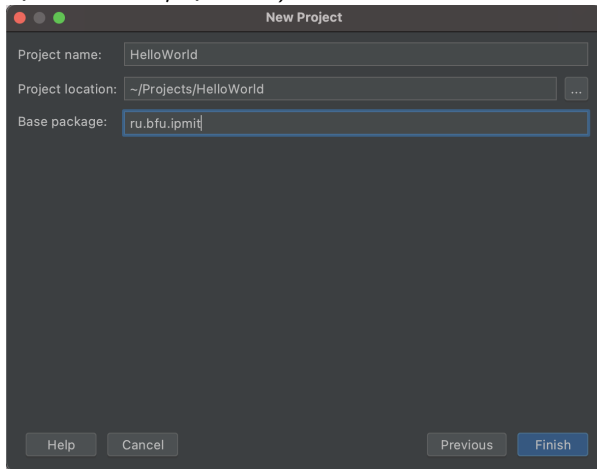
Новый проект в IntelliJ IDEA: часть 3

- Шаблон для проекта: приложение командной строки (Command Line App)



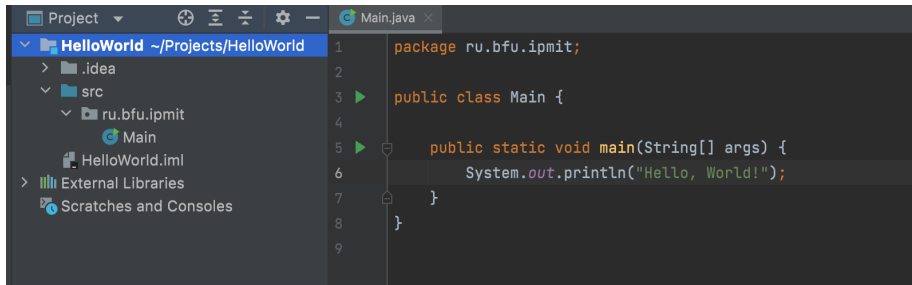
Новый проект в IntelliJ IDEA: часть 4

- Выбрать имя проекта и имя базового пакета (идентификатор организации/проекта)



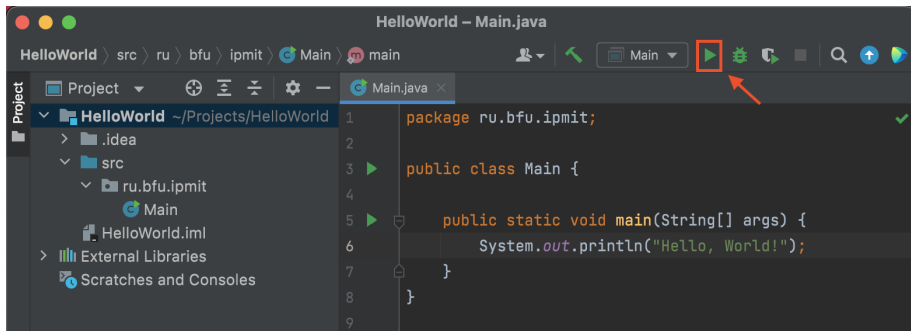
- Нажать "Finish"

Структура проекта "Hello, World!"

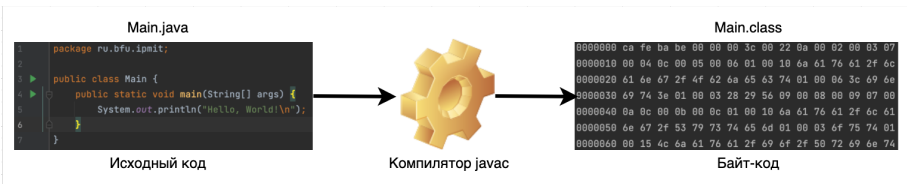


- src - директория с исходным кодом
- Главный класс - Main, расположен в файле src/ru/bfu/ipmit/Main.java
- Точка входа - метод main: публичный, статический

Запуск программы "Hello, World!"



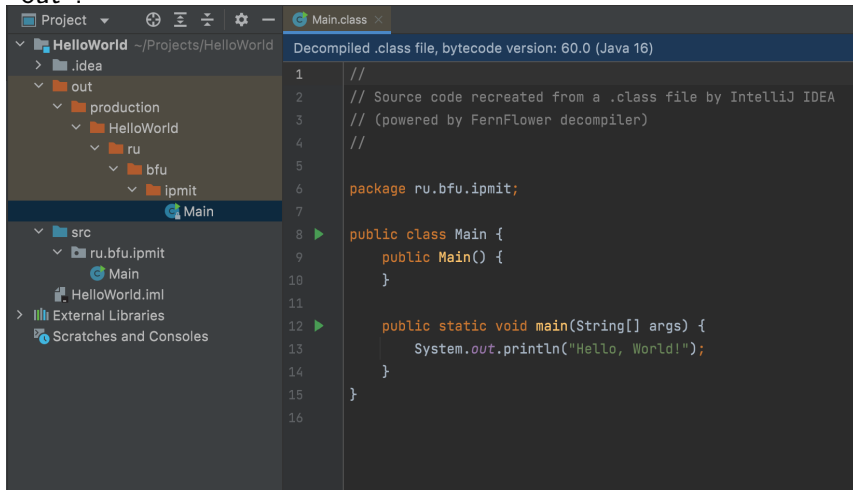
Компиляция Java-кода



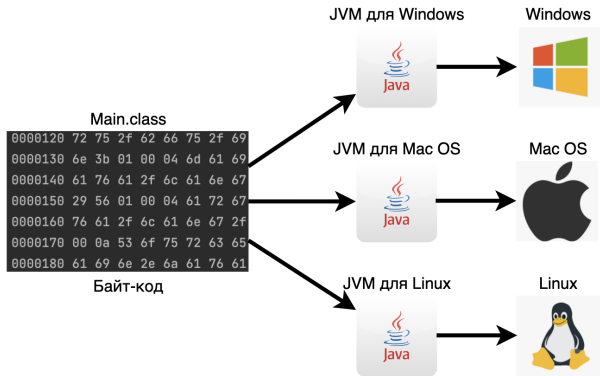
- Байт-код не зависит от платформы (ОС, архитектуры процессора)
- Байт-код не является машинным кодом конкретной платформы
- Байт-код - набор инструкций для виртуальной машины JVM

Пример: Компиляция "Hello, World!"

- Сборка проекта (Build → Build Project)
- Скомпилированные .class-файлы с байт-кодом находятся в папке "out":



Выполнение байт-кода на JVM



- JVM реализуется для каждой платформы отдельно
- JVM преобразует байт-код в машинный код конкретной платформы
 - Используем JIT-компиляция (Just-in-Time)
 - JIT ускоряет выполнение байт-кода

Управление памятью в JVM

C++

```
void method(int largeSize) {  
    // выделение динамической памяти  
    int *buffer = new int[largeSize];  
  
    doSomeWork(buffer);  
  
    // освобождение памяти  
    delete[] buffer;  
}
```

Java

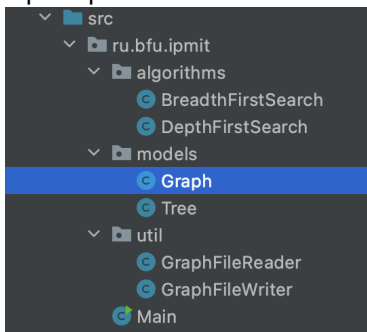
```
void method(int largeSize) {  
    // выделение динамической памяти  
    int[] buffer = new int[largeSize];  
  
    doSomeWork(buffer);  
}
```

- Освобождением памяти в JVM занимается "сборщик мусора" (Garbage Collector)
- **Осторожно!** Сборщик мусора не гарантирует отсутствие утечек памяти

Пакеты Java

- Пакет (package) - группа связанных по смыслу классов и подпакетов (subpackages).

Пример:



Базовый пакет: `ru.bfu.ipmit`:

- Класс `Main`
- Подпакет `ru.bfu.ipmit.algorithms`
- Подпакет `ru.bfu.ipmit.models`
- Подпакет `ru.bfu.ipmit.util`

- Полное имя класса: `<имя пакета>.<имя класса>`
 - Пример: `ru.bfu.ipmit.models.Graph` для класса `Graph`

Импорт классов

Вариант 1: ключевое слово `import`:

```
1 package ru.bfu.ipmit;
2
3 import ru.bfu.ipmit.algorithms.BreadthFirstSearch;
4 import ru.bfu.ipmit.models.Graph;
5 import ru.bfu.ipmit.util.GraphFileReader;
6
7 public class Main {
8
9     public static void main(String[] args) {
10         Graph graph;
11         BreadthFirstSearch search;
12         GraphFileReader reader;
13
14         // Чтение графа из файла и выполнение поиска в ширину
15         // ...
16     }
17
18 }
```

Вариант 2: Полное имя класса:

```
1 package ru.bfu.ipmit;
2
3 public class Main {
4
5     public static void main(String[] args) {
6         ru.bfu.ipmit.models.Graph graph;
7         ru.bfu.ipmit.algorithms.BreadthFirstSearch search;
8         ru.bfu.ipmit.util.GraphFileReader reader;
9
10        // Чтение графа из файла и выполнение поиска в ширину
11        // ...
12    }
13
14 }
```

Импорт пакетов

- `import <имя пакета>.*`
- Пример:

A screenshot of a Java IDE window titled "Main.java". The code is as follows:

```
1 package ru.bfu.ipmit;
2
3 import ru.bfu.ipmit.models.*;
4
5 public class Main {
6
7     public static void main(String[] args) {
8         Tree tree;
9         Graph graph;
10    }
11
12 }
```

- **Вопрос:** Будут ли импортированы подпакеты?
- **Рекомендация:** избегайте импорта пакетов и используйте импорт отдельных классов.

Виды пакетов

- Встроенные в JDK (Java API)

Пакет	Содержание
<code>java.lang</code>	Фундаментальные классы: Object, String, Math, System, ...
<code>java.util</code>	Collection API: List, Map, Tree, ...
<code>java.text</code>	Работа с текстом
<code>java.net</code>	Работа с сетевыми протоколами (HTTP, TCP/IP, ...)
<code>java.sql</code>	Работа с SQL базами данных

- Пользовательские

- JAR (Java Archive) - ZIP-архив, содержащий файлы с классами и метаданные
- JAR-файл используется для импорта пользовательских Java-пакетов
- Подробнее о работе с JAR: в практическом задании