

ВВЕДЕНИЕ В JAVA



Программирование на Java

ТЕХНОЛОГИЯ JAVA: ИСТОРИЯ

- Первая версия языка была официально опубликована в 1995 году
- Основная причина – потребность в независимом от платформы языке



ORACLE®

ЦЕЛИ СОЗДАНИЯ НОВОГО ЯЗЫКА:

- Повышение скорости разработки;
- Устранение ошибок других языков ;
- Переносимость кода;
- Многопоточность;
- Загрузка классов динамически, когда они на самом деле необходимы;
- Поддержка изменения программы динамически во время выполнения загрузки классов из разных источников;
- Обеспечение более высокого уровня безопасности.

СРЕДСТВА ДОСТИЖЕНИЯ ЦЕЛИ:

- The Java Virtual Machine (JVM™)
 - программы, написанные на языке Java, выполняются под управлением специальной программы – виртуальной Java-машины
- Garbage collection
 - сборщик мусора
- The Java Runtime Environment (JRE)
- JVM tool interface

THE JAVA VIRTUAL MACHINE

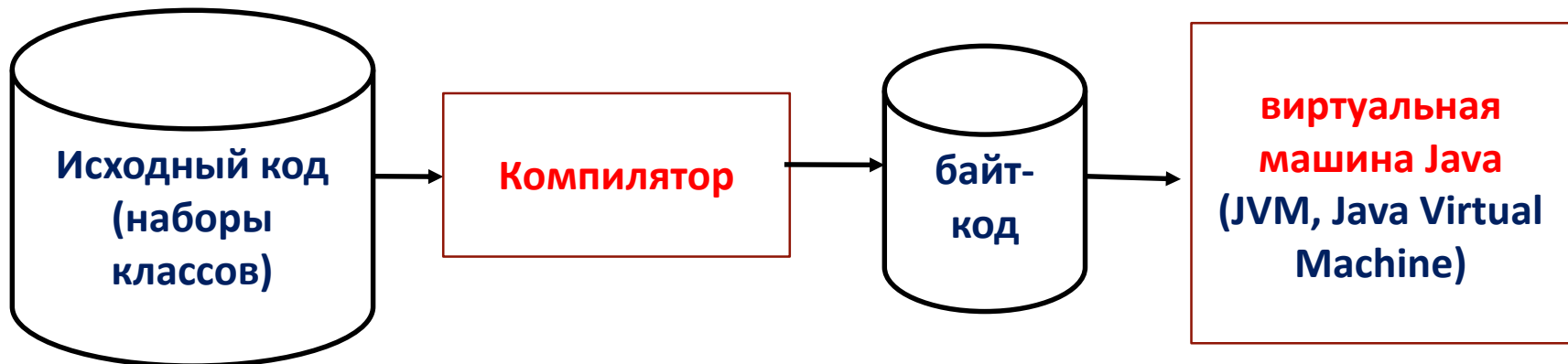
Java Virtual Machine (JVM) – программа виртуальной Java-машины, скомпилированная в коды соответствующего процессора – система времени выполнения Java.

- Взаимодействует с операционной системой, обеспечивая доступ к файлам или поддержку графики;
- Исполняет байт-код;
- Обеспечивает автоматическое высвобождение памяти, занятой ненужными объектами (сборка мусора).

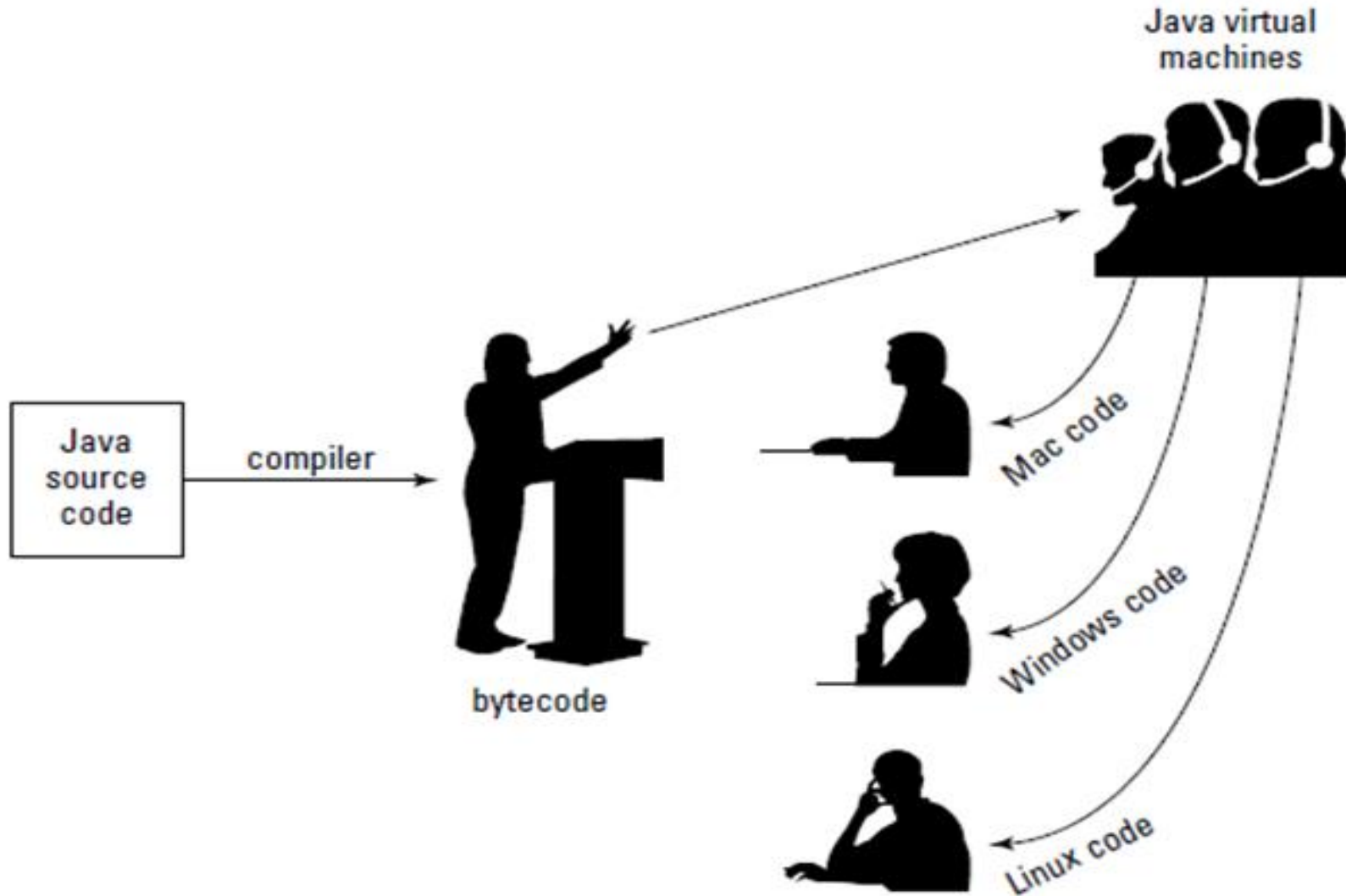
Выполнение Java-программы

Компиляция

Компоновка,
выполнение



JAVA VIRTUAL MACHINE

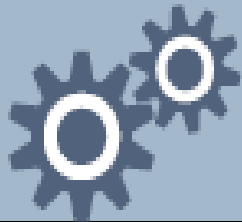


РЕАЛИЗАЦИЯ JVM

Java Technology Application

JVM

OS



Hardware Platform

ПЛАТФОРМА JAVA

- Платформа Java – это программное обеспечение, которое используется для создания и выполнения Java–программ, также коллекция компонентов готовых к использованию, которые существенно упрощают процесс разработки приложений. Эти компоненты сгруппированы в библиотеки классов и интерфейсов. Такие библиотеки называются пакетами (packages).
- Замечание: Платформа – это совокупность аппаратного и программного (ОС) обеспечения (Microsoft Windows, Linux, Solaris OS и Mac OS). Платформа Java отличается от этих платформ, так как является лишь программным обеспечением.

ПЛАТФОРМА JAVA

Платформа Java распространяется в двух вариантах – JDK(Java Development Kit) и JRE (Java Runtime Environment).

- **JDK** - средство разработчика, состоящее в основном из утилит (например, java compiler), стандартных библиотек классов и демонстрационных примеров. JDK является базовым средством разработки приложений. Оно не содержит никаких текстовых редакторов, а оперирует только с уже существующими java-файлами с помощью утилит командной строки.
- **JRE** (среда выполнения Java) - это минимальная реализация виртуальной машины, необходимая для исполнения Java-приложений, без компилятора и других средств разработки. Если пользователь хочет только запускать программы, это именно то, что ему нужно. JRE включен в состав JDK.

ПЛАТФОРМА JAVA

- **Варианты распространения платформы Java:**
 - **JDK (Java Development Kit);**
 - **JRE (Java Runtime Environment).**
- **Версии JDK (Java Development Kit) :**
 - **Java SE** – комплект **Java Standard Edition** – для написания программного обеспечения для обычных компьютеров;
 - **Java EE**– комплект **Java Enterprise Edition** – для написания серверного программного обеспечения;
 - **Java ME** – комплект **Java Micro Edition** – для написания программного обеспечения для устройств, обладающих малыми ресурсами, например, сотовых телефонов, микроконтроллеров.

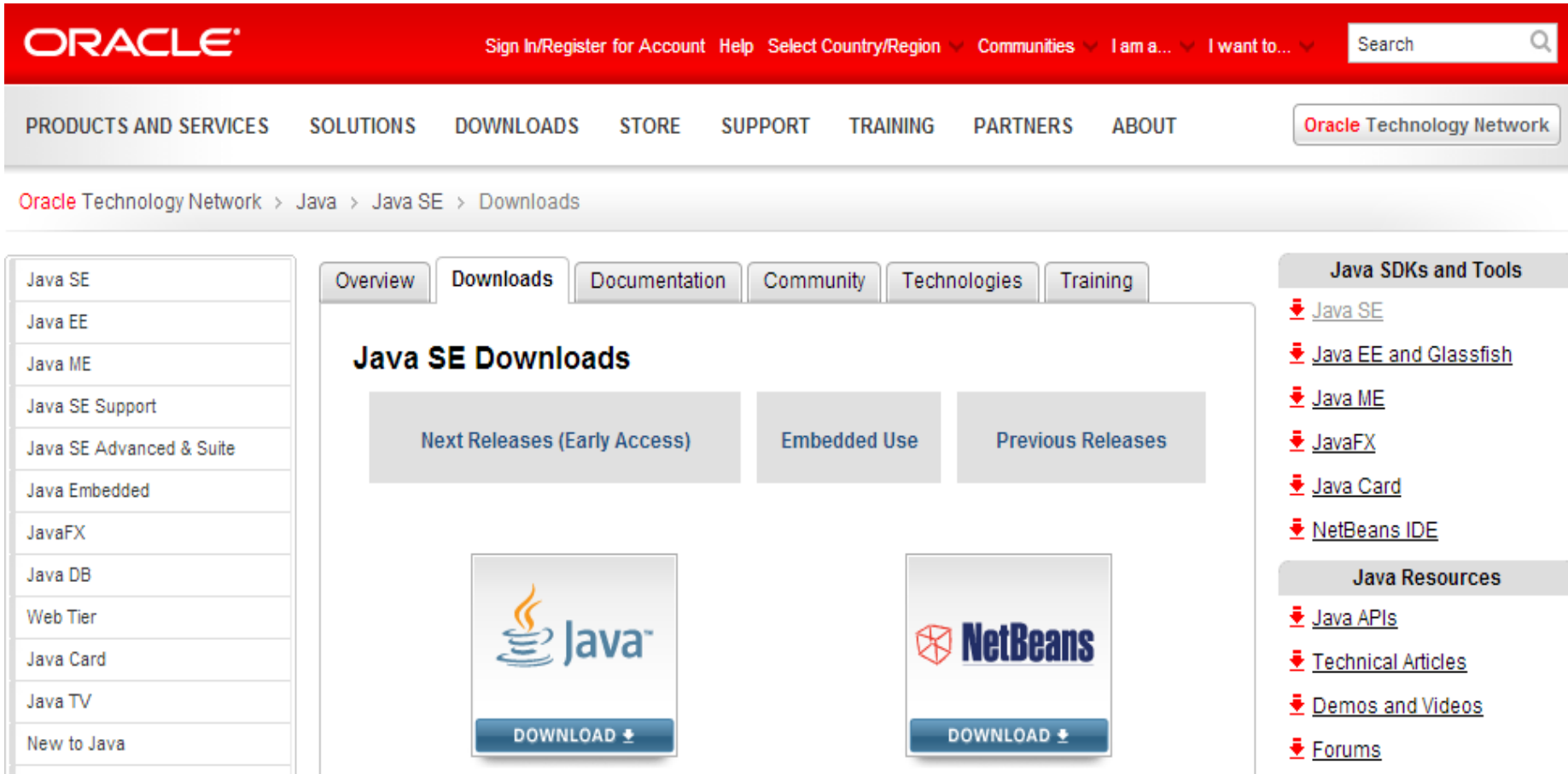
ПЛАТФОРМА JAVA ¹²

- **Наиболее часто используемые пакеты библиотеки классов:**
 - **java.lang** - базовые классы, необходимые для работы любого приложения (название - сокращение от language);
 - **java.util** - многие полезные вспомогательные классы;
 - **java.applet** - классы для создания апплетов;
 - **java.awt, java.swing** - библиотека для создания графического интерфейса пользователя(GUI);
 - **java.awt.image** - дополнительные классы для работы с изображениями;
 - **java.io** - работа с потоками данных (streams) и с файлами;
 - **java.net** - работа с сетью.

JAVA DEVELOPMENT KIT (JDK)

- Ссылка для скачивания JDK:

<http://www.oracle.com/technetwork/java/javase/downloads/>



The screenshot shows the Oracle Java SE Downloads page. At the top, there is a red header with the Oracle logo and navigation links. Below the header is a navigation bar with links for PRODUCTS AND SERVICES, SOLUTIONS, DOWNLOADS, STORE, SUPPORT, TRAINING, PARTNERS, and ABOUT. A search bar is located on the right side of the header. The main content area is titled "Java SE Downloads" and features three tabs: Overview, Downloads (selected), and Documentation. Below the tabs are three buttons: "Next Releases (Early Access)", "Embedded Use", and "Previous Releases". There are two large download buttons: one for "Java" and one for "NetBeans". On the left side, there is a sidebar with a list of Java SE versions and products. On the right side, there are two sections: "Java SDKs and Tools" and "Java Resources", each containing a list of links.

ORACLE Sign In/Register for Account Help Select Country/Region Communities I am a... I want to... Search


PRODUCTS AND SERVICES SOLUTIONS DOWNLOADS STORE SUPPORT TRAINING PARTNERS ABOUT Oracle Technology Network


Oracle Technology Network > Java > Java SE > Downloads

Overview Downloads Documentation Community Technologies Training

Java SE Downloads

Next Releases (Early Access) Embedded Use Previous Releases

 **DOWNLOAD** ↓

 **DOWNLOAD** ↓

Java SDKs and Tools

- Java SE
- Java EE and Glassfish
- Java ME
- JavaFX
- Java Card
- NetBeans IDE

Java Resources

- Java APIs
- Technical Articles
- Demos and Videos
- Forums
















Java SE
Java EE
Java ME
Java SE Support
Java SE Advanced & Suite
Java Embedded
JavaFX
Java DB
Web Tier
Java Card
Java TV
New to Java

JAVA DEVELOPMENT KIT (JDK)

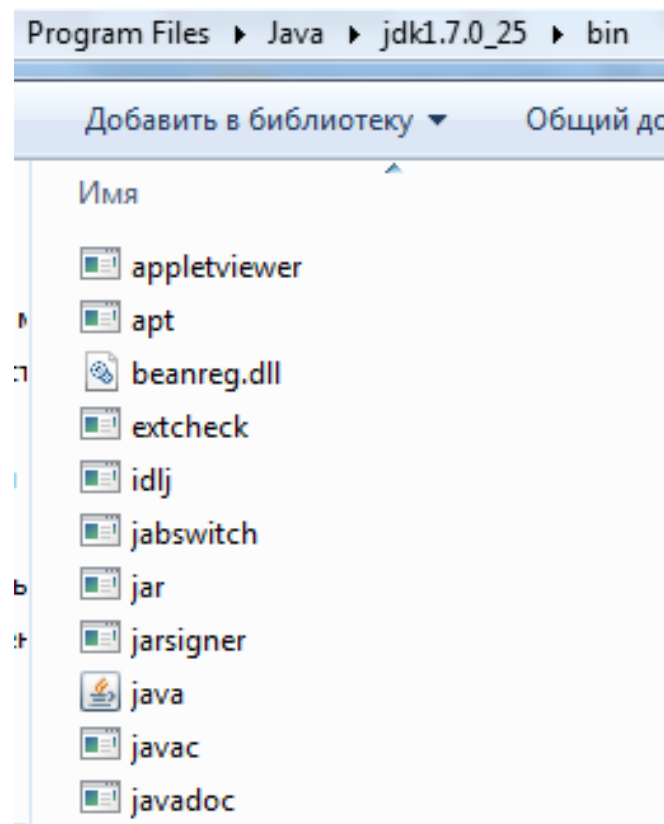
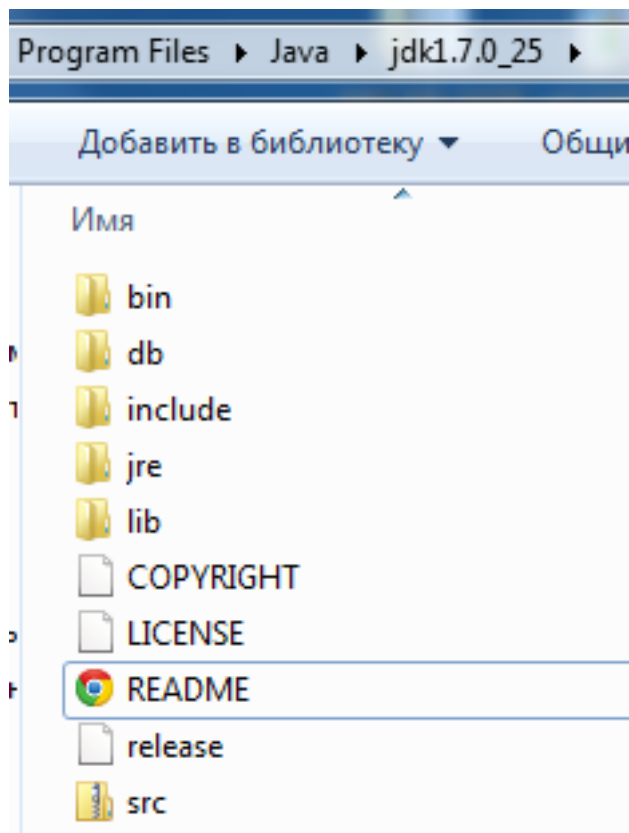
Java SE Development Kit 7u25

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

Accept License Agreement Decline License Agreement

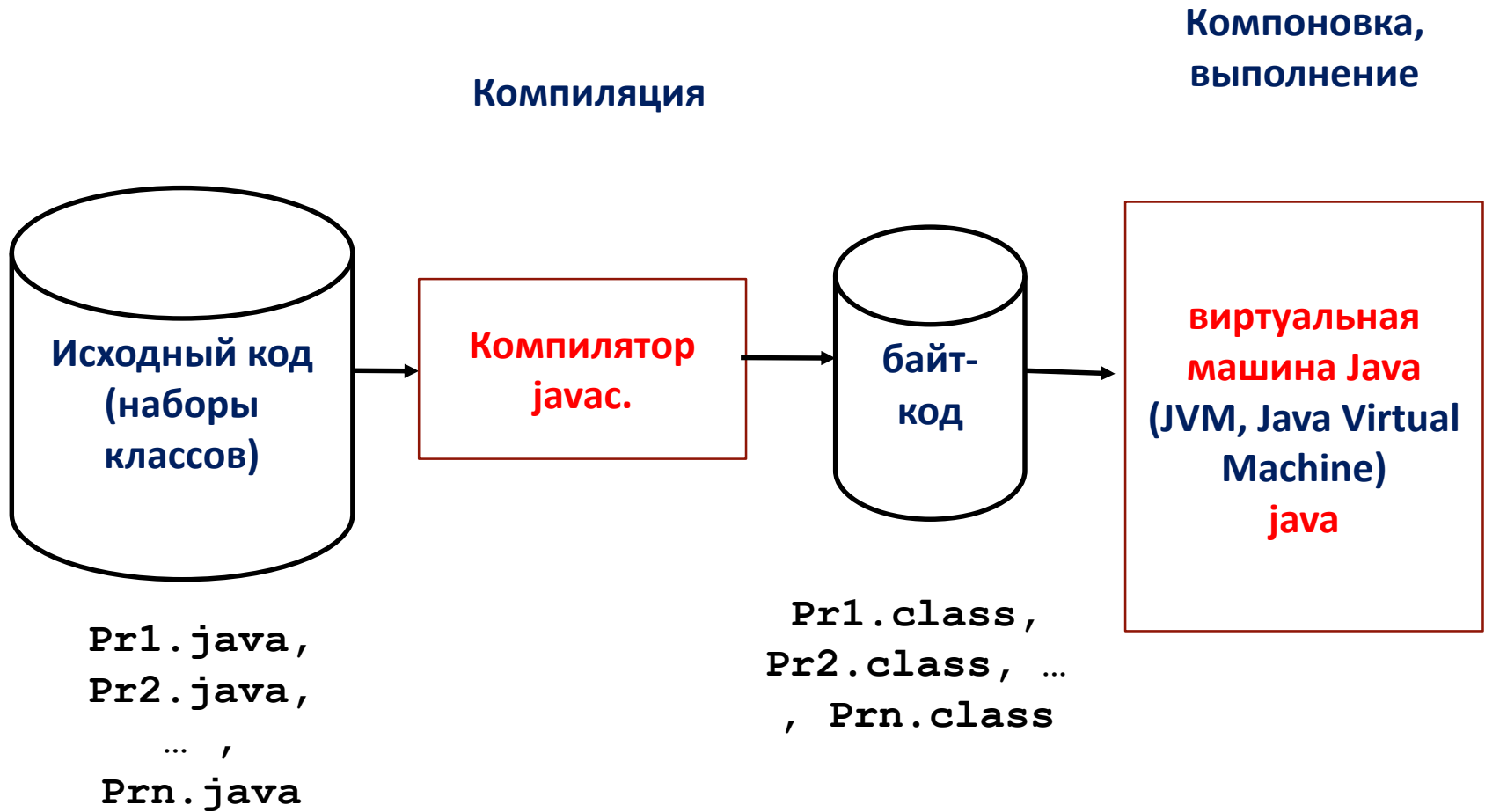
Product / File Description	File Size	Download
Linux x86	80.38 MB	 jdk-7u25-linux-i586.rpm
Linux x86	93.12 MB	 jdk-7u25-linux-i586.tar.gz
Linux x64	81.46 MB	 jdk-7u25-linux-x64.rpm
Linux x64	91.85 MB	 jdk-7u25-linux-x64.tar.gz
Mac OS X x64	144.43 MB	 jdk-7u25-macosx-x64.dmg
Solaris x86 (SVR4 package)	136.02 MB	 jdk-7u25-solaris-i586.tar.Z
Solaris x86	92.22 MB	 jdk-7u25-solaris-i586.tar.gz
Solaris x64 (SVR4 package)	22.77 MB	 jdk-7u25-solaris-x64.tar.Z
Solaris x64	15.09 MB	 jdk-7u25-solaris-x64.tar.gz
Solaris SPARC (SVR4 package)	136.16 MB	 jdk-7u25-solaris-sparc.tar.Z
Solaris SPARC	95.5 MB	 jdk-7u25-solaris-sparc.tar.gz
Solaris SPARC 64-bit (SVR4 package)	23.05 MB	 jdk-7u25-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit	17.67 MB	 jdk-7u25-solaris-sparcv9.tar.gz
Windows x86	89.09 MB	 jdk-7u25-windows-i586.exe
Windows x64	90.66 MB	 jdk-7u25-windows-x64.exe

JAVA DEVELOPMENT KIT (JDK)



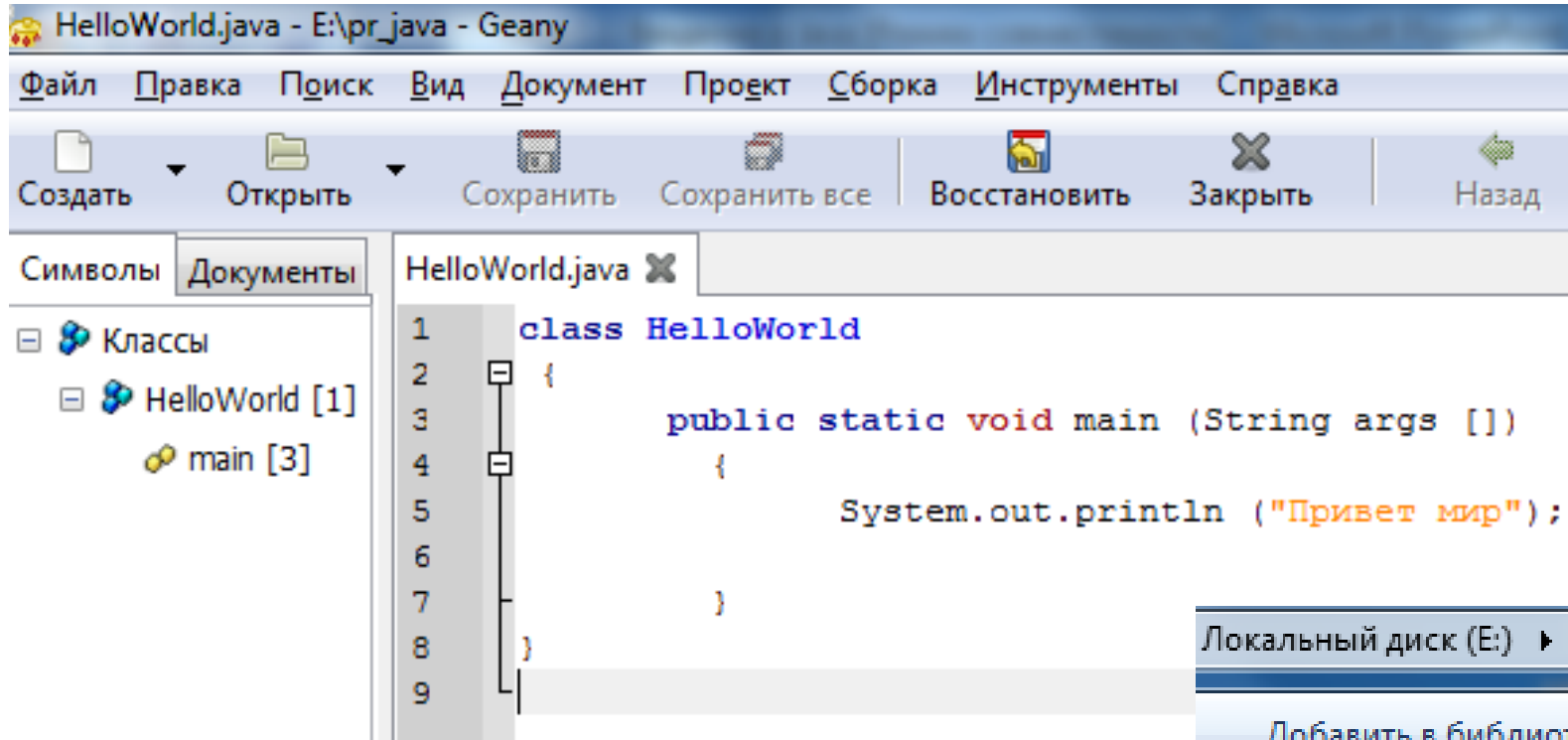
- **javac** – компилятор для перевода java-программы в байт-код, понятный виртуальной машине;
- **java** – утилита для запуска байт-кода java-программы ;
- **jar** – утилита для создания java-архивов

СТРУКТУРА JAVA-ПРОГРАММЫ

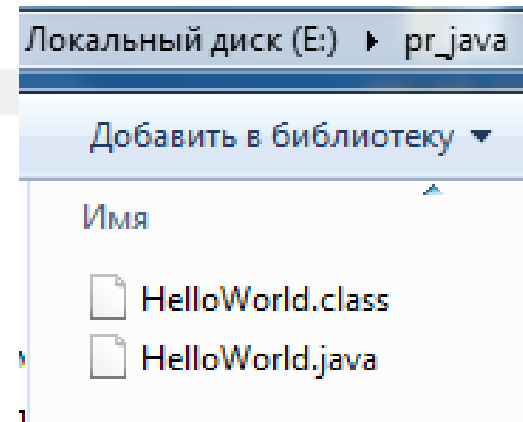


СТРУКТУРА JAVA-ПРОГРАММЫ

Пример программы:



```
1 class HelloWorld
2 {
3     public static void main (String args [])
4     {
5         System.out.println ("Привет мир");
6     }
7 }
8
9
```



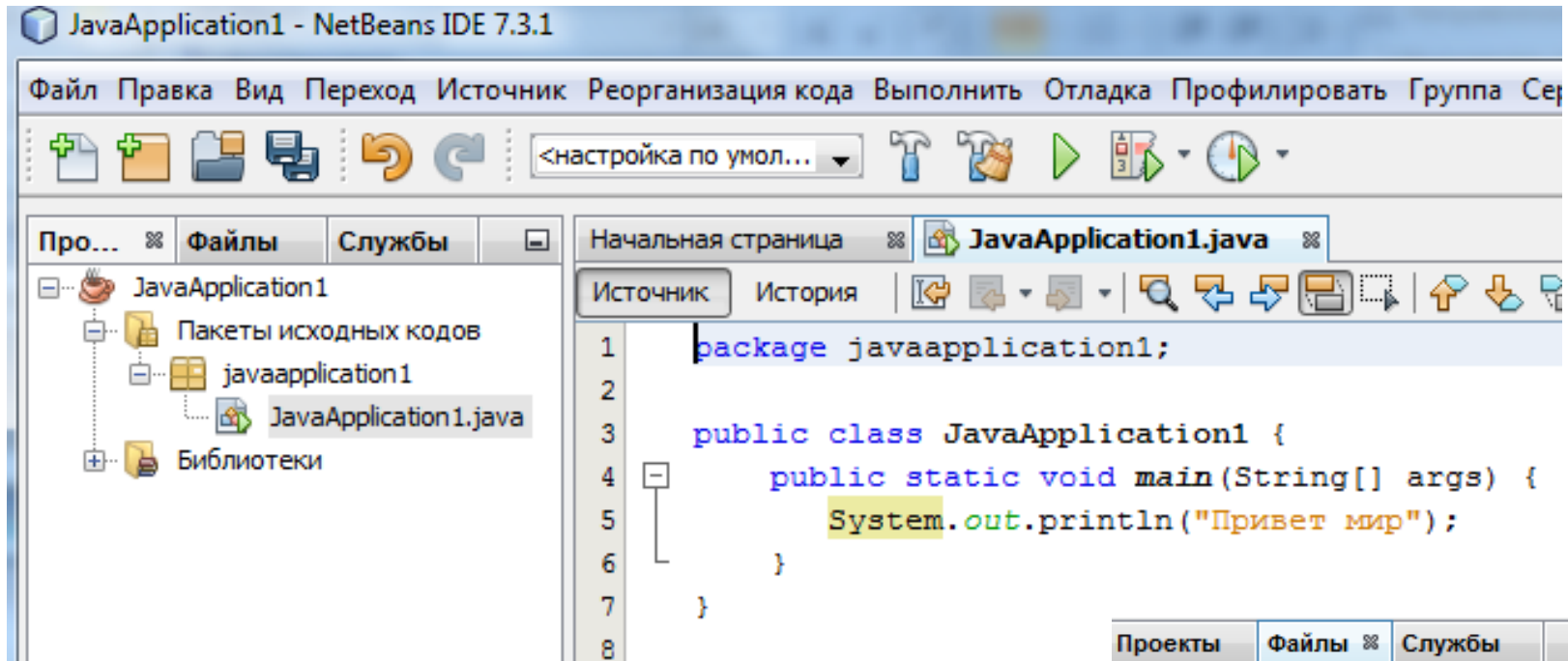
```
E:\pr_java>javac HelloWorld.java
E:\pr_java>java HelloWorld
Привет мир
```

СТРУКТУРА JAVA-ПРОГРАММЫ

Комментарии – пояснения по ходу программы.

- `//` - однострочный комментарий;
- `/*` - многострочный комментарий `*/`;
- `/**` - комментарии записываются в специальный файл формата `html` `*/` специальной программой `Javadoc` входящая в состав `JDK`.

Пример программы:



JavaApplication1 - NetBeans IDE 7.3.1

Файл Правка Вид Переход Источник Реорганизация кода Выполнить Отладка Профилировать Группа Сер

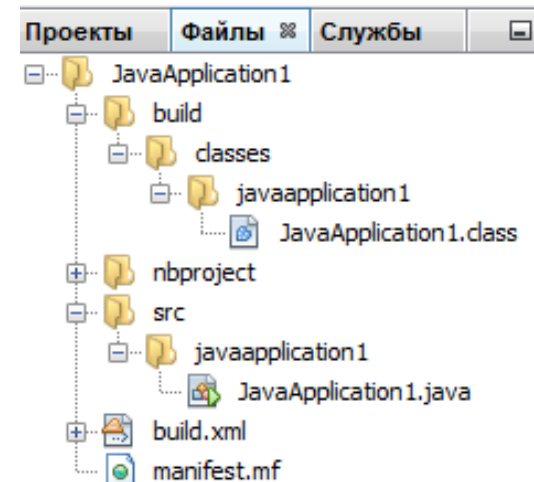
Про... Файлы Службы

Начальная страница JavaApplication1.java

```
1 package javaapplication1;
2
3 public class JavaApplication1 {
4     public static void main(String[] args) {
5         System.out.println("Привет мир");
6     }
7 }
8
```

Вывод - JavaApplication1 (run)

```
run:
Привет мир
СБОРКА УСПЕШНО ЗАВЕРШЕНА (общее время: 1 секунда)
```



Проекты Файлы Службы

- JavaApplication1
 - build
 - classes
 - javaapplication1
 - JavaApplication1.class
 - nbproject
 - src
 - javaapplication1
 - JavaApplication1.java
 - build.xml
 - manifest.mf

КОНСТАНТЫ

- Целые константы:

- **в десятичной форме:** +5, -7, 12345678 ;
- **в восьмеричной форме**, начиная с нуля: 027, -0326, 0777 ; в записи таких констант недопустимы цифры 8 и 9;
- **в шестнадцатеричной форме**, начиная с нуля и латинской буквы x или X: 0xff0a, 0xFC2D, 0x45a8, 0X77FF ; здесь строчные и прописные буквы не различаются.

- Действительные константы:

- **с фиксированной точкой:** 37.25, -128.678967, +27.035 ;
- **с плавающей точкой:** 2.5e34, -0.345e-25, 37.2E+4 ;

- **Символьные константы:**

- **Печатные символы** можно записать в апострофах: 'a', 'N' ;
- **Управляющие символы** записываются в апострофах с обратной наклонной чертой:
 - '\n' — символ перевода строки;
 - '\t' — символ горизонтального отступа;
 - и др.
- **Код любого символа с десятичной кодировкой от 0 до 255:**
 - '\123' — буква S , '\346' — буква Ж в кодировке CP1251
- **Код любого символа в кодировке Unicode:**
 - '\u0053' — буква S , '\u0416' — буква Ж .

- **Строковые константы:**

"Это строка\nс переносом"

"\"Спартак\" — Чемпион!"

"Сцепление" + "строк" дает в результате строку "Сцепление
строк" .

ИДЕНТИФИКАТОРЫ

Соглашения о регистре букв в идентификаторах:

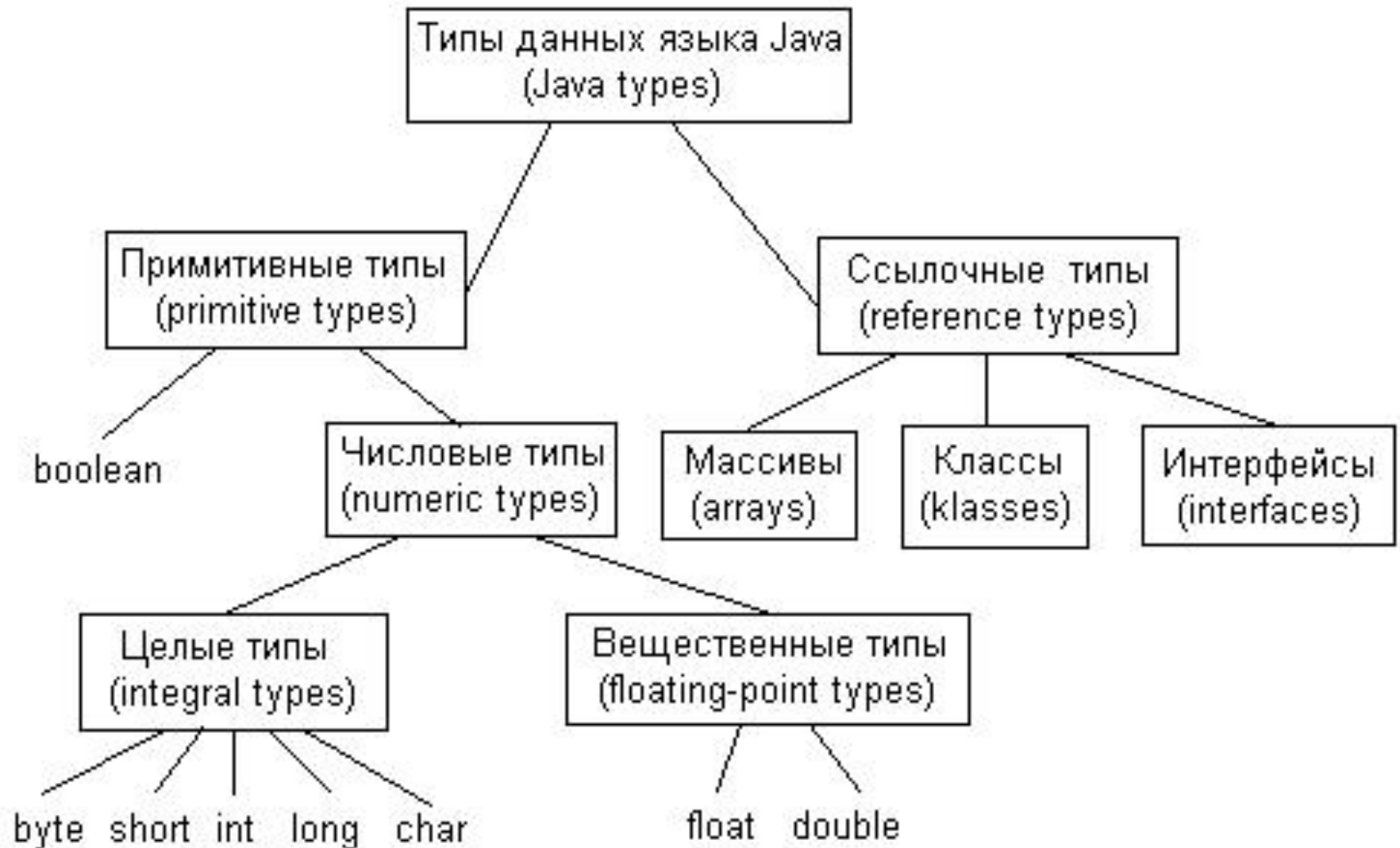
- **Имена примитивных типов** следует писать в нижнем регистре (строчными буквами).
Например, `int`, `float`, `boolean`
- **Имена ссылочных типов** следует начинать с заглавной (большой) буквы. Если имя составное, новую часть имени также начинают с заглавной буквы.
Например, `Object`, `Float`, `Collection`, `JButton`, `JTextField`
- **Для переменных и методов имена**, состоящие из одного слова, следует писать в нижнем регистре. Если имя составное, новую часть имени начинают с заглавной буквы.
Например, `i`, `j`, `obj1`, `myVar`, `jButton2`, `jTextField2.getText()`
- **Имена констант** следует писать в верхнем регистре (большими буквами), разделяя входящие в имя составные части символом подчёркивания “_”.
Например, `MIN_VALUE`, `MAX`

ПРИМИТИВНЫЕ И ССЫЛОЧНЫЕ ТИПЫ ДАННЫХ

Java все типы делятся на примитивные и ссылочные.

- Примитивными типами называются такие, для которых данные содержатся в одной ячейке памяти, и эта ячейка не имеет вложенных ячеек.
- Ссылочными типами называются такие, для которых в ячейке памяти (ссылочной переменной) содержатся не сами данные, а только адреса этих данных, то есть ссылки на данные. При присваивании в ссылочную переменную заносится новый адрес, а не сами данные. Но непосредственного доступа к адресу, хранящемуся в ссылочных переменных, нет. Это сделано для обеспечения безопасности работы с данными – как с точки зрения устранения непреднамеренных ошибок, характерных для работы с данными по их адресам в языках C/C++/PASCAL, так и для устранения возможности намеренного взлома информации.
- Если ссылочной переменной не присвоено ссылки, в ней хранится нулевой адрес, которому дано символическое имя **null**.

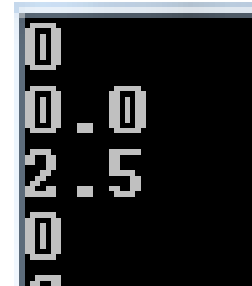
ПРИМИТИВНЫЕ И ССЫЛОЧНЫЕ ТИПЫ ДАННЫХ



Объявление переменных

```
int i1 = i2 = 5; // Ошибка! Запрещено в Java
class HelloWorld
{
    public static int i;
    public static double d1, d2 = 2.5;
    public static void main (String args [])
    {
        int k = 0;

        System.out.println(i);
        System.out.println(d1);
        System.out.println(d2);
        System.out.println(k);
    }
}
```



ПРИМИТИВНЫЕ ТИПЫ ДАННЫХ И ОПЕРАЦИИ

- Логический тип: `boolean a = true;`
`boolean b = false;`

Логические операции и операции отношения

Операция	Название	Пример
<code>&&</code>	логическое "И" (and)	<code>a&&b</code>
<code> </code>	логическое "ИЛИ" (or)	<code>a b</code>
<code>^</code>	логическое "исключающее ИЛИ" (xor)	<code>a^b</code>
<code>!</code>	логическое "НЕ" (not)	<code>!a</code>

a	0	0	1	1
b	0	1	0	1
Выражение	Значения			
<code>a&&b</code>	0	0	0	1
<code>a b</code>	0	1	1	1
<code>a^b</code>	0	1	1	0
<code>!a</code>	1	1	0	0

Операция	Название	Пример
<code>==</code>	равно	<code>a==b</code>
<code>!=</code>	не равно	<code>a!=b</code>
<code>></code>	больше	<code>a>b</code>
<code><</code>	меньше	<code>a<b</code>
<code>>=</code>	больше или равно	<code>a>=b</code>
<code><=</code>	меньше или не равно	<code>a<=b</code>
<code>instanceof</code>	Принадлежность объекта классу	<code>obj instanceof MyClass</code>

ПРИМИТИВНЫЕ ТИПЫ ДАННЫХ И ОПЕРАЦИИ

- Целые типы

Тип	Разрядность (байт)	Диапазон
byte	1	от -128 до 127
short	2	от -32768 до 32767
int	4	от -2147483648 до 2147483647
long	8	от -9223372036854775808 до 9223372036854775807
char	2	от '\u0000' до '\uFFFF', в десятичной форме от 0 до 65535

- Арифметические операции: + - / * % ++ --
- Приведение типов:

```
class HelloWorld
{
    public static void main (String [] args)
    {
        byte b1 = 50, b2 = -99;
        short k = b1 + b2; // Неверно!
        //short k = (short) (b1 + b2); // Верно!!!
        System.out.println("k=" + k);
    }
}
```

HelloWorld.java:4: error: possible loss of precision

```
short k = b1 + b2; //    !
```

^

required: short

found: int

1 error

ПРИМИТИВНЫЕ ТИПЫ ДАННЫХ И ОПЕРАЦИИ

- Вещественные типы:

Тип	Разрядность	Диапазон	Точность
float	4	$3,4e-38 < x < 3,4e38$	7—8 цифр
double	8	$1,7e-308 < x < 1,7e308$	17 цифр

- Операции присваивания:

- Простая операция присваивания: =

- Составные операции присваивания:

$+=$, $-=$, $*=$, $/=$, $\%=$, $\&=$, $|=$, $\wedge=$, $\ll=$, $\gg=$

- Схема действия составных операций присваивания:

$x \text{ op} = a$ эквивалентно $x = (\text{тип } x)(x \text{ op } a)$.

```
short ind = 1; ind += 7.8; // получим 8
```

```
ind = (short)(ind + 7.8); // получим 8
```

КОНСОЛЬНЫЙ ВВОД-ВЫВОД

- Потоковые объекты:

- `System.in` – объект класса `InputStream`;
- `System.out` – объект класса `OutputStream`;

- Вывод данных: МЕТОД `System.out.println()`;

```
System.out.println("Hello World"); // вывод строки
```

```
System.out.println(245); // вывод числовой константы
```

```
// Вывод строковой или численной переменной
```

```
int var = 245;          String text = "Hello World";
```

```
System.out.println(text); System.out.println(var);
```

```
// Вывод произвольного сочетания переменных и констант
```

```
System.out.println("Hello" + "number = " + var);
```

```
// Вывод вещ. чисел с ограничением числа знаков после запятой
```

```
double f = 2.567432;
```

```
NumberFormat nf = NumberFormat.getInstance();
```

```
nf.setMaximumFractionDigits(2);
```

```
System.out.println(nf.format(f));
```

КОНСОЛЬНЫЙ ВВОД-ВЫВОД

- Ввод данных с клавиатуры (вариант 1):

```
// установить связь с вводимыми с  
клавиатуры данными
```

```
import java.io.*
```

```
...
```

```
BufferedReader bReader = new  
BufferedReader (new  
InputStreamReader (System.in) );
```

```
// ввод строки данных
```

```
String cStr;  
cStr = bReader.readLine();
```

```
// преобразование строки данных
```

```
int iNum = Integer.parseInt(cStr);  
float fNum = Float.parseFloat(cStr);  
double dNum = Double.parseDouble(cStr);
```

КОНСОЛЬНЫЙ ВВОД-ВЫВОД

- Ввод данных с клавиатуры (вариант 2): Класс Scanner:

```
import java.util.Scanner; // импортируем класс
public class Main
{
    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);
        // создаём объект класса Scanner
        int i = 2;
        System.out.print("Введите целое число: ");
        i = sc.nextInt(); // считывает целое число
        с потока ввода и сохраняем в переменную
        System.out.println(i*2);
    }
}
```

КОНСОЛЬНЫЙ ВВОД-ВЫВОД

- Класс Scanner:

```
import java.util.Scanner; // импортируем класс
public class Main
{
    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in); // создаём объект
                                                // класса Scanner

        double i = sc.nextDouble(); // если ввести букву s, то
                                    // случится ошибка во время
                                    // выполнения программы

        System.out.println(i/3);
    }
}
```