

# lesson01.HelloWorld.java — первая программа на Java — введение в Java 001

---

## Установка JDK

В интернете существуют тысячи инструкций по установке окружения для работы в Java.

Я уверен, что вы сможете вбить в Google запрос на английском: "How to Install Java Development Kit" или на русском: "Установка JDK". Java принадлежит фирме Oracle и можно сразу скачать Java с её страниц, скачав просто JDK, скачав Netbeans вместе с JDK или скачав открытую AdoptOpenJDK (предпочтительнее для новичков).

## Java Development Kit

JDK — это программа, которая будет переводить написанный вами код в понятный для машин. На вашей машине может быть установлено несколько разных версий Java. Одной из них должна быть Java 8.x. Это та версия, на которой базируются все современные версии, и для изучения основ языка и ООП именно оно нам и "нада".

## Prebuilt OpenJDK Binaries for Free!

Java™ is the world's leading programming language and platform. AdoptOpenJDK uses [infrastructure](#), [build](#) and [test](#) scripts to produce prebuilt binaries from [OpenJDK™](#) class libraries and a choice of either the [OpenJDK HotSpot](#) or [Eclipse OpenJ9](#) VM. All AdoptOpenJDK binaries and scripts are [open source licensed](#) and available for free.

### Download for Windows x64

#### 1. Choose a Version

- ☒ OpenJDK 8 (LTS)
- ☐ OpenJDK 11 (LTS)
- ☐ OpenJDK 13 (Latest)

#### 2. Choose a JVM

[Help Me Choose](#)

- ☒ HotSpot
- ☐ OpenJ9



Latest release

jdk8u242-b08

Other platforms ↻

Release Archive & Nightly Builds 📦

AdoptOpenJDK now also distributes OpenJDK upstream builds! (Built by Red Hat)

Скачать JDK можно с Oracle:

Linux x64 RPM Package	56.53 MB	jdk-8u241-linux-x64-demos.rpm
Linux x64 Compressed Archive	56.36 MB	jdk-8u241-linux-x64-demos.tar.gz
macOS	56.55 MB	jdk-8u241-macosx-x86_64-demos.zip
Solaris SPARC 64-bit	12.23 MB	jdk-8u241-solaris-sparcv9-demos.tar.Z
Solaris SPARC 64-bit	8.47 MB	jdk-8u241-solaris-sparcv9-demos.tar.gz
Solaris x64	12.21 MB	jdk-8u241-solaris-x64-demos.tar.Z
Solaris x64	8.44 MB	jdk-8u241-solaris-x64-demos.tar.gz
Windows x86	57.26 MB	jdk-8u241-windows-i586-demos.zip
Windows x64	57.28 MB	jdk-8u241-windows-x64-demos.zip

Уверен, вы справитесь с установкой этой программы самостоятельно или с помощью Google.

## Работа с Java

Работа с Java "по старинке".

Во всех языках программирования стало стандартом первым уроком выводить на экране строчки "lesson01.HelloWorld!".

Кто я такой, что бы идти против правил? Сегодня мы напишем нашу первую программу и выведем на экран "Привет Мир!".

Мы создаём файл в редакторе и сохраняем его под названием lesson01.HelloWorld.java

```
/**
 * Created by www.andron13.de on 27.12.2016.
 */
public class lesson01.HelloWorld { // Начало нашего класса lesson01.HelloWorld
    public static void main(String[] args) { // Начинаем наш метод, главный метод (мэйн)
        System.out.print("Привет ");
        System.out.println("Мир! :-");
        System.out.println("\t Hello World!!!");
    } // Закрываем главный метод
} // Закрываем класс
```


Сразу обращаем внимание на два слеша (косая черта), которыми в конце строки можно отделить пометки в тексте. Также комментировать код можно, выделив его с двух сторон звёздочкой и слешем вот так: /\* комментарий тут \*/. Комментарии в коде очень важны. Уже пару дней спустя не всегда очевидны некоторые решения и названия переменных. В иной программе может быть с десяток классов и в каждом — по несколько переменных. И все они что-то там считают, куда-то

там что-то передают и зачем-то все вот прям здесь, и "кто это вообще написал". Не стесняйтесь комментировать свой код и свои решения. Привычка писать "понятный" код поможет вам и в учёбе, и в работе.

Запускаем терминал или командную строку под Windows: Старт-> Выполнить-> CMD. Вместо **CMD** можно ввести **powershell** или **bash**. Это всё программы для работы с компьютером напрямую. Их называют терминалами, консолью или по собственному названию.

Компилируем и запускаем код:

```
javac lesson01.HelloWorld.java
// или
javac -encoding utf8 lesson01.HelloWorld.java
// и запускаем с помощью
java lesson01.HelloWorld"
```

 Выбрать Windows PowerShell

```
PS C:\Users\ptcho\Projects\teaching\MyJavaLessons\src> javac HelloWorld.java
HelloWorld.java:6: error: unmappable character for encoding Cp1251
    System.out.print("\t 1? 111111C 1Б1В1А1Д1З1||1  :-)");
                        ^
1 error
PS C:\Users\ptcho\Projects\teaching\MyJavaLessons\src> javac -encoding utf8 HelloWorld.java
PS C:\Users\ptcho\Projects\teaching\MyJavaLessons\src> java HelloWorld
    Привет Мир! :-)
    Hello World!!!
    И ещё строчка :-)Это моя первая программа
Ура
- Mein erstes Programm :-)
PS C:\Users\ptcho\Projects\teaching\MyJavaLessons\src>
```

С помощью **javac** мы создаём из нашей программы код, понятный машине. Не считаю нужным углубляться в детали этого процесса, но и ничто не мешает вам почитать о:

- байт-коде,
- компиляторе,
- JVM,
- SDK,
- JDK

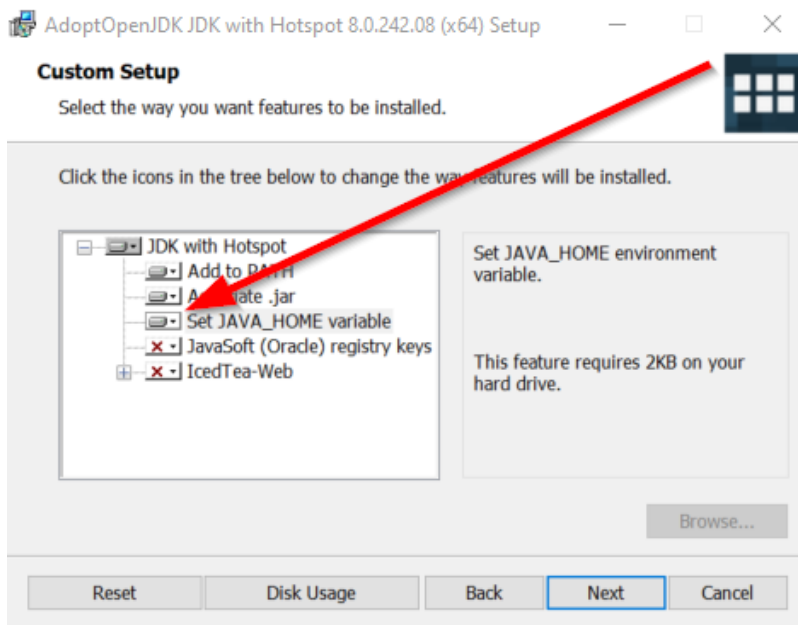
на других полезных ресурсах. История JVM (Java Virtual Machine) очень интересна, ведь ей уже более 20 лет.

При запуске мы увидим все системные сообщения на экране. Изменяя параметры программы, мы изменяем результаты. Это и есть программирование. И мы только что написали нашу первую программу.

Если у вас не прошла компиляция, как у меня, попробуйте скомпилировать с помощью учёта кодировки. Кириллица до сих пор не всегда хорошо работает в терминалах Windows. Если у вас не прошла компиляция вообще, то проверьте параметры настроек Java под Windows.

Не забывайте, что вы должны находиться в той папке, которая внесена в настройках в рабочие [Java\\_Home](#).

В AdoptOpenJDK можно при инсталляции пакета указать на установку Java\_Home.



Если у вас всё получилось — поздравляю! Вы только что поняли, как именно работали первые Java-кодеры.

Если у вас не всё получилось — не расстраивайтесь. В следующих уроках я подробнее расскажу, как удобнее работать с программами без командной строки и ручной компиляции. Вы можете спокойно переходить ко второму и третьему уроку. С ними всё встанет на свои места.

## Среда разработки Java — выбираем IDE программирования

Среда разработки Java: Android Studio, IntelliJ IDEA, NetBeans IDE, Eclipse Java Neon...

Первую программу на Java я написал в обычном эдиторе на Solaris. Потом компилировал "ручками". И запускал тоже ручками, примерно так, как было описано выше. И радовался тогда, очень радовался, если компилятор не выдавал ошибки и, уж тем более, испытывал настоящий восторг, если приложение делало, то что я и хотел. Позже я помню дотошную настройку под Windows 95. С внесением рабочей Java-папки в переменную среду пользователя. Поверьте на слово, порой я настраивал эти переменные среды часами.

Было много чего испробовано:

- UltraEdit
- Notepad и Notepad++
- PHP Expert Editor
- Visual Studio Code
- Sublime Text
- и многие другие

Но время редакторов и даже очень хороших редакторов прошло, и если для очень быстрых вещей, особенно за чужим компьютером, я с удовольствием использую портативный "Notepad++" с флешки, то дома я работаю в другой среде. Не в редакторе. А именно **среде**. И с объяснения термина "**интегрированная среда разработки**" мы и начнём.

## Интегрированная среда разработки - IDE

Программа для написания кода должна упрощать жизнь программиста и ускорять его работу, например:

- "подсвечивать" нужные фрагменты, скобки, переменные и т.д. и давать возможность настроить или отключить часть подсветки;
- подсказывать или "автозаполнять" код программисту, угадывать, если хотите, что именно он хочет написать;
- возможность менять дизайн отображения кода;
- быстро и чётко показывать результат;
- быстро переключаться между проектами;
- "самостоятельно" контролировать версии (например закидывать всё на GIT);

- помогать найти ошибки в коде и, тем более, синтаксические и орфографические ошибки;
- и многое другое.

Понятно, что всё это не про Notepad и даже не про Notepad++, хотя с подсветкой справляются многие "мелкие" редакторы. На рынке сейчас есть несколько крупных программ, которые заявляют себя как JAVA-IDE: это Eclipse (Eclipse Java Neon), IntelliJ IDEA (от JetBrains), NetBeans IDE (от Oracle), Android Studio (от Google & JetBrains).

Давайте я скажу по "два слова" о каждой из них:

**Android Studio** основан на IntelliJ IDEA с добавлениями, которые нужны при работе с android приборами. И, на самом деле, это среда разработки под Android на Java. И, говоря о плюсах или минусах, мы всё равно обсуждаем IntelliJ IDEA.

**NetBeans IDE** — рабочая среда от владельцев Java. Можно программировать, есть подсказка, есть подсветка. С моей точки зрения, это очень тяжёлый Notepad++ со встроенным компилятором. Но, на самом деле, NetBeans совсем не так уж плох. И не надо забывать, что он **бесплатный** и предоставляется компанией, которой принадлежат права на Java.

**IntelliJ IDEA** — редактор, утверждающий, что это IDE для Java и всех JVM-языков. Вполне возможно, что это так. Есть бесплатная версия и платная расширенная версия для Java EE (Java Platform, Enterprise Edition). Думаю, что многим будет важно узнать, что программу делают русскоязычные ребята.

**Eclipse** — среда разработки для всего. Всех языков, всех платформ. Очень старая программа. Примерно треть всех разработчиков Java сидят в этой программе. Программа выросла из внутренних разработок фирмы IBM и, действительно, известна всем. На мой взгляд, главный конкурент любой среде разработки.

**Visual Studio Code** — изначально этот редактор не был описан. Однако, со временем, я всё чаще и чаще писал программы и документацию именно здесь. Мощный редактор с огромным комьюнити и огромнейшим количеством полезных плагинов для всех видов разработки.

Я использую на работе NetBeans IDE, а дома IntelliJ IDEA и, на самом деле, вы не всегда можете контролировать то, в какой среде вы будете работать. Бывают ситуации, когда работать приходится просто в блокноте. Я веду к тому, что ознакомиться стоит со всеми программами и быть готовым к тому, что их придётся использовать.

Прежде чем выбирать окончательное решение, стоит поставить, может быть, все IDE и решить самому. Я лично надеюсь, что и Eclipse, и IntelliJ IDEA, NetBeans IDE будут чувствовать себя хорошо и жить долго, чтобы мы могли наслаждаться качеством конкуренции.

Я считаю, что в 2020-м году стоит начать с адекватной программы, и мой выбор падает на бесплатную **intellij idea community edition**, именно в ней будут даны первые уроки.

## Установка и настройка IntelliJ IDEA для работы в Java

**IntelliJ IDEA** — среда разработки программного обеспечения, сделанная в ~~Кронштадте~~ Санкт-Петербурге. Отличие среды разработки (IDE) от редакторов в том, что среда разработки должна "предугадывать" мысли и действия программиста. Можно, пожалуй, сравнить обычную бумажную записную книжку с телефонами друзей и электронную в телефоне. Удобство пользования электронной намного выше.

Повторюсь, что, прежде всего, у разработчика должна быть установлена на компьютере восьмая и/или актуальная версия JDK. На данный момент за Java Development Kit отвечает Oracle и у него можно скачать актуальную версию [тут](#). На момент написания статьи это была версия 8.02, и скачать её можно было как отдельно от NetBeans IDE 8.2, так и вместе. Я выбрал **JDK 8u111 with NetBeans 8.2**. Я очень надеюсь, что будущему разработчику не надо объяснять, какой кнопкой скачивать из интернета. После установки JDK можно перейти к IntelliJ IDEA.

**IntelliJ IDEA** можно скачать на сайте [www.jetbrains.com/idea/](http://www.jetbrains.com/idea/). Пожалуй, стоит знать, что там же качается и PhpStorm для php и PyCharm для Python.

IntelliJ IDEA есть в двух версиях — в бесплатной **Community** и платной Ultimate. Обе версии бывают только на одном языке, и это не русский. Для введения в Java бесплатная версия ничем не хуже платной. Различия в версиях можно посмотреть на сайте производителя перед скачиванием.

Если вам жизненно необходима Ультимативная версия, то не скачивайте "лекарство", не ломайте софт и не прописывайте левый сервер в лицензиях. Вы не сможете обновиться, а обновления бывают достаточно часто. Просто скачайте ультимативную версию и закажите себе бесплатные студенческие логины. Подробности студенческой лицензии тут — [www.jetbrains.com/student/](http://www.jetbrains.com/student/). IDEA нет на русском языке. Если русский язык очень важен, то у вас есть возможность работать с русским интерфейсом на NetBeans.

При установке можно для начала согласиться со всеми настройками по умолчанию и дойти до создания нового проекта. Проектов может быть множество. Например, игра для андроид-приборов или прошивка для "умного холодильника". Предположу, что вы не знаете, как будет называться ваша первая программа, поэтому можете назвать проект "myFirstJavaProject", "JavaBeginner" или "JavaFromAndron13". Название можно поменять позже, проект можно удалить или перенести файлы из проекта в проект, и поэтому над названием можно особо долго не думать. Желательно, что б оно было говорящее само за себя. На случай если у вас будет 20-30 или даже 1000 проектов.

## Домашнее задание

- Поиграйтесь с выводом строчек в своей среде разработки и попробуйте написать нормальное предложение. Пример выполненного домашнего задания:

```
public class lesson01.HelloWorld { // Начало нашего класса lesson01.HelloWorld
    public static void main(String[] args) { // Начинаем наш метод, главный метод (мэйн)
        System.out.print("Привет ");
        System.out.println("Мир! :-)");
        System.out.println("\t Hello World!!!");
        System.out.print("\t И ещё строчка :-)");
        System.out.print("Это моя первая программа \n" );
        System.out.print("\u0423\u0440\u0430 \n");
        System.out.println("- Mein erstes Programm :-)"); //моя первая программа
    } // Закрываем главный метод
} // Закрываем класс
```

- Удалите все комментарии из кода и запустите снова.

## Дополнительные материалы

List of Unicode characters [https://en.wikipedia.org/wiki/List\\_of\\_Unicode\\_characters](https://en.wikipedia.org/wiki/List_of_Unicode_characters)