

Язык программирования Java



Урок № 1

Введение в язык программирования Java

Содержание

1. Введение в технологию Java	3
1.1. Базовые понятия технологии Java	7
1.2. Жизненный цикл, компиляции и исполнения приложения Java.	8
1.3. Установка платформы Java	9
1.4. Пример написания первого приложения	18
1.5. Обзор существующих IDE.	21

1. Введение в технологию Java

В современном мире существует множество различных операционных систем (*Windows, Mac OS, Unix-подобные*). Программное обеспечение, написанное для одной системы, не будет работать на другой операционной системе. Это связано с различным устройством исполняемых файлов и функций операционных систем.

Java позволяет решить проблему запуска одного приложения на различных ОС, предлагая пользователям не только язык разработки, но и целую платформу.

Этапы развития технологии Java

Январь 1991 г. Начало работы над проектом Stealth	Июнь 1991 г. Гослинг начал работу над компилятором Oak	1992 г. Основание FirstPreson	Апрель 1993 г. Выпущен NCSA Mosaic 1.0	Апрель 1994 г. Основание Sun Interactive
Июнь 1994 г. Начало работы над проектом LiveOak	Сентябрь 1994 г. Впервые представлен продукт HotJava	Май 1995 г. Компания Sun представила Java, HotJava и SunWorld 95	Декабрь 1995 г. Sun и Netscape объявили о создании JavaScript	Июнь 1996 г. Выпущен пакет JDK 1.0
1996 г. Объявлены JavaBeans и сервлеты	Конец 1996 г. Объявлены Javacard и 100% Pure Java	Январь–февраль 1997 г. Реализована поддержка JavaBeans и JDK 1.1	Весна 1997 г. Выпущены средства JNDI	Весна 1997 г. Выпущены Java Media API и JDK для Linux
Март 1998 г. Выпущен пакет Swing; объявлен продукт Personal Java for Win CE	Апрель 1998 г. Начались поставки Java Plugin	Декабрь 1998 г. Началось распространение Java 2	Январь 1999 г. Объявлена поддержка службы JINI	Февраль 1999 г. Опубликованы исходные коды Java 2 M



Цели и задачи технологии Java

Основной целью технологии Java является принцип – «Написанное один раз – работает везде» (англ. «*Write once – run anywhere.*» WORA). Идея состоит в том, что написав один раз код приложения, можно повторно его использовать на различных устройствах и операционных системах, без каких-либо дополнительных усилий.

Немаловажной задачей Java является обеспечение высокой эффективности приложений, что достигается быстрым преобразованием байт-кода в язык машинных кодов.

Большое внимание в Java уделяется безопасности. В платформе присутствует множество классов для шифрования и работе с сертификатами.

Язык Java задумывался как «язык для домохозяек», т.е. он очень простой для понимания и изучения.

Сравнение технологии Java с другими технологиями программирования (.Net Framework, C++ и др.)

При разработке Java, за основу был взят язык C++, этим обусловлено множество совпадений в синтаксисе

и семантике. За основу объектной модели взят подход из языка Smalltalk.

Сравнение C++ и Java:

- в Java отсутствует перегрузка операторов;
- в Java отсутствует прямой доступ к памяти через указатель;
- в Java отсутствуют деструкторы;
- в Java отсутствует возможность процедурного программирования;
- в Java статическая длина массивов и проверка индекса при доступе к элементу;
- в Java значение переменных экземпляра (полей класса), в случае отсутствия инициализации, принимают значения по умолчанию (для целых и дробных типов это 0, для boolean – это false, для ссылочных типов – это null);
- в Java отсутствуют препроцессорные операции;
- в Java отсутствуют файлы заголовки (header);
- в Java контроль за расходом памяти возложен на специальный механизм сборки мусора, в C++ за этим следит сам программист.

Платформы Java и ее составные части

Java платформа – это набор средств для разработки на языке Java, которые поставляются разработчику в виде единого готового решения. В платформу входят:

- стандартные библиотеки;
- средств компиляции;
- средства выполнения кода;

- средства подготовки документации по коду;
- примеры;
- другие средства для разработки.

Редакции платформ Java

Java платформа поставляется в виде единого пакета разной редакции (комплектации). Для каждой редакции существует отдельная нумерация версий платформы.

- **JSE или J2SE** (*Java Standard Edition*) – это стандартная редакция платформы Java, используемая при разработке консольных приложений, в виде апплетов или standalone приложений с графическим интерфейсом, приложений работающих с базами данных и сетью.
- **JEE или J2EE** (*Java Enterprise Edition*) – это редакция платформы, которая предназначена для разработки распределенных приложений уровня предприятия. Содержит в себе технологии Servlet, JSP, EJB.
- **JME** (*Java Micro Edition*) – это редакция платформы, которая предназначена для разработки приложений под микрокомпьютеры (мобильные телефоны, планшеты и т.д.). В нее входят облегченные стандартные классы и классы для написания мидлетов (Midlets).
- **Java FX** – это платформа (2007 г.) для разработки приложений с графическим интерфейсом. Включена в 8 JSE, до этого была отдельной библиотекой.

1.1. Базовые понятия технологии Java

Виртуальная машина (JVM) – это спецификация, описывающая абстрактную машину, в которой могут выполняться приложения на Java. Машина считается виртуальной, так как большую ее часть составляют различные программные компоненты. Существуют JVM реализованные различными производителями, но все они придерживаются единой [спецификации](#). Реализация JVM это программное обеспечение, интерпретирующее инструкции байт-кода программы в команды операционной системы.

Компилятор – это средство, которое преобразует исходный код в байт-код. Java компилятор написан на языке Java. Существуют компиляторы, которые преобразуют исходный код с других языков (*Ada, JavaScript, Python, Ruby*) в байт-код. Существуют так же языки, адаптированные для работы на виртуальной Java машине (*Scala, Groovy*).

Байт код – это набор инструкций на промежуточном языке, предназначенный для выполнения виртуальной машинной. Каждая инструкция кодируется одним байтом. Существует 205 инструкций, остальные (51) являются резервом. Байт-код создается в результате компиляции кода программы, находящейся в файлах с расширением java. Байт-код сохраняется в файлах с расширением class.

Основным принципом технологии Java является тезис «Написанное один раз – работает везде.» (WORA). Благодаря особенностям платформы, программы, на-

писанные на Java, гарантированно работают на всех операционных системах, для которых существуют реализации виртуальной Java машины (Платформенная независимость).

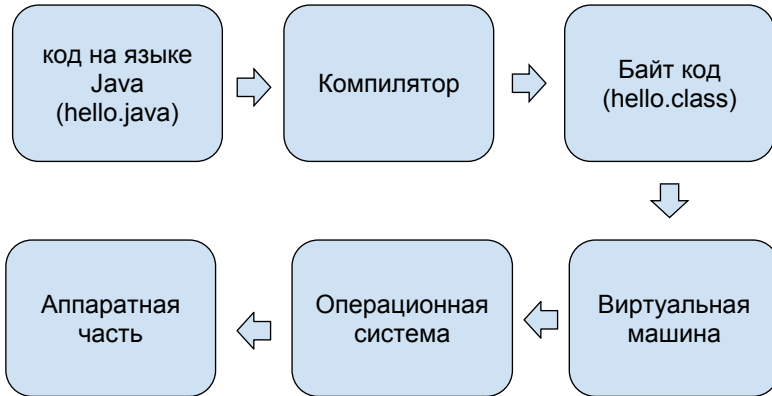
Сборка мусора

В некоторых языках программирования (например таких, как C++) контроль за использованием памяти возложен на плечи программиста, что усложняет процесс разработки и приводит к большому количеству ошибок и утечкам в памяти.

В Java контроль за использованием памяти возложен на специальный механизм, который называется сборкой мусора (garbage collector). Наличие механизма автоматической сборки мусора (более не используемых данных) упрощает процесс разработки и уменьшает количество ошибок при работе с памятью.

1.2. Жизненный цикл, компиляции и исполнения приложения Java

Разработка приложений на Java начинается с написания исходного кода. Код хранится в файлах с расширением java. Исходный код преобразуется (компилируется) программой компилятор в байт-код (файлы, имеющие расширения class). Для выполнения программы, байт-код запускается на виртуальной машине, которая интерпретирует его в команды операционной системы, которая в свою очередь передает команды аппаратной части электронного устройства.



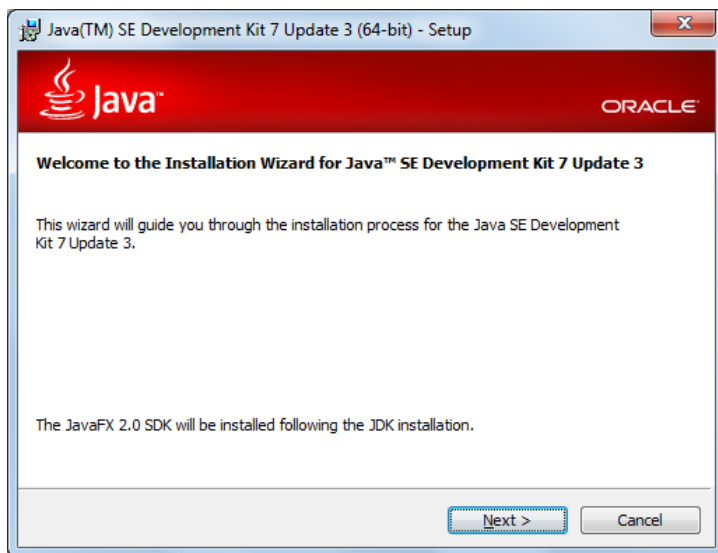
1.3. Установка платформы Java

Шаг 1.

Найти и скачать установочный пакет можно по адресу: <http://www.oracle.com/technetwork/java/javase/downloads/index.html>.

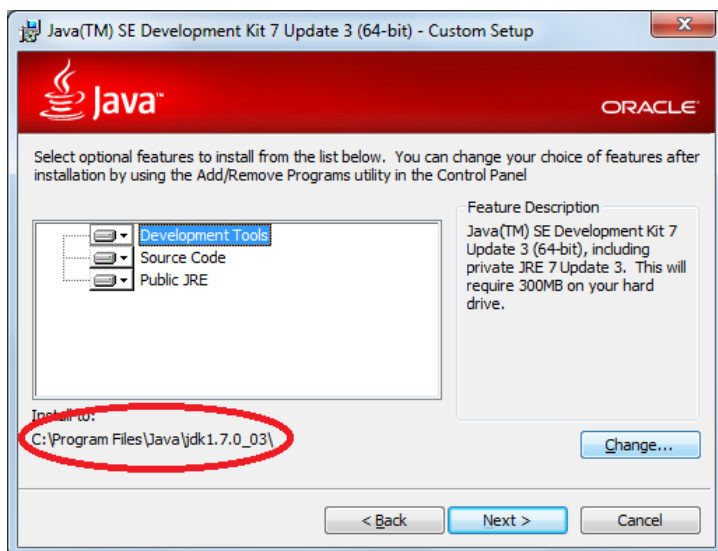
Выберете версию JDK и операционную систему (обратите внимание на разрядность), на которую будет установлена платформа.

Шаг 2. Запустите установочный пакет.



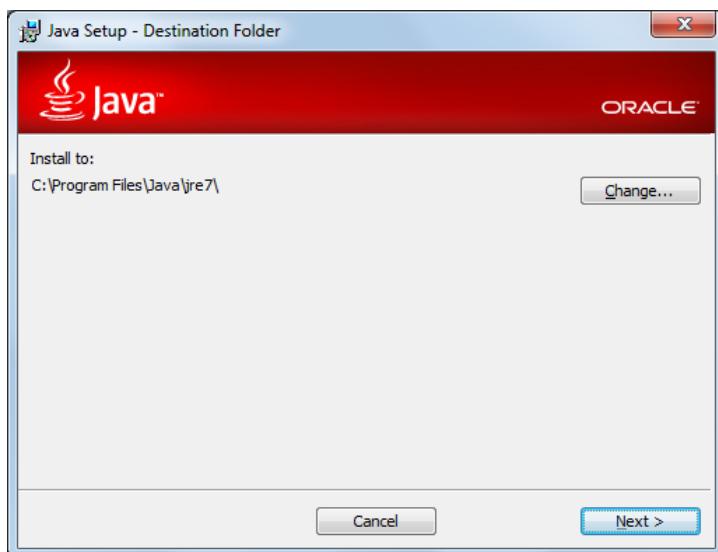
Шаг 3.

Если необходимо, смените путь для установки JDK.



Шаг 4.

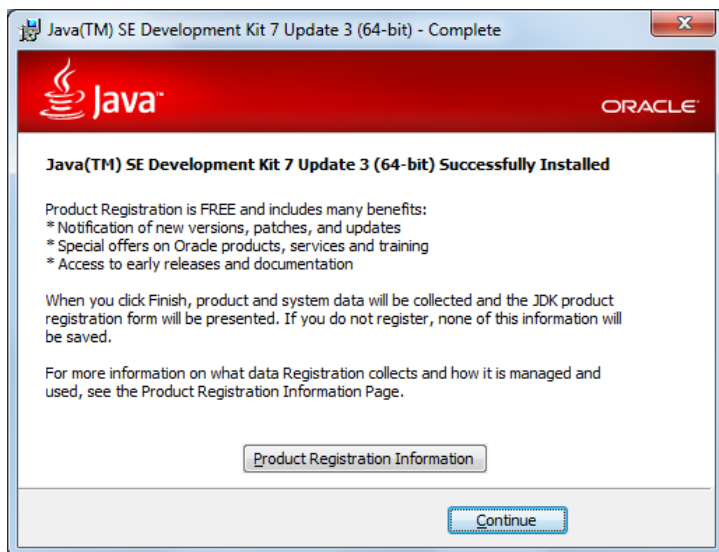
Установка JRE начнется сразу после установки JDK.



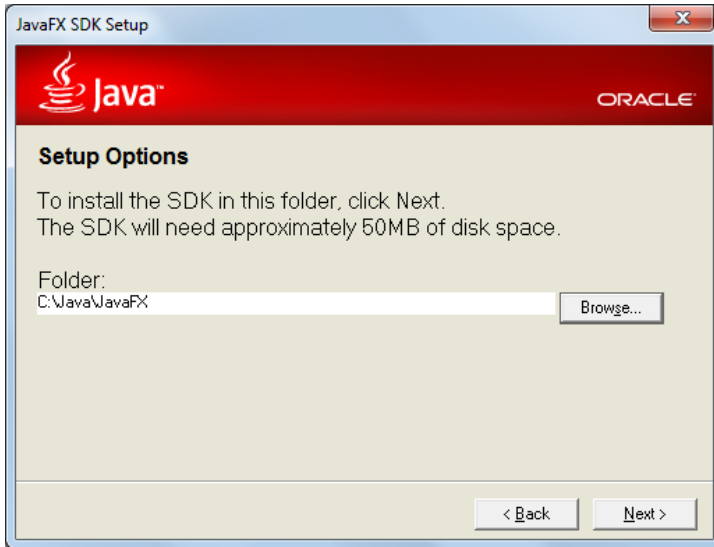
Процесс установки отображается в виде прогресса «Status» на экране окна.



Окончание установки. Есть возможность зарегистрировать установленную JDK на сайте «Oracle», нажав на кнопку «Product Registration Information». Но это не обязательный шаг.



Начиная с версии JSE 1.7, в установочный пакет входит платформа JavaFX, и ее установка начнется сразу после установки JRE.

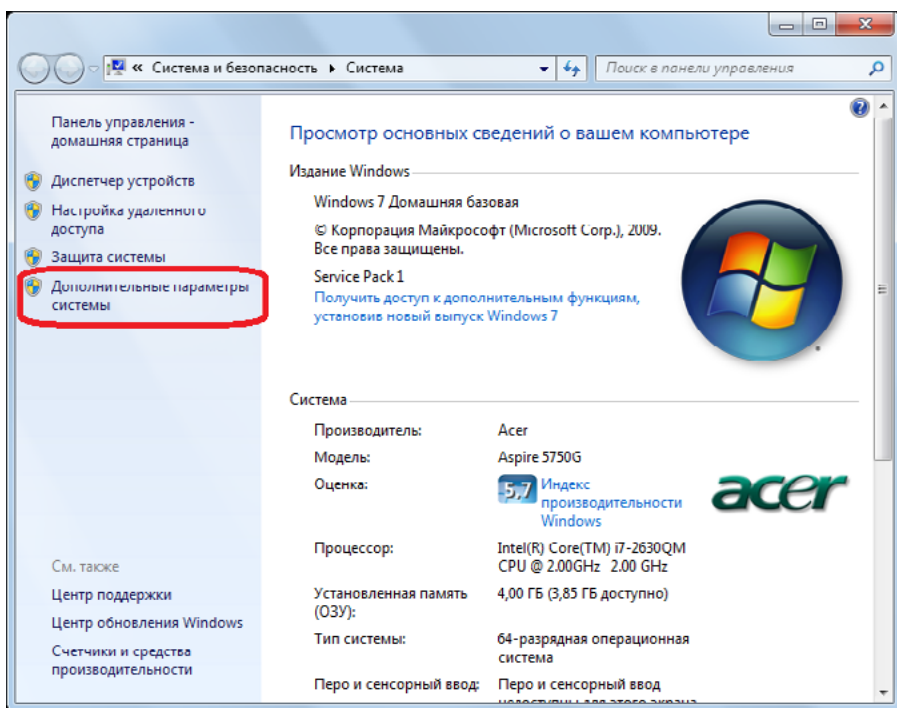


Шаг 5.

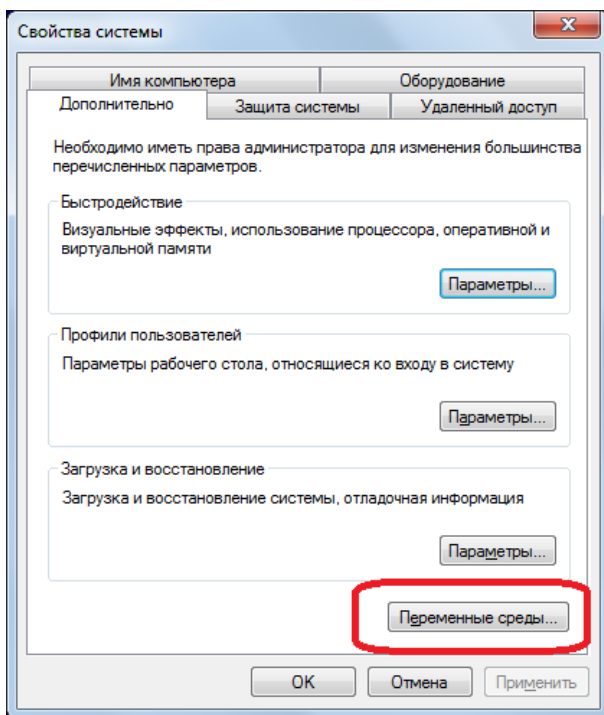
После установки Java платформы, необходимо сконфигурировать операционную систему для работы с Java.

Для Windows найдите «Компьютер» («Мой компьютер») и выберите его с помощью правой кнопки мыши – в появившемся меню выберите пункт «Свойства».

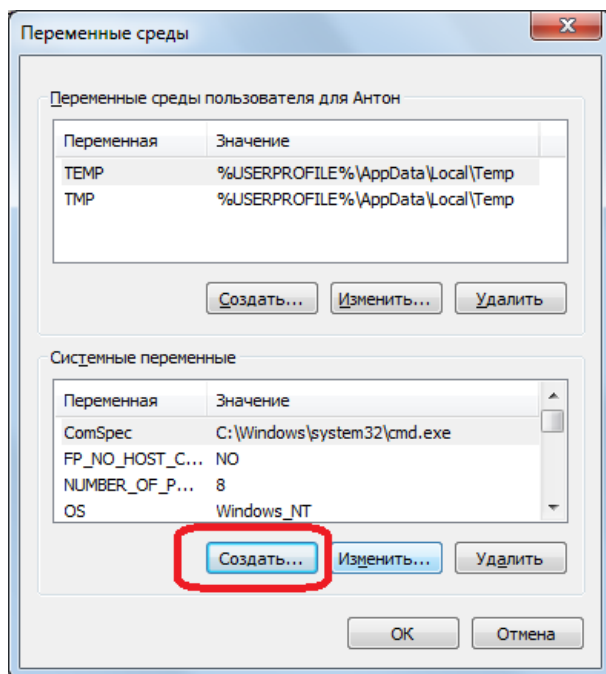
В появившемся окне выберите пункт «Дополнительные параметры системы».



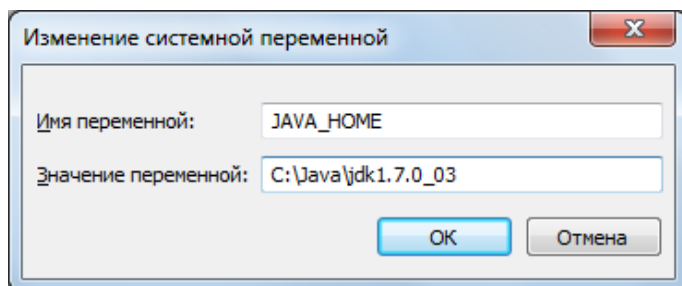
В новом окне необходимо нажать кнопку «Переменные среды».



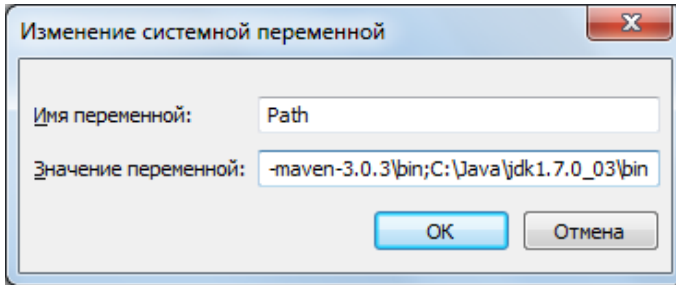
Выберите кнопку «Создать».



В поле «Имя переменной» внесите значение «Java_HOME», в поле «Значение переменной» внесите путь, куда была установлена JDK и нажмите «ОК».



Найдите и выберите в окне «Переменные среды», в поле «Системные переменные» пункт «Patch», и нажмите кнопку «Изменить». В поле «Значение переменной» добавьте точку с запятой и путь к директории, где установлены утилиты JDK (директория bin).



ВНИМАНИЕ!!! Не очищайте и не удаляйте информацию из поля «Значения переменной».

JRE (*Java Runtime Environment*) – среда выполнения для Java приложений. Минимальный набор необходимый запуска Java приложений. Содержит виртуальную машину и библиотеки Java классов.

JDK (*Java Development Kit*) – это набор для Java разработчика, включающий в себя:

- компилятор (javac);
- стандартные библиотеки классов;
- примеры кода;
- документацию;
- утилиты;
- исполнительную систему (java).

java – исполняемый файл (находящийся в ...\\Java\\jdk\\bin), запускающий виртуальную машину и передающий

байт-код для интерпретации. Файлы с байт-кодом называются в качестве параметров.

Например:

```
java.exe -d HelloWorld.class
```

javac – компилятор исходного кода в байт-код.

Например:

```
javac.exe HelloWorld.java.
```

jar – утилита создания архива (java ARchive) ZIP формата, в котором упаковываются файлы с байт-кодом и ресурсы программы. Предназначен для объединения модулей программы в цельные логические единицы. Jar файл может быть исполняемым, если содержит файл манифеста.

javadoc – утилита создания документации на основе исходного кода.

appletviewer – утилита просмотра приложений типа апплет.

1.4. Пример написания первого приложения

Почти во всех средах разработки программного кода на Java, сам процесс компиляции и запуска программы скрыт от пользователя набором различных конфигураций запуска. В результате этого сокрытия многие начинающие разработчики не знают, как скомпилировать или запустить свою программу без IDE, хотя такие знания могут пригодиться при формировании различных скриптов сборки.

Запустить первую программу без IDE можно в три простых шага.

Шаг 1.

Создайте в блокноте файл HelloWorld.java. Заполните файл кодом из листинга 1.1 и сохраните его.

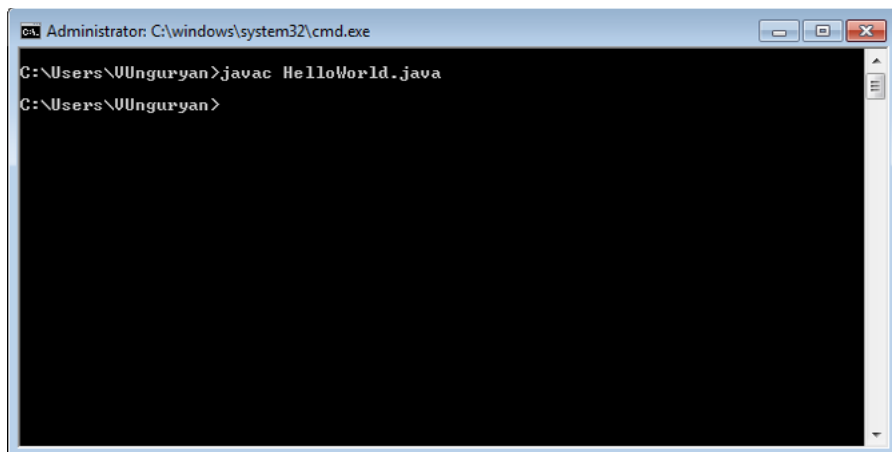
Листинг 1.1

```
/**
 * Моя первая программа
 */
public class HelloWorld
{
    /**
     * Точка входа в программу
     */
    public static void main(String[] args)
    {
        System.out.println("Hello world!");
    }
}
```

ВАЖНО!!! Имя файла и регистр букв в имени файла должен в точности совпадать с именем класса в программе (следующим за ключевым словом `class`).

Шаг 2.

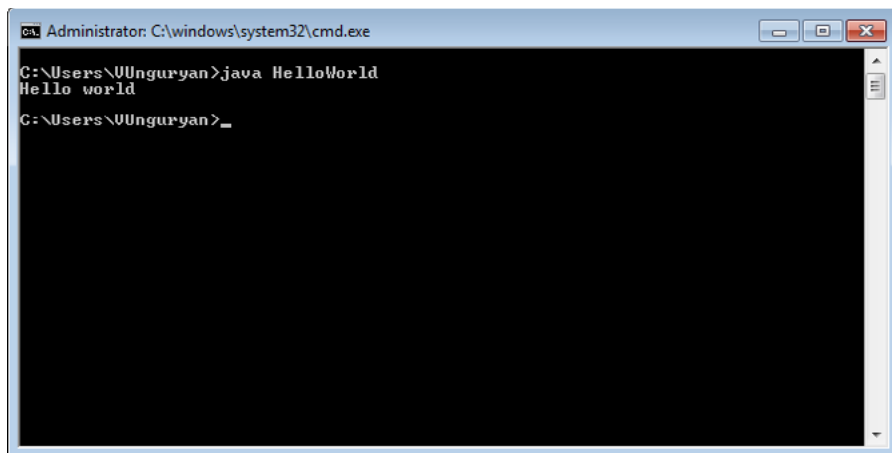
Откройте утилиту командной строки. Введите «javac HelloWorld.java» и нажмите на клавиатуре клавишу «Enter». В случае успешной компиляции появится новый файл, имеющий то же имя, но с расширением «class». Это файл с байт-кодом вашей программы.



```
Administrator: C:\windows\system32\cmd.exe
C:\Users\Uunguryan>javac HelloWorld.java
C:\Users\Uunguryan>
```

Шаг 3.

Введите в командную строку `java HelloWorld.class` и нажмите Enter. Программа выполнится, после ее выполнения в окне командной строки будет выведено сообщение: `Hello world!`



```
Administrator: C:\windows\system32\cmd.exe
C:\Users\Uunguryan>java HelloWorld
Hello world
C:\Users\Uunguryan>_
```

1.5. Обзор существующих IDE

Для упрощения и ускорения процесса разработки программ под Java, создано большое количество различных интегрированных сред разработки, далее IDE (*Integrated Development Environment*).

Все IDE можно разделить на две основные категории, которые существенно влияют на выбор:

- платные;
- бесплатные.

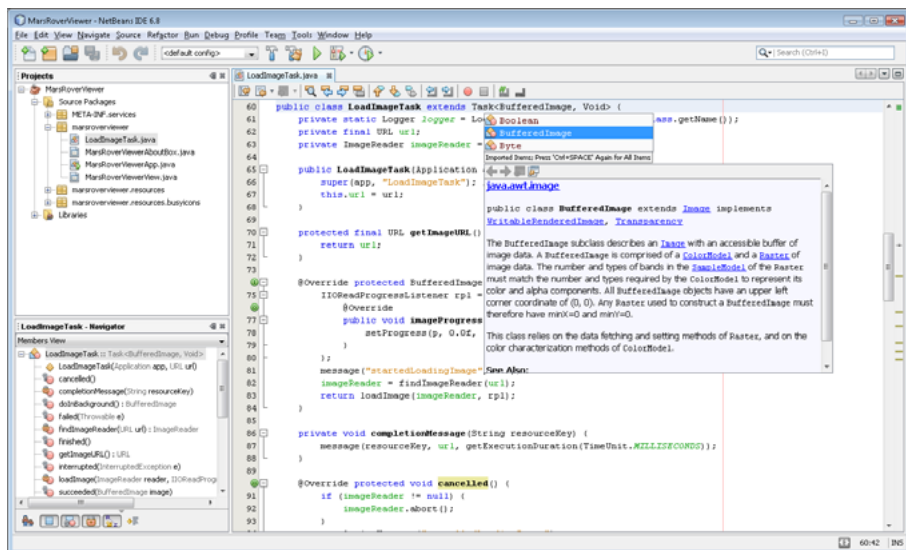
Так же немаловажными параметрами при выборе IDE является время, требуемое для первоначальной загрузки, время установки среды, объем занимаемой в ОЗУ и виртуальной памяти.

Лидерами по частоте использования являются: *NetBeans*, *Eclips*, *Intelej IDEA*.

Как правило, многие продуктовые компании диктуют программисту, какую IDE ему использовать на проекте, поэтому будет очень хорошо, если вы будете иметь представление и небольшой опыт в каждой из популярных IDE.



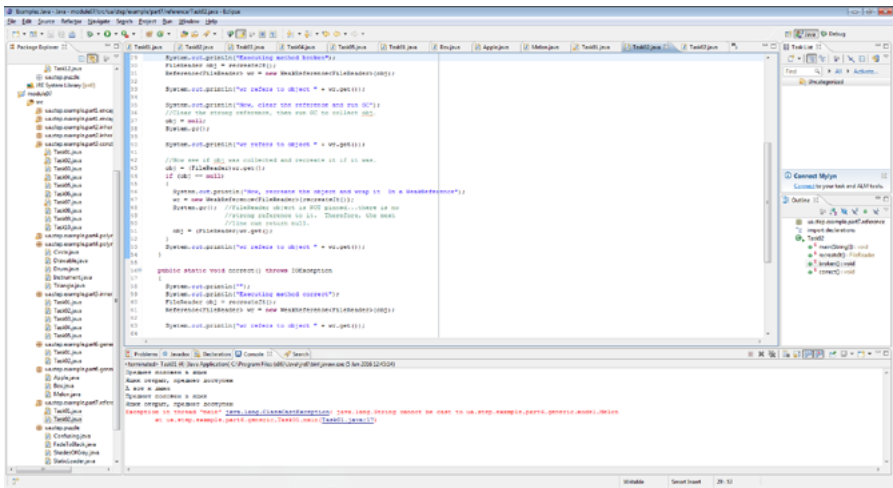
NetBeans – бесплатная свободная среда разработки с открытым исходным кодом. Поддерживает различные языки программирования и разметки. Поддержка большинства современных Java технологий. Присутствует интерфейс на русском языке и большое русскоязычное сообщество.



Официальный сайт: <https://netbeans.org/>.



IDE Eclipse – бесплатная свободная среда для разработки многомодульных и кроссплатформенных приложений. Имеет открытый исходный код. Развивается и поддерживается некоммерческой организацией «Eclipse Foundation». Изначально среда принадлежала IBM. Представляет собой многослойный конструктор состоящий из сервисной платформы «OSGI» и «SWT/JFace». Расширение функционала осуществляется с помощью плагинов (plugins), которых имеется большое количество.

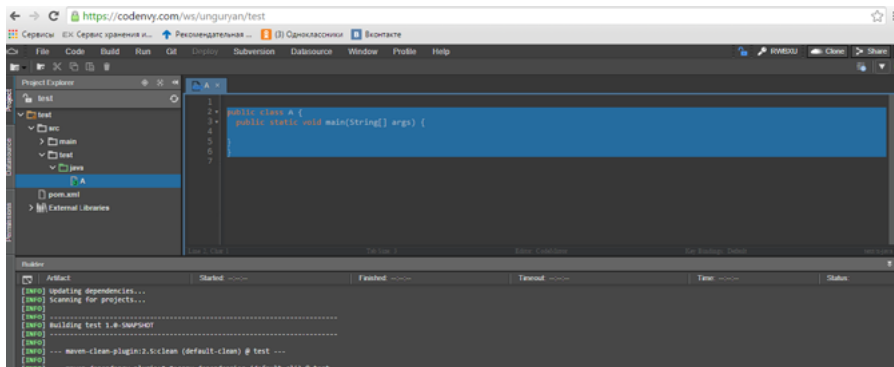


Официальный сайт: <https://eclipse.org/>.

«**InteleJ IDEA**» – условно-бесплатная интеллектуальная среда разработки. Разрабатывается компанией «Jet Brains». Отличительной особенностью является понимание контекста и большого количества поддерживаемых технологий и фреймворков прямо из коробки. Имеется бесплатная версия Community (отсутствует поддержка JEE технологий) и платная Ultimate (30-дневный пробный период).

Официальный сайт: <https://www.jetbrains.com/idea/>.

Codenvy – бесплатная среда разработки, не требующая установки приложения на компьютер. Разработка ведется в браузере, после загрузки в него необходимого контента.



Официальный сайт: <https://codenvy.com/>.



Урок № 1

Введение в язык программирования Java

© Виталий Унгурян

© Компьютерная Академия «Шаг»

www.itstep.org

Все права на охраняемые авторским правом фото-, аудио- и видеопроизведения, фрагменты которых использованы в материале, принадлежат их законным владельцам. Фрагменты произведений используются в иллюстративных целях в объёме, оправданном поставленной задачей, в рамках учебного процесса и в учебных целях, в соответствии со ст. 1274 ч. 4 ГК РФ и ст. 21 и 23 Закона Украины «Про авторське право і суміжні права». Объём и способ цитируемых произведений соответствует принятым нормам, не наносит ущерба нормальному использованию объектов авторского права и не ущемляет законные интересы автора и правообладателей. Цитируемые фрагменты произведений на момент использования не могут быть заменены альтернативными, не охраняемыми авторским правом аналогами, и как таковые соответствуют критериям добросовестного использования и честного использования.

Все права защищены. Полное или частичное копирование материалов запрещено. Согласование использования произведений или их фрагментов производится с авторами и правообладателями. Согласованное использование материалов возможно только при указании источника.

Ответственность за несанкционированное копирование и коммерческое использование материалов определяется действующим законодательством Украины.