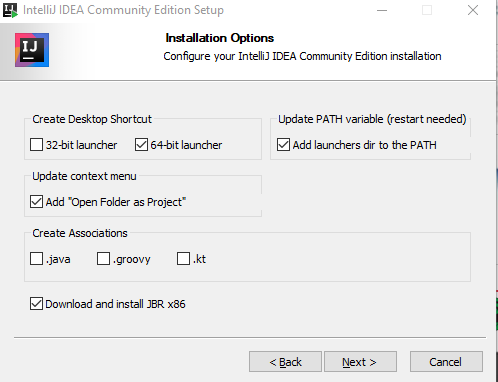
**Урок 0. Установка Java и IntelliJ IDEA**

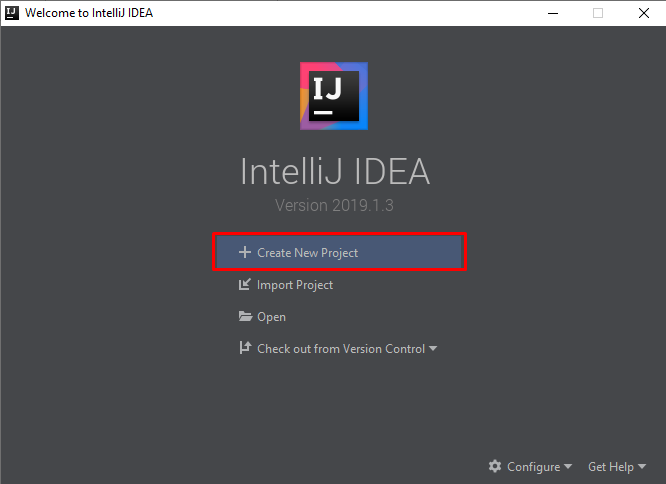
Для того чтобы начать программировать на Java необходимо установить средства разработки и среду разработки.

1. Средство разработки(JDK). Можно найти самостоятельно, либо скачать отсюда <https://www.oracle.com/technetwork/java/javase/downloads/jdk12-downloads-5295953.html> . Устанавливаем.
2. Среда разработки(Intellij IDEA) <https://www.jetbrains.com/idea/download/#section=windows>. Выбирать нужно Community Edition. В настройках отметить эти чекбоксы 

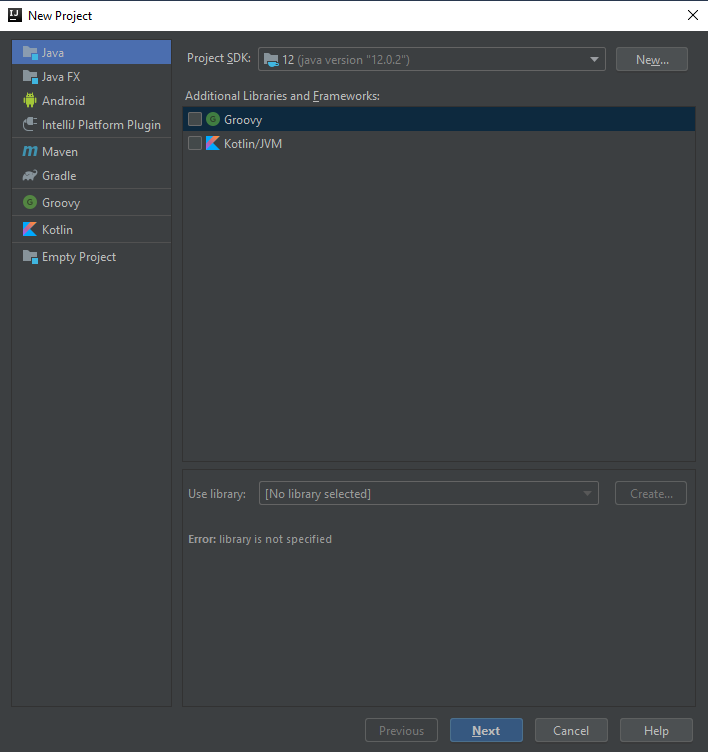
Устанавливаем.

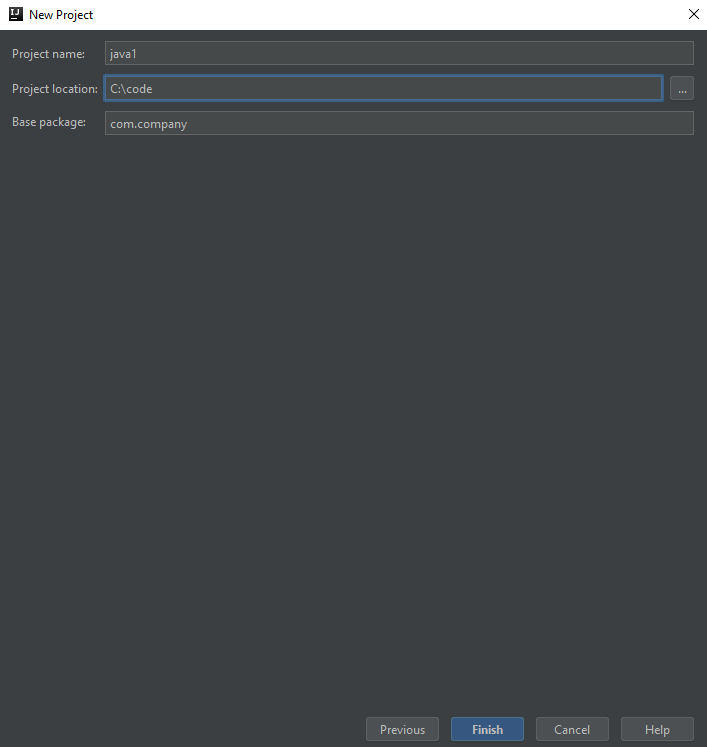
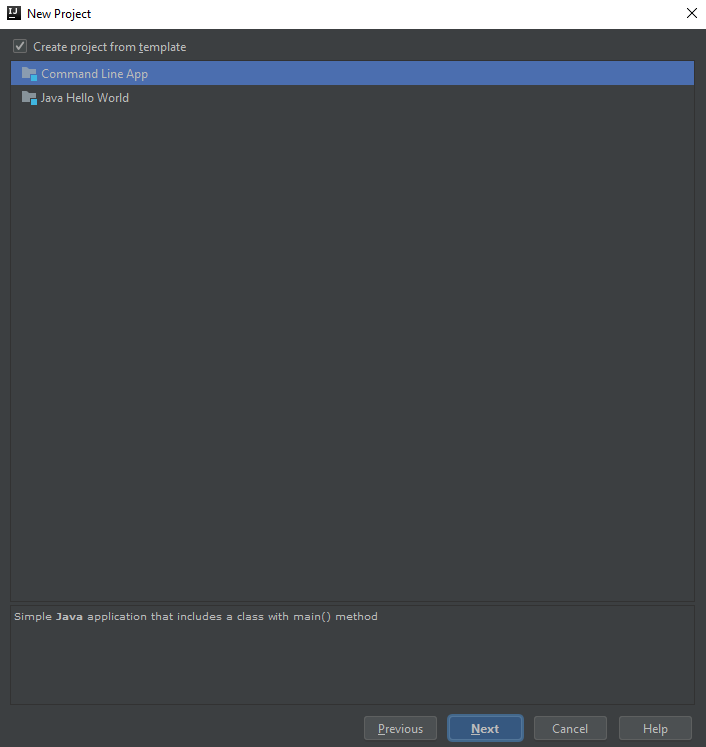
1. Создаём папку на диске С под названием “code”, в ней мы будем хранить свои проекты.

Пример создания проекта

Запускаем Intellij IDEA. Создаём новый проект 

Сверху выбираем 12 версию JDK. Дополнительные чекбоксы не выбираем. Нажимаем далее.



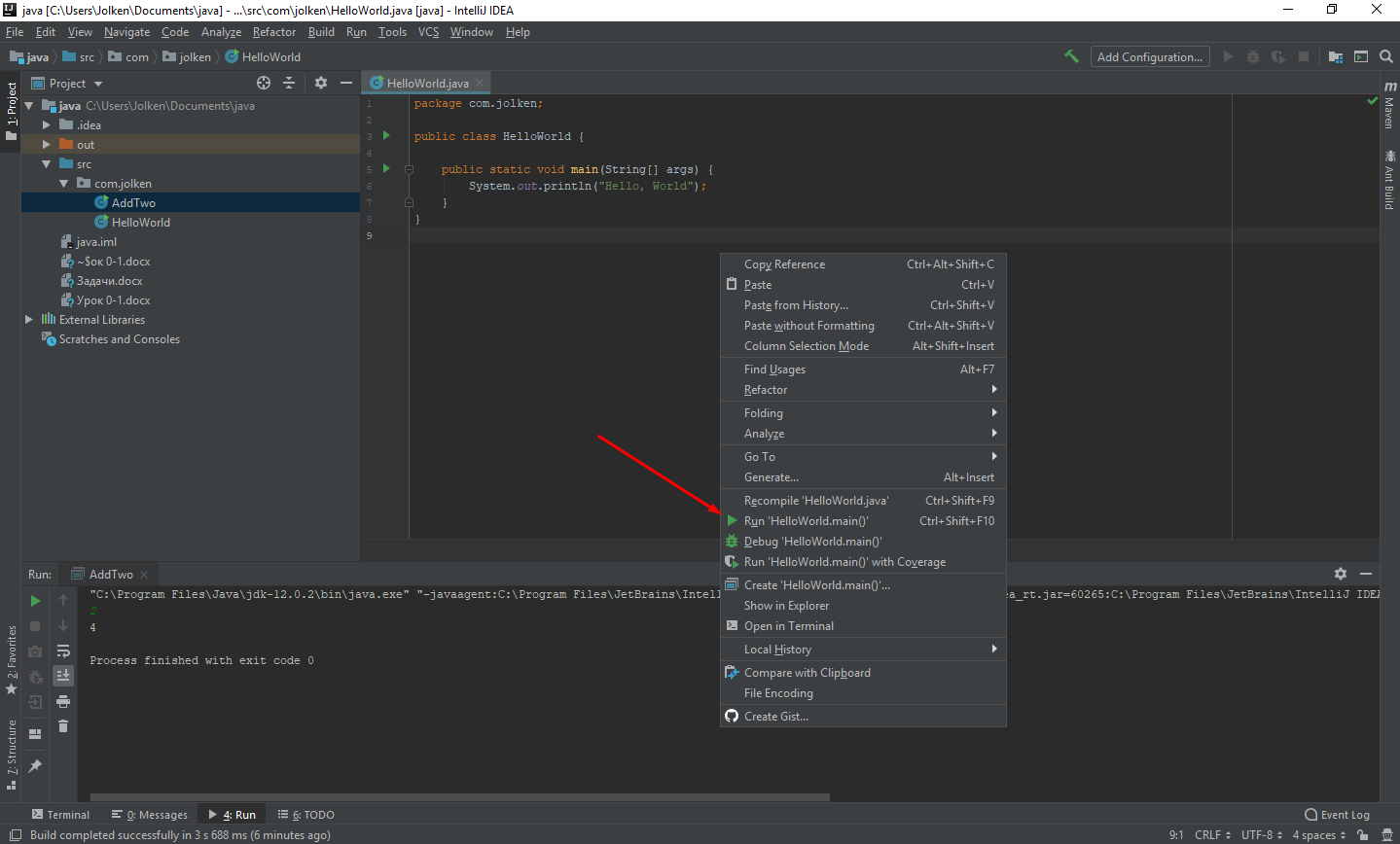


Вставляем этот код.

package com.jolken;  
  
public class HelloWorld {  
  
 public static void main(String[] args) {  
 System.*out*.println("Hello, World");  
 }  
}

|  |  |
| --- | --- |
|  |  |

Запускаем программу нажимая ПКМ и в меню выбираем “Run имя\_класса.main()”, лиюо нажимаем Ctrl+Shift+F10



Снизу можно увидеть то, что вывела программа.

**Глава 1. Основы Java(стр. 16-9999999)**

Язык программирования Java является полностью объектно-ориентированным. Это означает, что для составления даже самой простой программы необходимо описать класс.

|  |  |
| --- | --- |
| 1  2  3  4  5 | class Intro {  public static void main(String[] args) {  System.out.println("Мы программируем на Java!");  }  } |
|  |  |

Класс – абстрактное понятие типа данных, имеющего свои функции(методы) и переменные. Пример класса:

Животное. Переменные: рост, масса, возраст, пол. Функции(методы): кушать, пить, размножаться.

Наследование – создание нового класса на основе другого. Пример наследования:

Класс Собака наследует класс Животное, дополняя его функции функцией лаять.

Пока что этого достаточно для написания программ.

В рассматриваемом примере программный код состоит всего из одной команды

System.out.println("Мы программируем на Java!"). Команда заканчивается точкой

с запятой — это стандарт для Java. Командой с помощью встроенного метода

println() на консоль (по умолчанию консолью является экран) выводится со-

общение "Мы программируем на Java!".

Java существует три типа комментариев.

Однострочный комментарий. Такой комментарий начинается

1. с двойной косой черты (символ //). Все, что находится в строке кода справа от двойной косой черты, компилятором игнорируется.

2. Многострочный комментарий. Такой комментарий начинается последовательностью символов /\* и заканчивается последовательностью символов \*/. Все,

что находится между символами /\* и \*/, компилятором игнорируется.

3. Многострочный комментарий документационной информации. Начинается

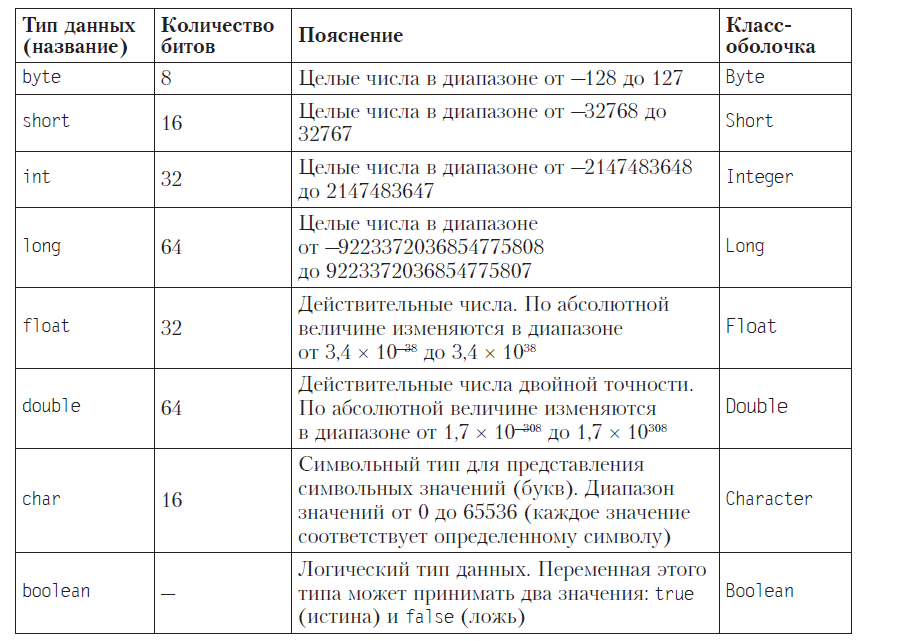
последовательностью символов /\*\* и заканчивается последовательностью

символов \*/. Обычно используется для выделения в качестве комментария

данных справочного характера.

Не следует недооценивать значения комментариев. Хороший комментарий существенно улучшает читабельность программного кода и позволяет избежать многих неприятностей. К сожалению, большинство программистов учатся этому правилу на своих горьких ошибках.

Обычно программы пишут для того, чтобы обрабатывать данные. Методы и возможности по обработке данных в значительной степени зависят от типа данных. Язык Java относится к строго типизованным языкам. Это означает, что любая переменная в программе относится к определенному типу данных — одному и только одному.



Указать тип переменной недостаточно. Переменной рано или поздно придется присвоить значение. Делается это с помощью литералов. Литерал — это постоянное значение, предназначенное для восприятия человеком, которое не может быть изменено в программе. В рассмотренном ранее примере уже использовался строчный литерал — фраза "Мы программируем на Java!". Читатель, вероятно, не удивится, узнав, что целочисленные литералы вводятся с помощью арабских цифр от 0 до 9. Также вводятся действительные числа. При этом в качестве десятичного разделителя используется точка. Символы вводятся в одинарных кавычках (не путать с текстом, который заключается в двойные кавычки!), а для ввода логических значений указывают ключевые слова true и false.

Что касается непосредственно объявления переменных в Java, то выполняется оно по следующим правилам. В первую очередь при объявлении переменной перед ее именем в обязательном порядке указывается идентификатор типа. Например, инструкцией int n объявляется переменная n целочисленного типа int. Впоследствии этой переменной может быть присвоено значение. В качестве

оператора присваивания в Java используется оператор =. Следующими командами объявляется целочисленная переменная, после чего ей присваивается значение 12:

int n;

n=12;

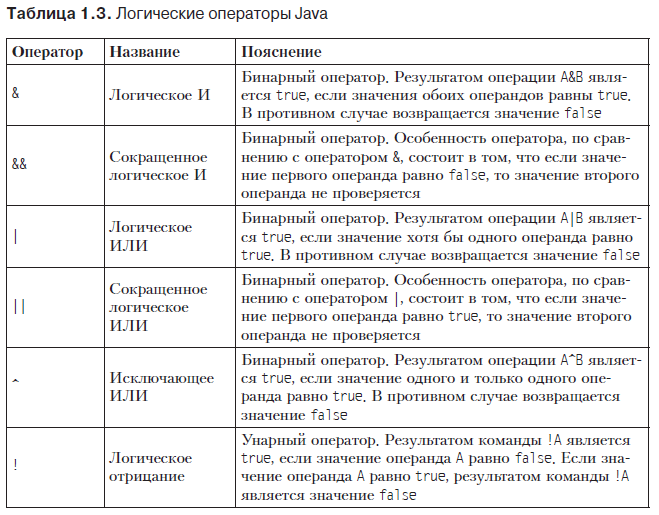
При этом всю означенную конструкцию из двух команд можно объединить в одну инструкцию вида int n=12

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15 | class VarDemo{  public static void main(String[] args){  // Инициализация переменных:  byte age=34;  char sex='м';  double weight=103.6;  int height=182;  // Вывод данных:  System.out.println("Персональные данные пользователя:");  System.out.println("Возраст: "+age+" лет");  System.out.println("Пол (м/ж): "+sex+".");  System.out.println("Вес: "+weight+" кг");  System.out.println("Рост: "+height+" см");  }  } |

Все операторы Java можно разделить на четыре группы: арифметические, логические, побитовые и сравнения. Рассмотрим последовательно каждую группу операторов. Начнем с арифметических. Эти операторы перечислены в табл. 1.2.

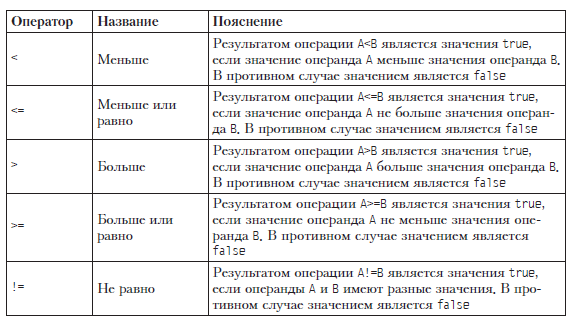
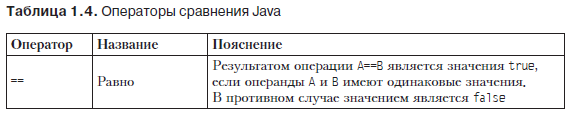


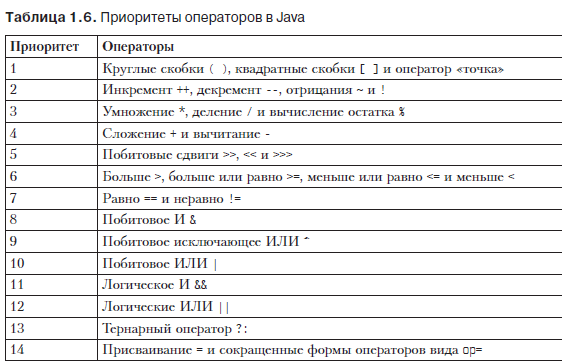
Следующую группу образуют логические операторы. Операндами логических операторов являются переменные и литералы типа boolean. Логические операторы Java перечислены в табл. 1.3.



Логические операторы обычно используются в качестве условий в условных операторах и операторах цикла.

В табл. 1.4 перечислены операторы сравнения, используемые в Java.





Стр 33-47 примеры просмотреть

Пример программы ввода

|  |  |
| --- | --- |
| package com.jolken;  import java.util.Scanner;  public class AddTwo {  public static void main(String[] args) {  Scanner sc = new Scanner(System.*in*); //Сканер нужен для считывания данных из потока, в данном случае консоли  int x = sc.nextInt(); //Считываем переменную типа int  System.*out*.println(x+2);  } } |  |

Вопросы 1-11

Задачи 1.1-1.12