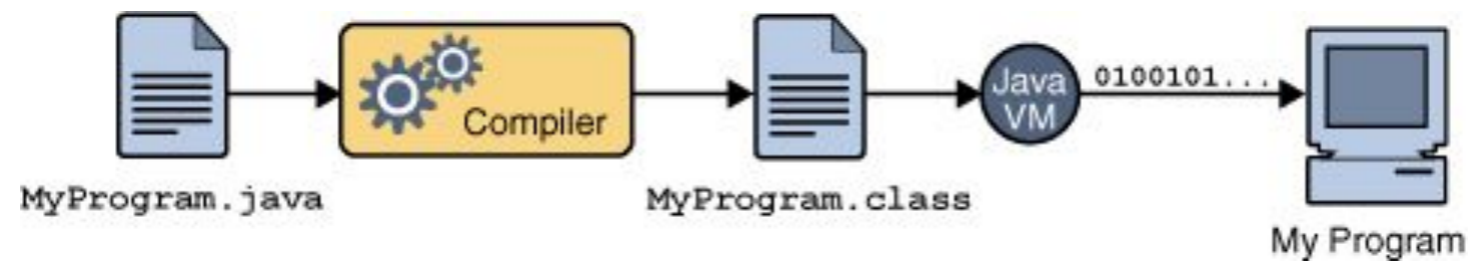


# I. Введение

## 1. Первые шаги



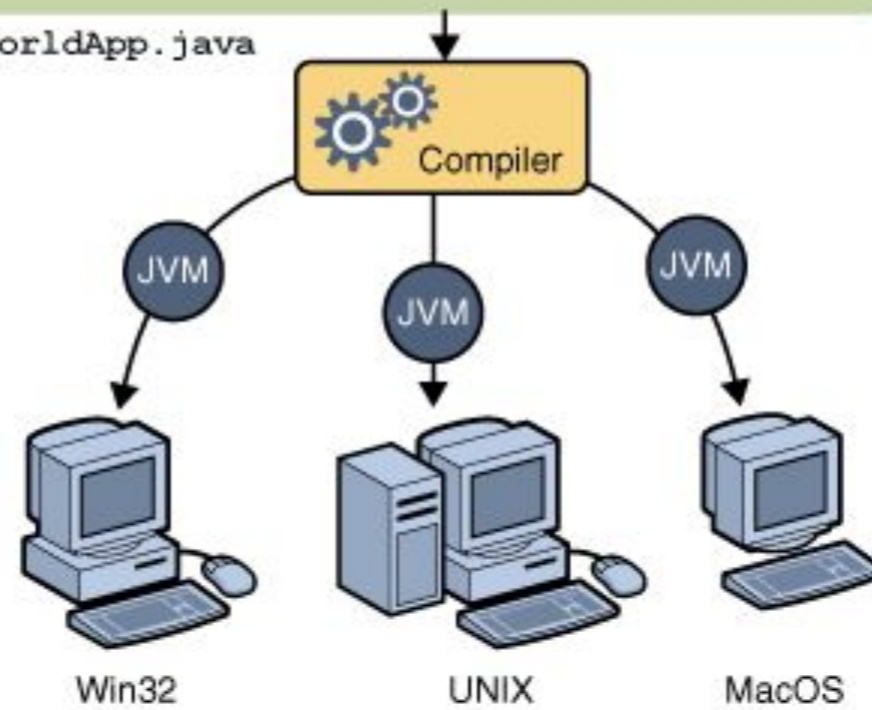
Java — объектно-ориентированный, платформо-независимый язык программирования. Приложения Java обычно компилируются в платформо-независимый байт-код который может выполняться с использованием специального приложения называемым виртуальной машиной Java (Java Virtual Machine JVM). Компилятор и виртуальная машина для каждой архитектуры и операционной системы свои.

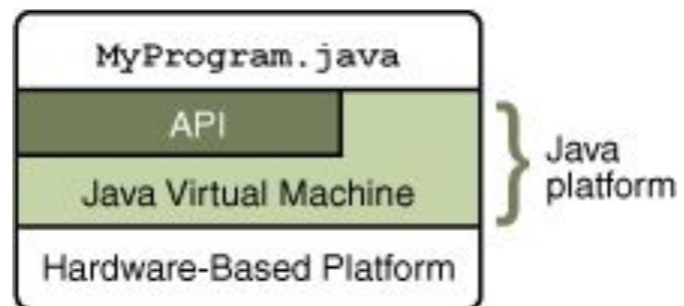


Java Program

```
class HelloWorldApp {  
    public static void main(String[] args) {  
        System.out.println("Hello World!");  
    }  
}
```

HelloWorldApp.java





Java Runtime Environment (JRE) — исполнительная среда Java состоит из реализации виртуальной машины Java, необходимой для исполнения Java - приложений и библиотеки Java - классов.



Java Development Kit (JDK) — комплект разработчика приложений на языке Java включает компилятор Java (javac), стандартные библиотеки классов Java, примеры, документацию, различные утилиты и исполнительную среду Java (JRE).

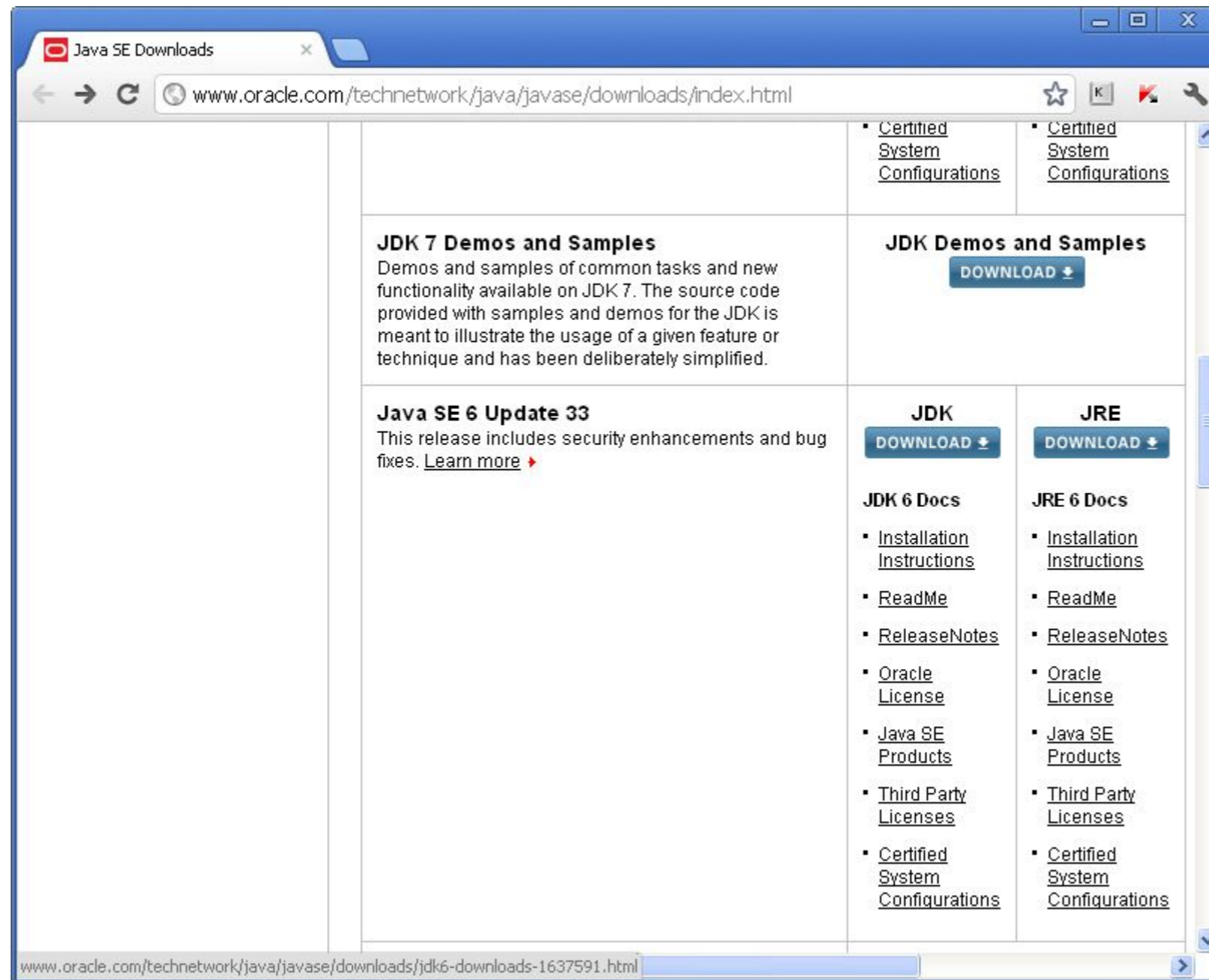
# История Java


# Версии Java

| Версия    | Год  | Число классов и интерфейсов в библиотеке | Версия class файлов | Что нового   |
|-----------|------|--|---------------------|--|
| JDK 1.0   | 1996 | 211                                      | 45.3                | Сам язык   |
| JDK 1.1   | 1997 | 477                                      | 45.3                | Внутренние классы  |
| J2SE 1.2  | 1998 | 1,524                                    | 46.0                | Ничего   |
| J2SE 1.3  | 2000 | 1,840                                    | 47.0                | Ничего   |
| J2SE 1.4  | 2002 | 2,723                                    | 48.0                | Утверждения  |
| J2SE 5.0  | 2004 | 3,279                                    | 49.0                | Цикл for each. Родовые типы. Автоматическая упаковка и распаковка. Перечисления. Метаданные. Переменное число аргументов. static import. |
| Java SE 6 | 2006 | 3,777                                    | 50.0                | Ничего   |
| Java SE 7 | 2011 |  | 51.0                | Использование класса String в блоке switch. Объединённая обработка исключений. Подчёркивание в числовых литералах.                       |

# Загрузка и установка JDK





 JDK можно абсолютно бесплатно загрузить с сайта <http://www.oracle.com/technetwork/java/javase/downloads>. На момент разработки курса последней версии Java SE 6 была Java SE 6 Update 33.

Oracle Java Development Kit 6 Update 33

You must accept the [Oracle Binary Code License Agreement for Java SE](#) to download this software.


Thank you for accepting the Oracle Binary Code License Agreement for Java SE; you may now download this software.

| Product / File Description | File Size | Download                                       |
|----------------------------|-----------|--|
| Linux x86                  | 65.42 MB  | <a href="#">jdk-6u33-linux-i586-rpm.bin</a>    |
| Linux x86                  | 68.42 MB  | <a href="#">jdk-6u33-linux-i586.bin</a>        |
| Linux x64                  | 65.64 MB  | <a href="#">jdk-6u33-linux-x64-rpm.bin</a>     |
| Linux x64                  | 68.69 MB  | <a href="#">jdk-6u33-linux-x64.bin</a>         |
| Solaris x86                | 68.33 MB  | <a href="#">jdk-6u33-solaris-i586.sh</a>       |
| Solaris x86                | 119.88 MB | <a href="#">jdk-6u33-solaris-i586.tar.Z</a>    |
| Solaris SPARC              | 73.3 MB   | <a href="#">jdk-6u33-solaris-sparc.sh</a>      |
| Solaris SPARC              | 124.45 MB | <a href="#">jdk-6u33-solaris-sparc.tar.Z</a>   |
| Solaris SPARC 64-bit       | 12.18 MB  | <a href="#">jdk-6u33-solaris-sparcv9.sh</a>    |
| Solaris SPARC 64-bit       | 15.59 MB  | <a href="#">jdk-6u33-solaris-sparcv9.tar.Z</a> |
| Solaris x64                | 8.44 MB   | <a href="#">jdk-6u33-solaris-x64.sh</a>        |
| Solaris x64                | 12.24 MB  | <a href="#">jdk-6u33-solaris-x64.tar.Z</a>     |
| Windows x86                | 69.66 MB  | <a href="#">jdk-6u33-windows-i586.exe</a>      |
| Windows x64                | 59.67 MB  | <a href="#">jdk-6u33-windows-x64.exe</a>       |
| Linux Intel Itanium        | 53.94 MB  | <a href="#">jdk-6u33-linux-ia64-rpm.bin</a>    |
| Linux Intel Itanium        | 60.65 MB  | <a href="#">jdk-6u33-linux-ia64.bin</a>        |
| Windows Intel Itanium      | 57.86 MB  | <a href="#">jdk-6u33-windows-ia64.exe</a>      |

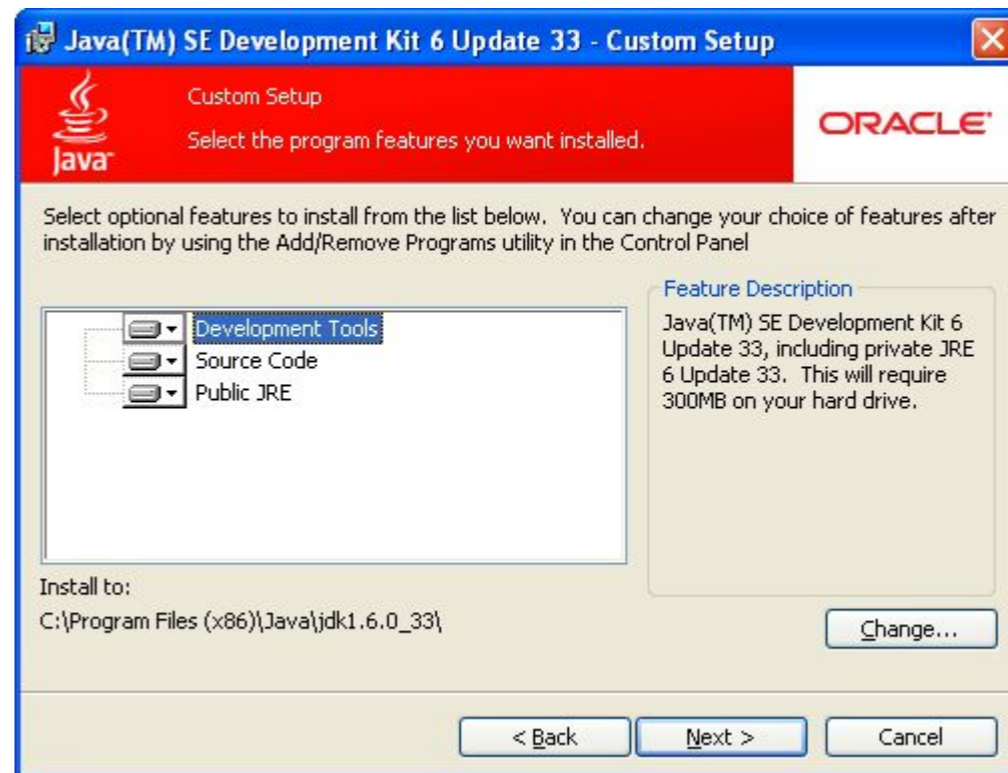
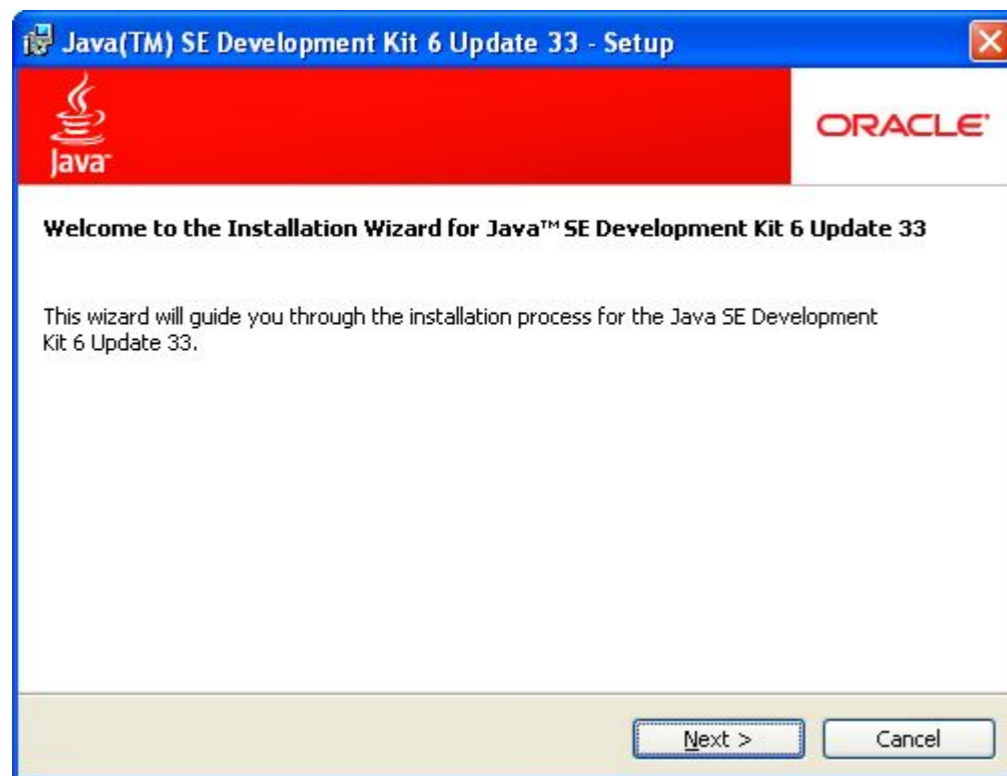
Java SE Development Kit 6u33 Demos and Samples Downloads

jdk-6u33-windows-i586.exe  
25.4/69.7 MB, 52 secs left

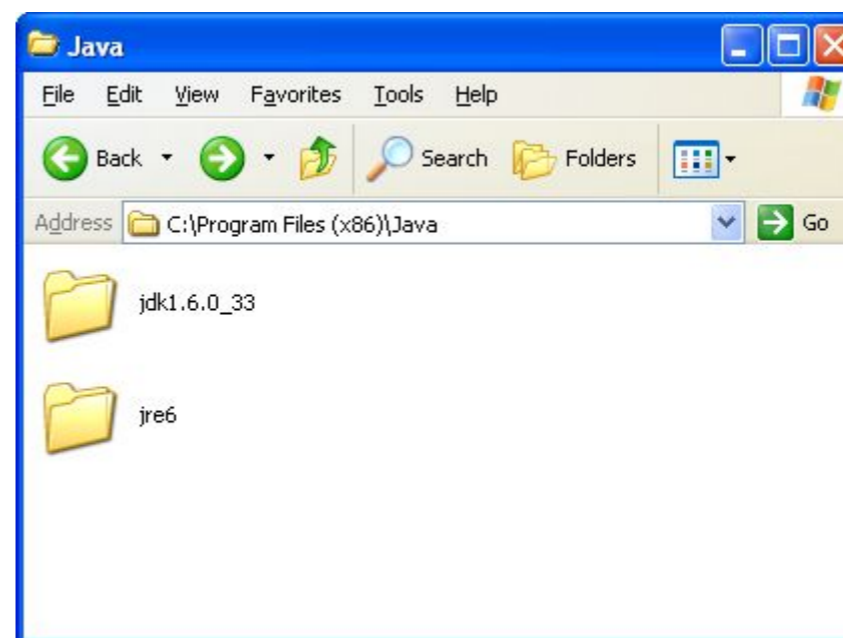
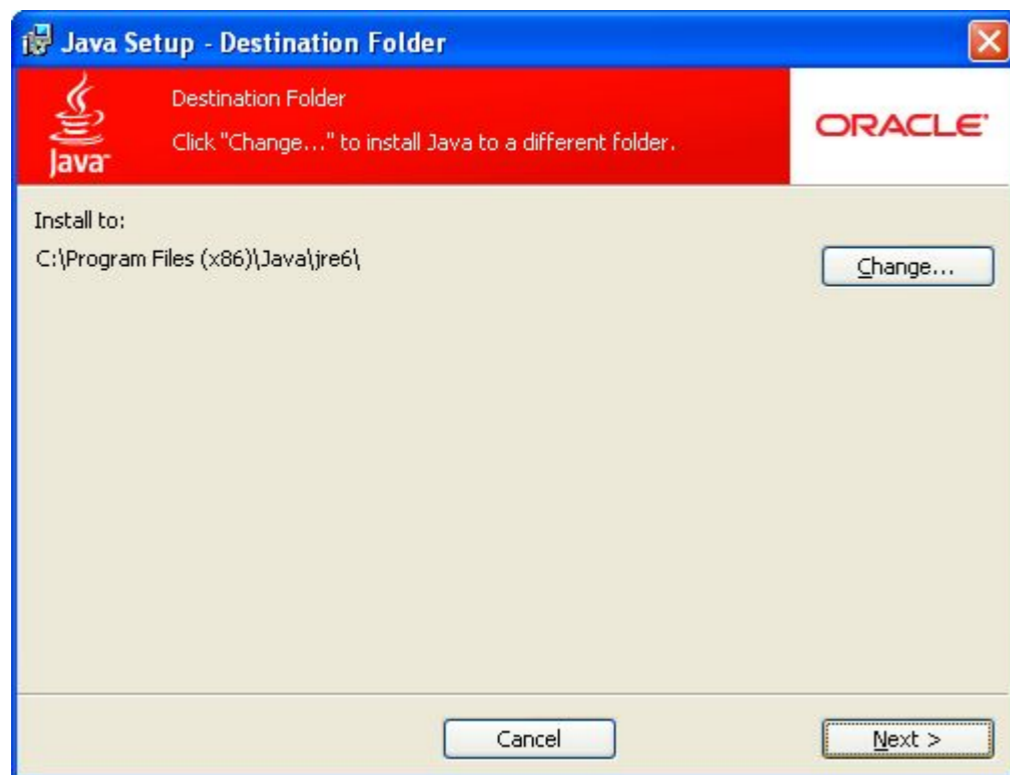
Show all downloads...

 Для загрузки необходимо принять условия лицензионного соглашения и выбрать архитектуру/операционную систему. В данном примере выбран вариант windows-i586.

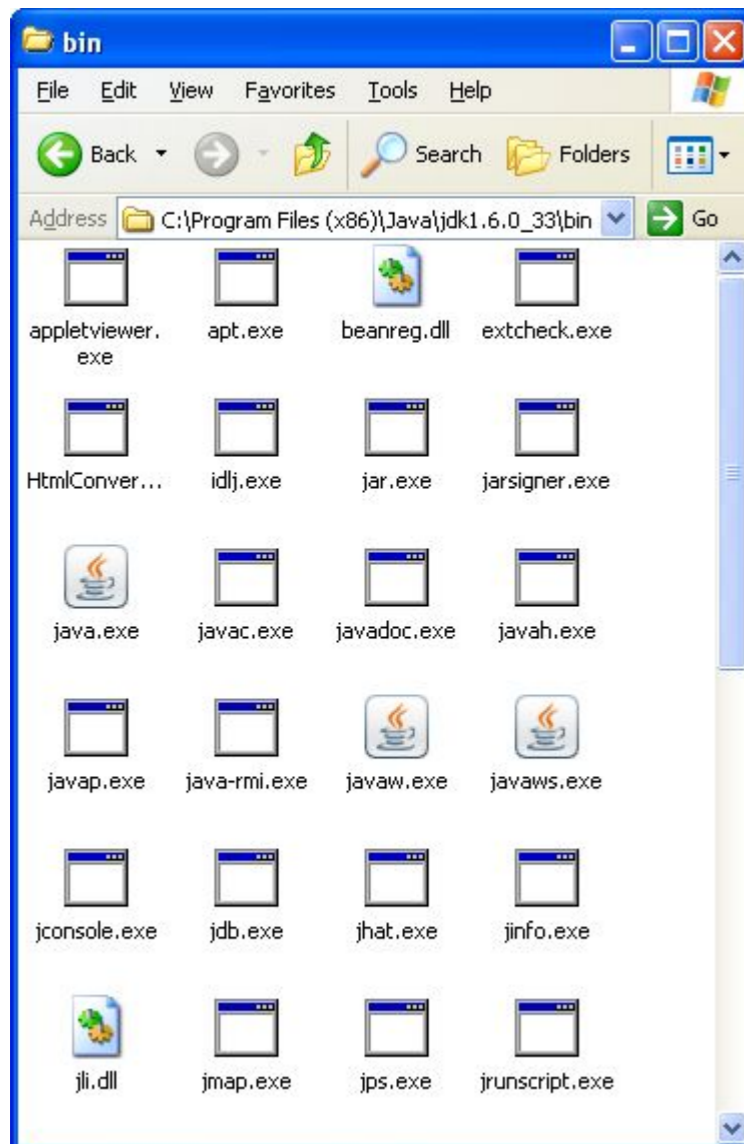
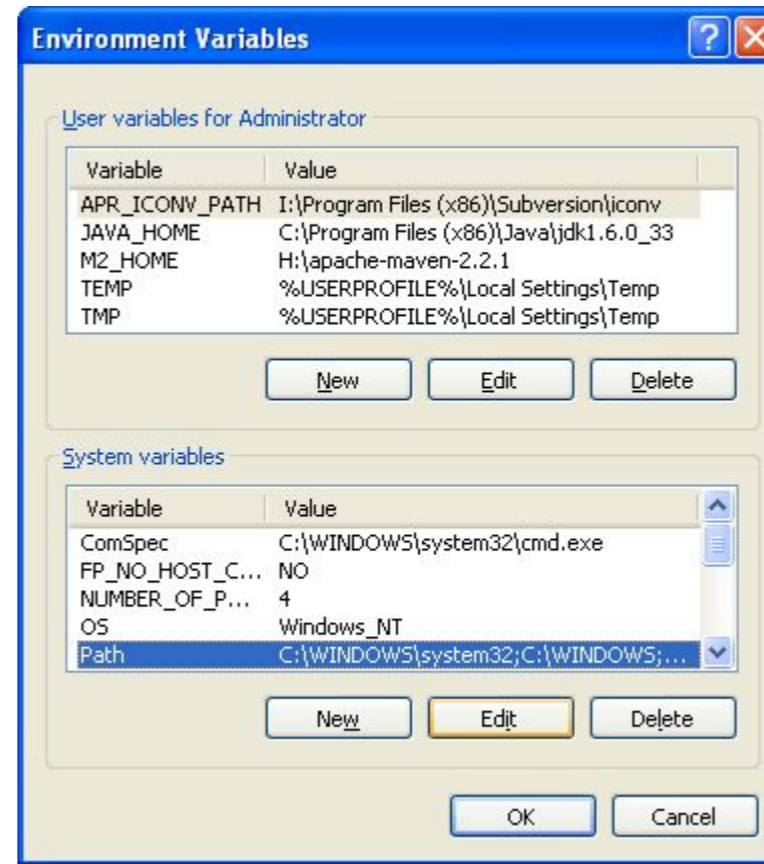
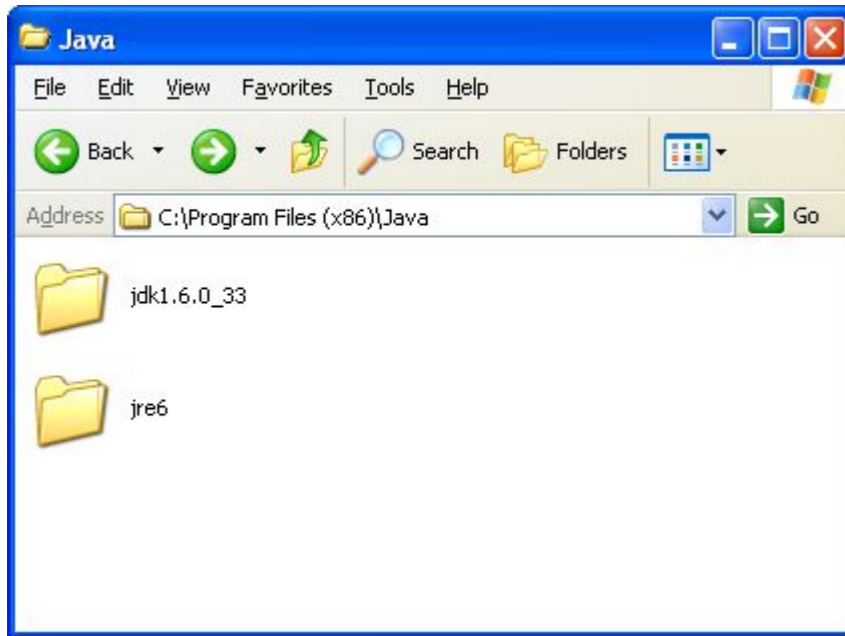
# Установка JDK




# Установка JDK



# Задание переменных окружения



 В переменную среды Path необходимо добавить поддиректорию bin из директории установки JDK. Переменные окружения будут действительны только во вновь открытой консоли.

# Проверка установки

```
Microsoft Windows [Version 5.2.3790]  
(C) Copyright 1985-2003 Microsoft Corp.
```

```
C:\Documents and Settings\Administrator>java -version
```

```
java version "1.6.0_33"  
Java(TM) SE Runtime Environment (build 1.6.0_33-b03)  
Java HotSpot(TM) Client VM (build 20.8-b03, mixed mode, sharing)
```

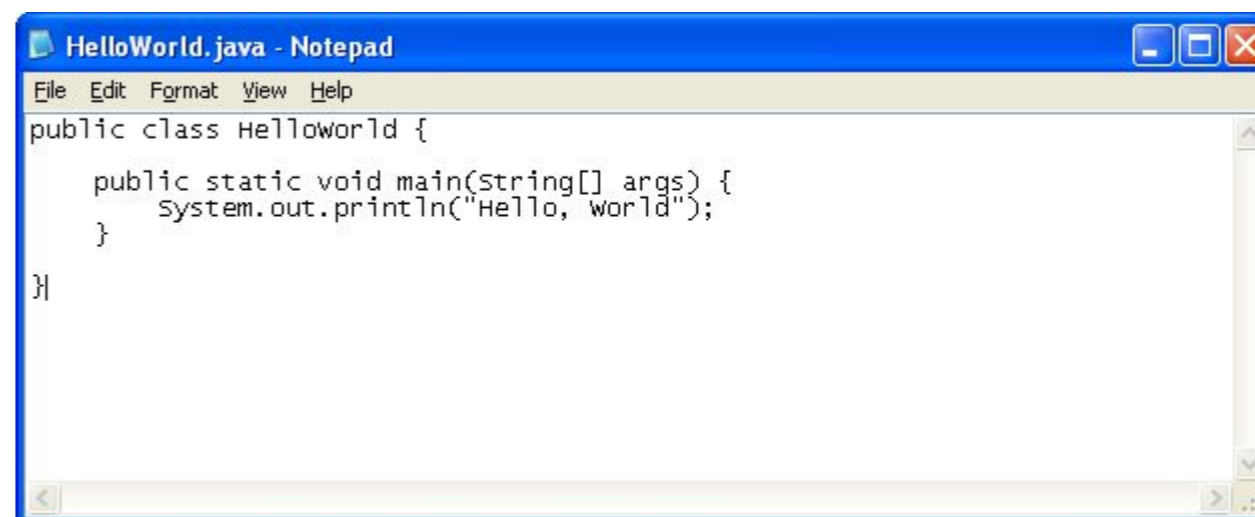
```
C:\Documents and Settings\Administrator>javac -version
```

```
javac 1.6.0_33
```

```
C:\Documents and Settings\Administrator>
```


Простейшее приложение

# Создание приложения



```
File Edit Format View Help
public class HelloWorld {
    public static void main(String[] args) {
        System.out.println("Hello, world");
    }
}
```

A screenshot of a Notepad window titled 'HelloWorld.java - Notepad'. The window contains the following Java source code:

 Для разработки java приложений можно использовать любой текстовый редактор. Убедитесь что созданный файл не имеет скрытое расширение txt.



# Точка входа

```
public class HelloWorld {  
    public static void main(String[] args) {  
        System.out.println("Hello World!");  
    }  
}
```



Точкой входа приложения может быть любой класс содержащий метод `public static void main(String[] args)`.

# Компиляция приложения

```
I:\helloworld>dir
Volume in drive I has no label.
Volume Serial Number is 44AB-CB89

Directory of I:\helloworld

02/20/2013  10:17 AM    <DIR>          .
02/20/2013  10:17 AM    <DIR>          ..
11/14/2012  03:16 PM                140 HelloWorld.java
                1 File(s)        140 bytes
                2 Dir(s)  48,620,081,152 bytes free

I:\helloworld> javac HelloWorld.java

I:\helloworld>dir
Volume in drive I has no label.
Volume Serial Number is 44AB-CB89

Directory of I:\helloworld

02/20/2013  10:17 AM    <DIR>          .
02/20/2013  10:17 AM    <DIR>          ..
02/20/2013  10:17 AM                426 HelloWorld.class
11/14/2012  03:16 PM                140 HelloWorld.java
                2 File(s)        566 bytes
                2 Dir(s)  48,620,081,152 bytes free

I:\helloworld>
```



Для компиляции используется компилятор `javac.exe`. В качестве параметра ему передаётся название модуля компиляции. Для каждого класса из модуля компиляции создаётся `class` файл содержащий байт код. По умолчанию `class` файлы добавляются в директорию где находится модуль компиляции.

# Компиляция приложения

```
G:\> javac -verbose HelloWorld.java
[parsing started HelloWorld.java]
[parsing completed 31ms]
[search path for source files: .]
[search path for class files: C:\Program Files (x86)\Java\jdk1.6.0_35\jre\lib\re
sources.jar,C:\Program Files (x86)\Java\jdk1.6.0_35\jre\lib\rt.jar,C:\Program Fi
les (x86)\Java\jdk1.6.0_35\jre\lib\sunrsasign.jar,C:\Program Files (x86)\Java\jd
k1.6.0_35\jre\lib\jsse.jar,C:\Program Files (x86)\Java\jdk1.6.0_35\jre\lib\jce.j
ar,C:\Program Files (x86)\Java\jdk1.6.0_35\jre\lib\charsets.jar,C:\Program Files
(x86)\Java\jdk1.6.0_35\jre\lib\modules\jdk.boot.jar,C:\Program Files (x86)\Java
\jdk1.6.0_35\jre\classes,C:\Program Files (x86)\Java\jdk1.6.0_35\jre\lib\ext\dns
ns.jar,C:\Program Files (x86)\Java\jdk1.6.0_35\jre\lib\ext\localedata.jar,C:\Pro
gram Files (x86)\Java\jdk1.6.0_35\jre\lib\ext\sunjce_provider.jar,C:\Program Fil
es (x86)\Java\jdk1.6.0_35\jre\lib\ext\sunmscapi.jar,C:\Program Files (x86)\Java\
jdk1.6.0_35\jre\lib\ext\sunpkcs11.jar,.]
[loading java\lang\Object.class(java\lang:Object.class)]
[loading java\lang\String.class(java\lang:String.class)]
[checking HelloWorld]
[loading java\lang\System.class(java\lang:System.class)]
[loading java\io\PrintStream.class(java\io:PrintStream.class)]
[loading java\io\FilterOutputStream.class(java\io:FilterOutputStream.class)]
[loading java\io\OutputStream.class(java\io:OutputStream.class)]
[wrote HelloWorld.class]
[total 157ms]

G:\>
```



Для получения подробной информации во время компиляции можно использовать флаг `-verbose`.

# Запуск приложения

```
F:\>dir
Volume in drive F has no label.
Volume Serial Number is 58B9-DDFE

Directory of F:\

10/15/2012  05:54 PM                426 HelloWorld.class
10/15/2012  05:52 PM                131 HelloWorld.java
           2 File(s)                557 bytes
           0 Dir(s)  41,562,402,816 bytes free

F:\>java HelloWorld
Hello World!

F:\>
```



Для запуска приложения используется виртуальная машина Java – java.exe. В качестве параметра указывается название класса который будет точкой входа.

Байт код

# Байт код

```
F:\> javap -c HelloWorld
Compiled from "HelloWorld.java"
public class HelloWorld extends java.lang.Object{
public HelloWorld();
  Code:
    0:   aload_0
    1:   invokespecial   #1; //Method java/lang/Object."<init>":()V
    4:   return

public static void main(java.lang.String[]);
  Code:
    0:   getstatic       #2; //Field java/lang/System.out:Ljava/io/PrintStream;
    3:   ldc            #3; //String Hello World!
    5:   invokevirtual  #4; //Method java/io/PrintStream.println:(Ljava/lang/String;)V
    8:   return
}

F:\>
```



Для просмотра байт кода можно использовать дизассемблер javap.

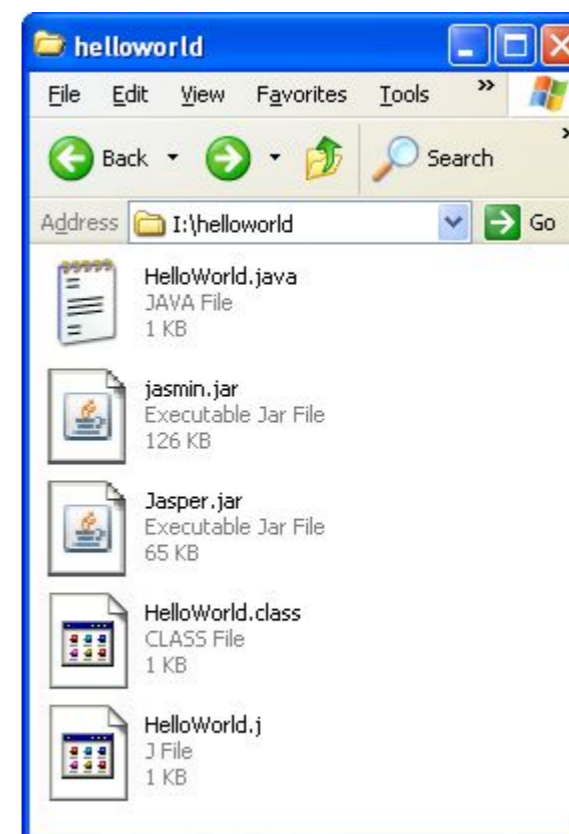
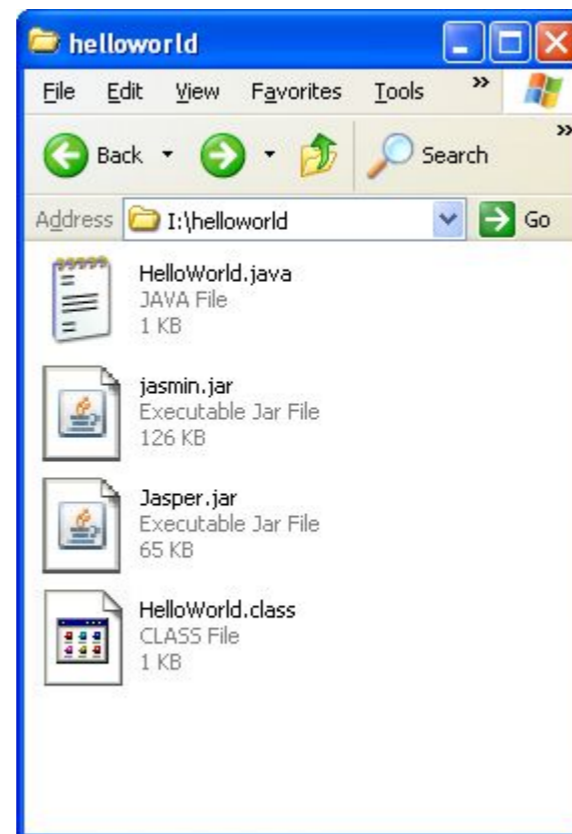


Более удобными средствами для дизассемблирования класс файлов и ознакомления с байт кодом являются пара jasper – jasmín.

# Дизассемблирование с Jasper

```
I:\helloworld> java -jar Jasper.jar HelloWorld.class
Reading: HelloWorld.class
Generated: HelloWorld.j

I:\helloworld>
```



Для дизассемблирования можно использовать Jasper. В качестве параметра передаётся название class файла. Создаётся текстовый файл с расширением j.

# Байт код

```
HelloWorld.j - Notepad
File Edit Format View Help
.source      Helloworld.java
.class      public Helloworld
.super      java/lang/Object


.method      public <init>()V
  .limit stack 1
  .limit locals 1
  .line      1
  aload_0
  invokespecial java/lang/Object/<init>()V
  return
.end method

.method      public static main([Ljava/lang/String;)V
  .limit stack 2
  .limit locals 1
  .line      5
  getstatic   java/lang/System/out Ljava/io/PrintStream;
  ldc         "Hello world!"
  invokevirtual java/io/PrintStream/println(Ljava/lang/String;)V
  .line      6
  return
.end method
```

```
HelloWorld.j - Notepad
File Edit Format View Help
.source      Helloworld.java
.class      public Helloworld
.super      java/lang/Object

.method      public <init>()V
  .limit stack 1
  .limit locals 1
  .line      1
  aload_0
  invokespecial java/lang/Object/<init>()V
  return
.end method

.method      public static main([Ljava/lang/String;)V
  .limit stack 2
  .limit locals 1
  .line      5
  getstatic   java/lang/System/out Ljava/io/PrintStream;
  ldc         "Hellooooooooooooooooooooooooooooooooooooooooo world!"
  invokevirtual java/io/PrintStream/println(Ljava/lang/String;)V
  .line      6
  return
.end method
```

 После дизассемблирования в байт код можно внести изменения чтобы потом заново получить class файл.



# КОДЫ ТИПОВ

|   |         |
|---|---------|
| B | byte    |
| C | char    |
| D | double  |
| F | float   |
| I | int     |
| J | long    |
| L | object  |
| S | short   |
| Z | boolean |
| [ | array   |

# jasmin

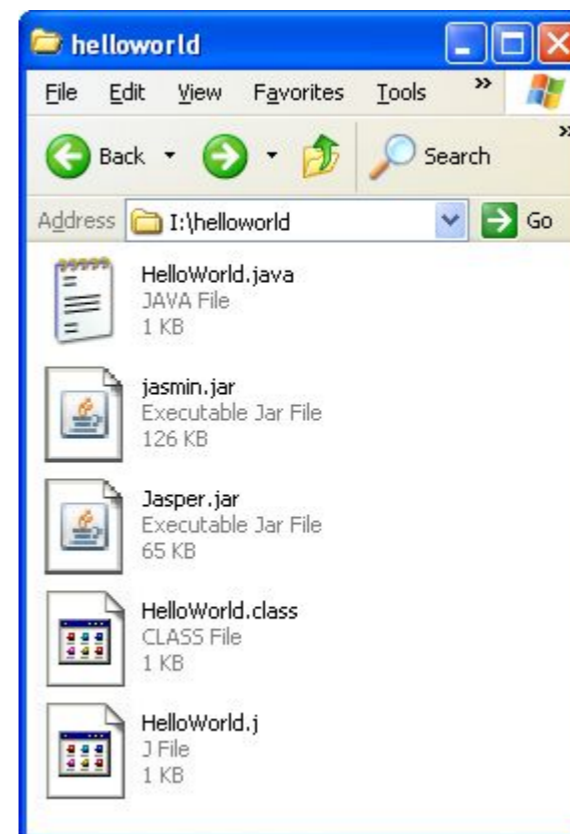
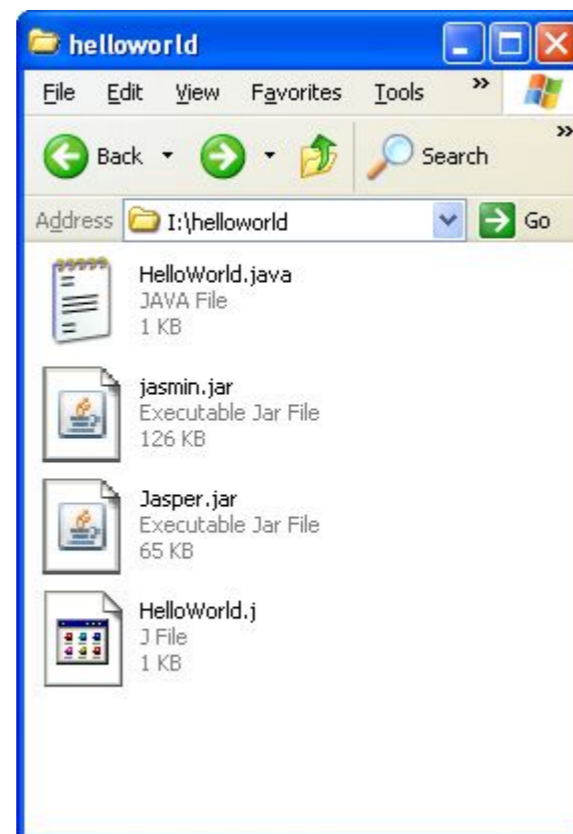
```
I:\helloworld>java -jar jasmin.jar HelloWorld.j
```

```
Generated: HelloWorld.class
```

```
I:\helloworld>java HelloWorld
```

```
Hellooooooooooooooooooooooooooooooooooooo World!
```

```
I:\helloworld>
```



Для получения class файла можно использовать `jasmin`. В качестве параметра передаётся название `j` файла. Создаётся class файл.