

**Лекция №2.
по курсу «Основы Java»**

Москва 2020

Современные платформы для разработки программ

- **Java (*Sun Microsystems, 1995*)** – платформа для разработки программ на объектно-ориентированном языке *Java*, программы на котором компилируются в Java байт-код (до сих пор имеет статус ведомственного стандарта Sun)
- **.NET (*Microsoft, 2000*)** – **многоязыковая** объектно-ориентированная платформа для разработки программ с общим промежуточным языком (CIL), общей инфраструктурой языков (CLI) и единым представлением данных на основе XML (стандарты *ISO/ECMA*). Язык C# - наиболее удобный язык программирования для .NET, **но не единственный и не обязательный для использования**
- Обе платформы уделяют особое внимание надежности и безопасности, на основе **исполнения управляемого кода и динамического контроля топов**. И Java, и .NET – наиболее безопасные платформы

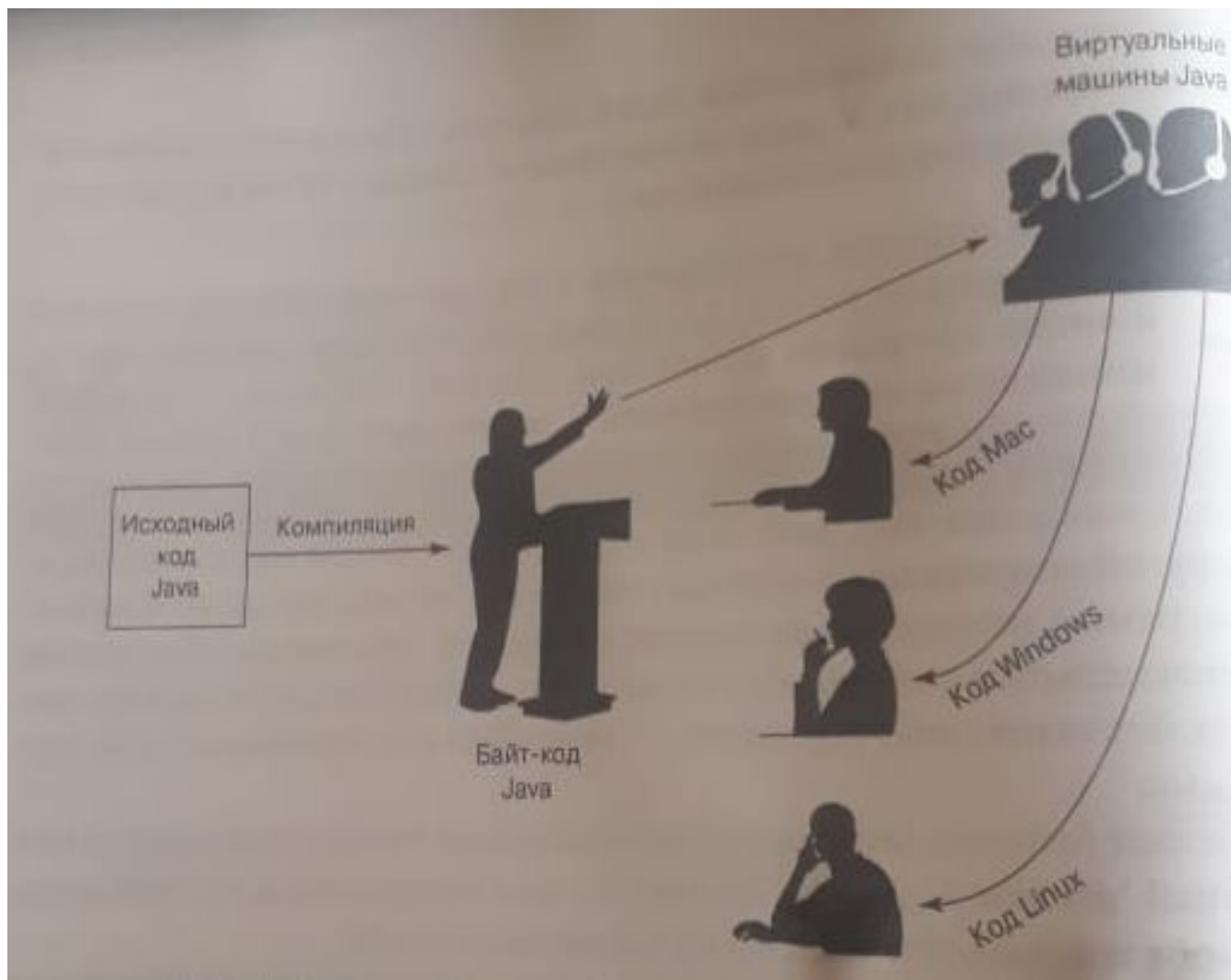
Виртуальная машина Java

Посредник между байт-кодом и конкретной платформой

Байт-код

Портируемая
(переносимая) программа

Java Development Kit (инструмент
комплект средств поддержки разработок)



Работа с Java

```
import java.awt.*;
import java.awt.event.*;
class Party {
    public void buildInvitel() {
        Frame f = new Frame();
        Label l = new Label("Вечеринка у Тима");
        Button b = new Button("Ваша ставка");
        Button c = new Button("Сбросить");
        Panel p = new Panel();
        p.add(l);
        // Еще код...
    }
}
```

Исходник

1

Наберите свой исходный код.

Сохраните его как **Party.java**.

```
File Edit Window Help Plead
javac Party.java
```

Компилятор

2

Скомпилируйте файл **Party.java**, запустив утилиту `javac` (приложение-компилятор). Если все пройдет без ошибок, вы получите еще один файл с именем **Party.class**.

Файл **Party.class**, сгенерированный компилятором, состоит из байт-кода.

```
Method Party()
0 aload_0
1 invokespecial #1 <Method java.lang.Object.>
4 return
Method void buildInvitel()
0 new #2 <Class java.awt.Frame>
3 dup
4 invokespecial #3 <Method java.awt.Frame.>
```

Вывод (код)

3

Скомпилированный код: файл **Party.class**.

```
File Edit Window Help Swear
java Party
Вечеринка у Тима
[Ваши ставки] [Сбросить]
```

Виртуальные машины

4

Выполните программу, запустив виртуальную машину Java (Java Virtual Machine, или JVM) с файлом **Party.class**. JVM транслирует байт-код в такой формат, который поймет целевая платформа, и запустит вашу программу.

Создание модуля

The screenshot displays the Android Studio interface for a project named 'ShowNameNow'. The main editor shows the 'activity_main.xml' file in Design mode, featuring a white and a dark blue theme. The Palette on the left lists various UI components like TextView, Button, and RecyclerView. The Component Tree below it shows the hierarchy of the layout, including a ConstraintLayout with three child views: a button labeled 'Show Name', a TextView, and an EditText labeled 'Enter Name'. The bottom status bar indicates a successful build: 'Build: completed successfully at 31.08.2019 14:46' with a total duration of 2 s 81 ms. The Event Log on the right shows the execution of tasks and the completion of the Gradle build in 2 s 90 ms. A Windows activation watermark is visible in the bottom right corner.

Build: Build Output x Sync x

- Build: completed successfully at 31.08.2019 14:46 (2 s 81 ms)
- Run build E:\AndroidProjects\ShowNameNow (1 s 841 ms)
 - Load build (6 ms)
 - Evaluate settings (4 ms)
 - Finalize build cache configuration
 - Configure build (222 ms)
 - Calculate task graph (97 ms)

Event Log

- You can disable this experimental feature from File → Settings → Experimental → Gradle → Only sync the active variant
- 14:46 Executing tasks: [:app:assembleDebug] Активация Windows
- 14:46 Gradle build finished in 2 s 90 ms Чтобы активировать Windows, перейдите в раздел "Параметры".

Gradle build finished in 2 s 90 ms (3 minutes ago)

Java code

The screenshot displays the Android Studio interface. The main editor shows the following Java code in `MyClass.java`:

```
1 package com.example.learnjava;
2
3 public class MyClass {
4     public static void main(String[] args) {
5         |
6     }
7 }
8
```

The left sidebar shows the project structure with `MyClass` selected under `com.example.learnjava`. The bottom panel is split into two tabs: `Build Output` and `Event Log`.

Build Output:

- Build: completed successfully at 31.08.2019 14:46 (2 s 81 ms)
- Run build E:\AndroidProjects\ShowNameNow (1 s 841 ms)
 - Load build (6 ms)
 - Evaluate settings (4 ms)
 - Finalize build cache configuration
 - Configure build (222 ms)
 - Calculate task graph (97 ms)

Event Log:

- 14:46 Executing tasks: [:app:assembleDebug]
- 14:46 Gradle build finished in 2 s 90 ms

A Windows activation watermark is visible in the bottom right corner of the Event Log area.

At the bottom of the screen, the status bar shows: `Gradle build finished in 2 s 90 ms (31 minutes ago)` and `5:9 CRLF UTF-8 4 spaces`.

Java code

The screenshot displays the Android Studio interface. The top toolbar shows the Run button (a green play icon) with a tooltip that reads "(989) Входящие - Почта Mail.ru". The breadcrumb navigation indicates the current file path: ShowNameNow > LearnJava > src > main > java > com > example > learnjava > MyClass. The Package Explorer on the left shows the project structure, with the 'MyClass' file selected under the 'com.example.learnjava' package. The main editor window shows the following Java code:

```
1 package com.example.learnjava;
2
3 public class MyClass {
4     public static void main(String[] args) {
5         System.out.println("Hello world");
6     }
7 }
8
```

Below the code editor, the Run tab is active, showing the execution output:

```
"C:\Program Files\Android\Android Studio\jre\bin\java.exe" ...
Hello world
Process finished with exit code 0
```

The Event Log on the right side of the Run tab shows the following build and execution events:

- 15:22 Gradle build finished in 1 m 8 s 755 ms
- 15:23 Executing tasks: [:app:assembleDebug]
- 15:23 Gradle build finished in 22 s 513 ms
- 15:26 Executing tasks: [:LearnJava:compileJava, :LearnJava:testClasses]
- 15:26 Gradle build finished in 654 ms

At the bottom of the screen, a Windows notification is visible: "Активация Windows. Чтобы активировать Windows, перейдите в раздел 'Параметры'". The status bar at the very bottom shows "No changes to deploy. // (Don't show again) (11 minutes ago)" and "3:7 CRLF UTF-8".

IntelliJ IDEA



Version: 2018.2.6
Build: 182.5107.16
Released: November 13, 2018
[Release notes](#)

[System requirements](#)
[Installation instructions](#)
[Previous versions](#)

Download IntelliJ IDEA

[Windows](#) [macOS](#) [Linux](#)

Ultimate

For web and enterprise development

[DOWNLOAD](#) [.EXE](#)

Free trial

Community

For JVM and Android development

[DOWNLOAD](#) [.EXE](#)

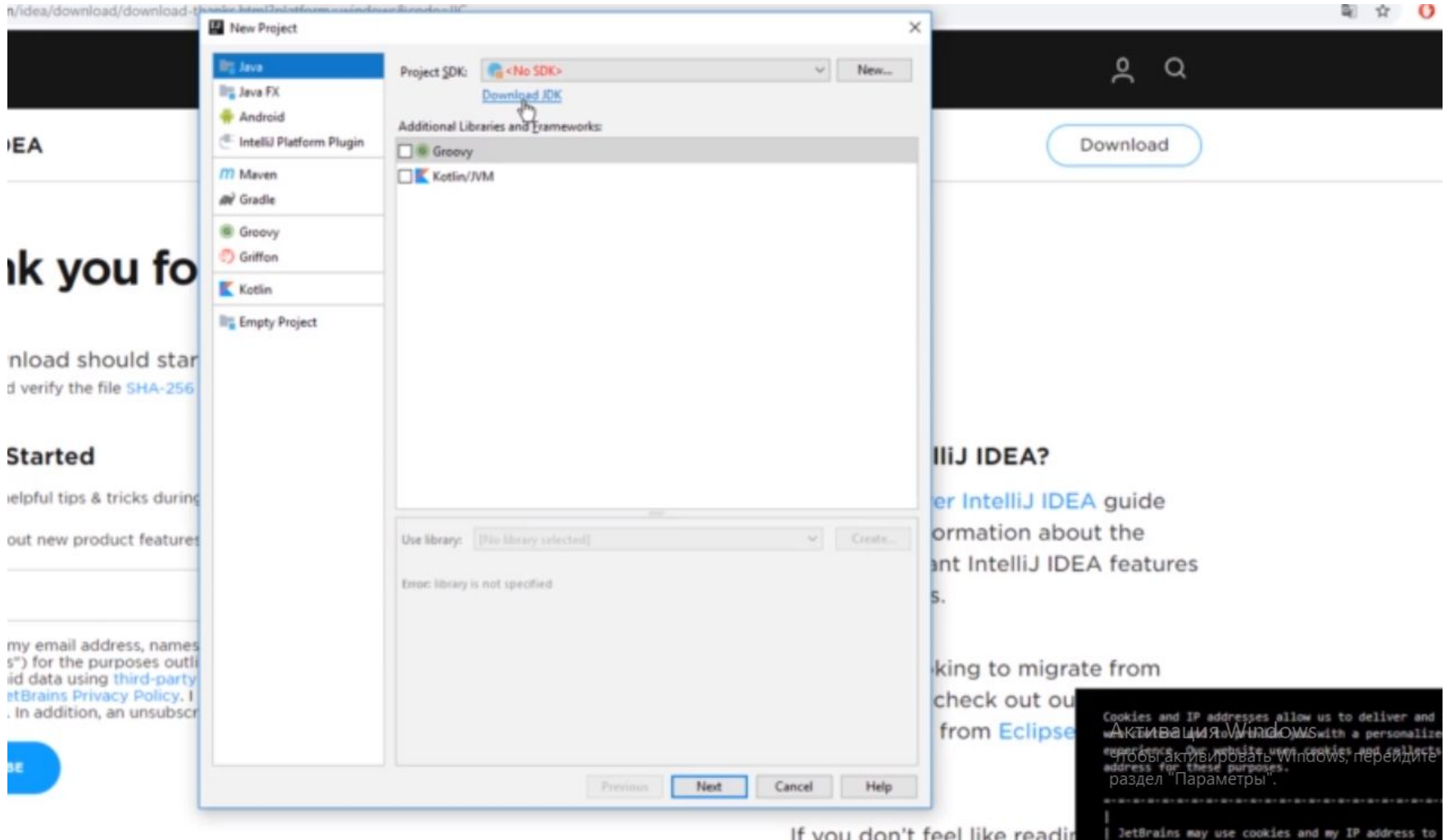
Free, open-source

	Commercial	Open-source, Apache 2.0
License	Commercial	Open-source, Apache 2.0
Java, Kotlin, Groovy, Scala	✓	✓
Android	✓	✓
Maven, Gradle, SBT	✓	✓

```
-----  
| JetBrains may use cookies and my IP address to  
| collect individual statistics and to provide  
| personalized offers and ads subject to the Pr  
| Policy and the Terms of Use. JetBrains may us  
| necessary services for this purpose. I can  
| opt out of this at any time by visiting the Opt-Out
```

Активация Windows
Чтобы активировать Windows, перейдите в
раздел "Параметры".

Создание проекта IntelliJ IDEA




If you don't feel like reading

Установка Java development kit

Oracle Technology Network / Java / Java SE / Downloads

Overview Downloads Documentation Community Technologies Training

Java SE Downloads



Java Platform (JDK) 11

Java Platform, Standard Edition

Java SE 11.0.1(LTS)
Java SE 11.0.1 is the latest release for the Java SE 11 Platforms
[Learn more](#)

- [Installation Instructions](#)
- [Release Notes](#)
- [Oracle JDK License](#)
- [Java SE Licensing Information User Manual](#)
 - Includes Third Party Licenses
- [Certified System Configurations](#)
- [Readme](#)

Oracle JDK
[DOWNLOAD](#)

Looking for Oracle OpenJDK builds?

- **Oracle Customers and ISVs targeting Oracle LTS releases:** Oracle JDK is Oracle's supported Java SE version for customers and for developing, testing, prototyping or demonstrating your Java applications.
- **End users and developers looking for free JDK versions:** Oracle OpenJDK offers the same features and performance as Oracle JDK under the GPL license.

Java SDKs and Tools

- [Java SE](#)
- [Java EE and Glassfish](#)
- [Java ME](#)
- [Java Card](#)
- [NetBeans IDE](#)
- [Java Mission Control](#)

Java Resources

- [Java APIs](#)
- [Technical Articles](#)
- [Demos and Videos](#)
- [Forums](#)
- [Java Magazine](#)
- [Developer Training](#)
- [Tutorials](#)
- [Java.com](#)

Java SE
Java EE
Java ME
Java SE Subscription
Java Embedded
Java Card
Java TV
Community
Java Magazine

Установка jdk

- [Java Developer Day hands-on workshops \(free\) and other events](#)
- [Java Magazine](#)

JDK 11.0.1 checksum

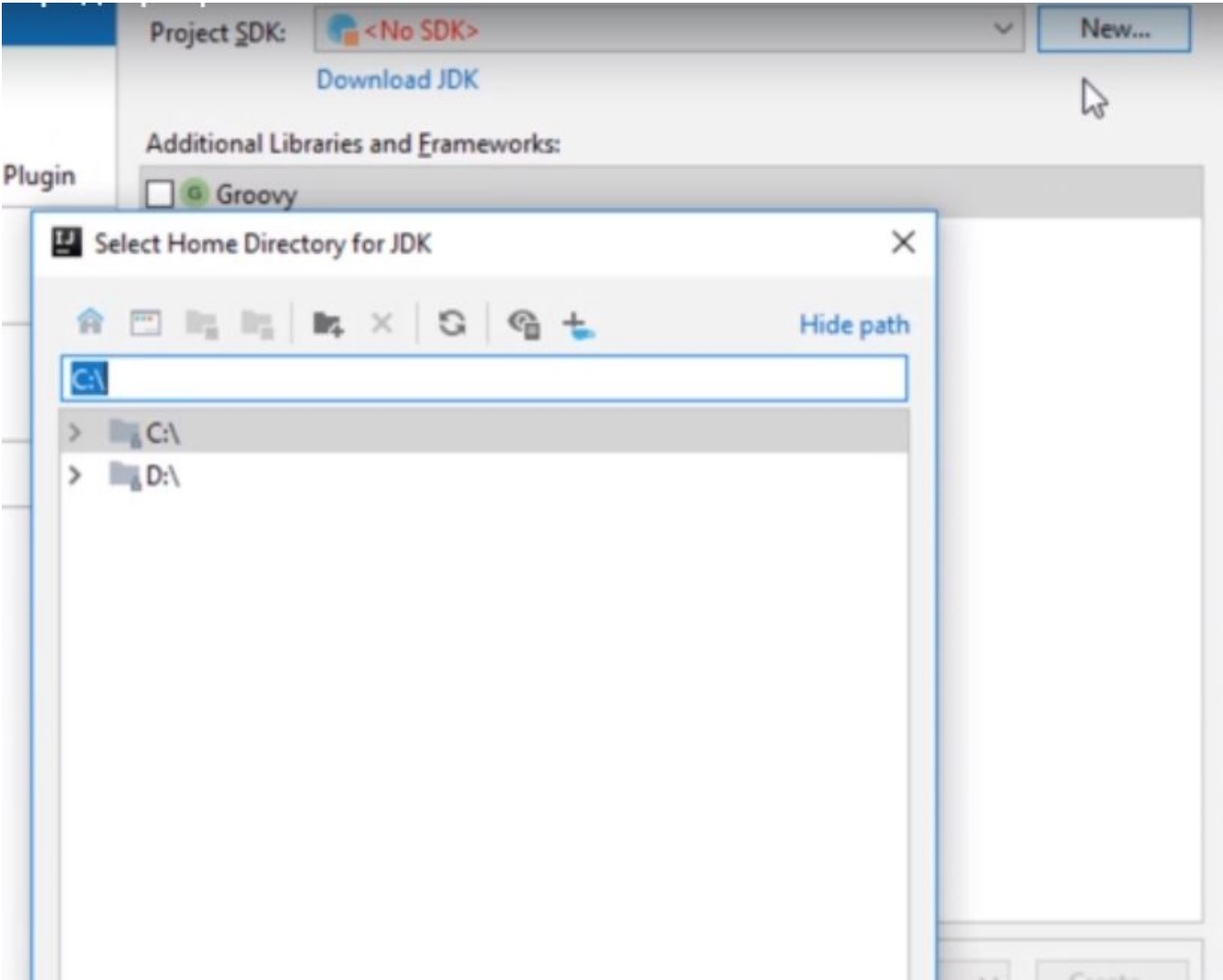
Java SE Development Kit 11.0.1

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

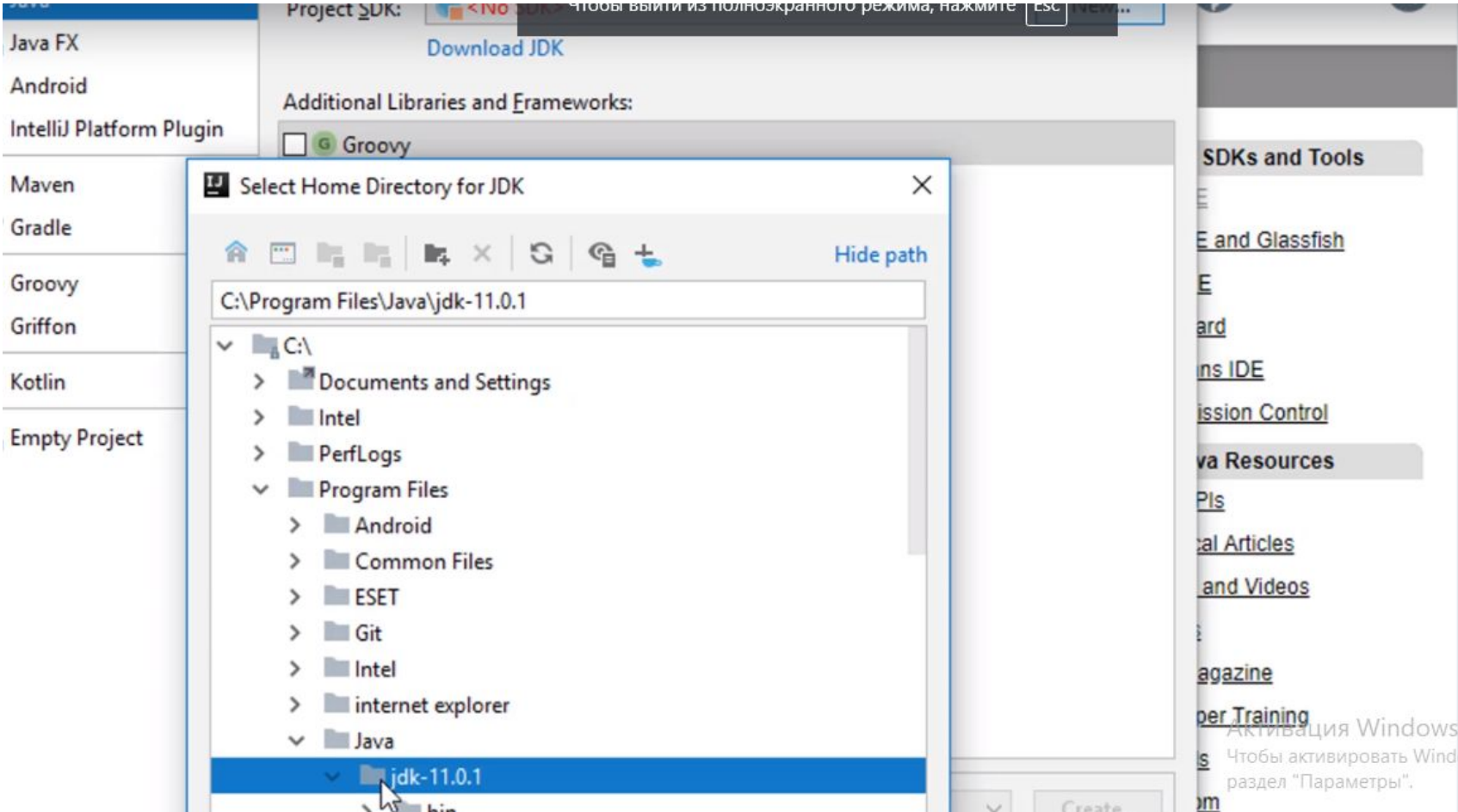
Thank you for accepting the Oracle Technology Network License Agreement for Oracle Java SE; you may now download this software.

Product / File Description	File Size	Download
Linux	147.4 MB	jdk-11.0.1_linux-x64_bin.deb
Linux	154.09 MB	jdk-11.0.1_linux-x64_bin.rpm
Linux	171.43 MB	jdk-11.0.1_linux-x64_bin.tar.gz
macOS	166.2 MB	jdk-11.0.1_osx-x64_bin.dmg
macOS	166.55 MB	jdk-11.0.1_osx-x64_bin.tar.gz
Solaris SPARC	186.8 MB	jdk-11.0.1_solaris-sparcv9_bin.tar.gz
Windows	150.98 MB	jdk-11.0.1_windows-x64_bin.exe
Windows	170.99 MB	jdk-11.0.1_windows-x64_bin.zip

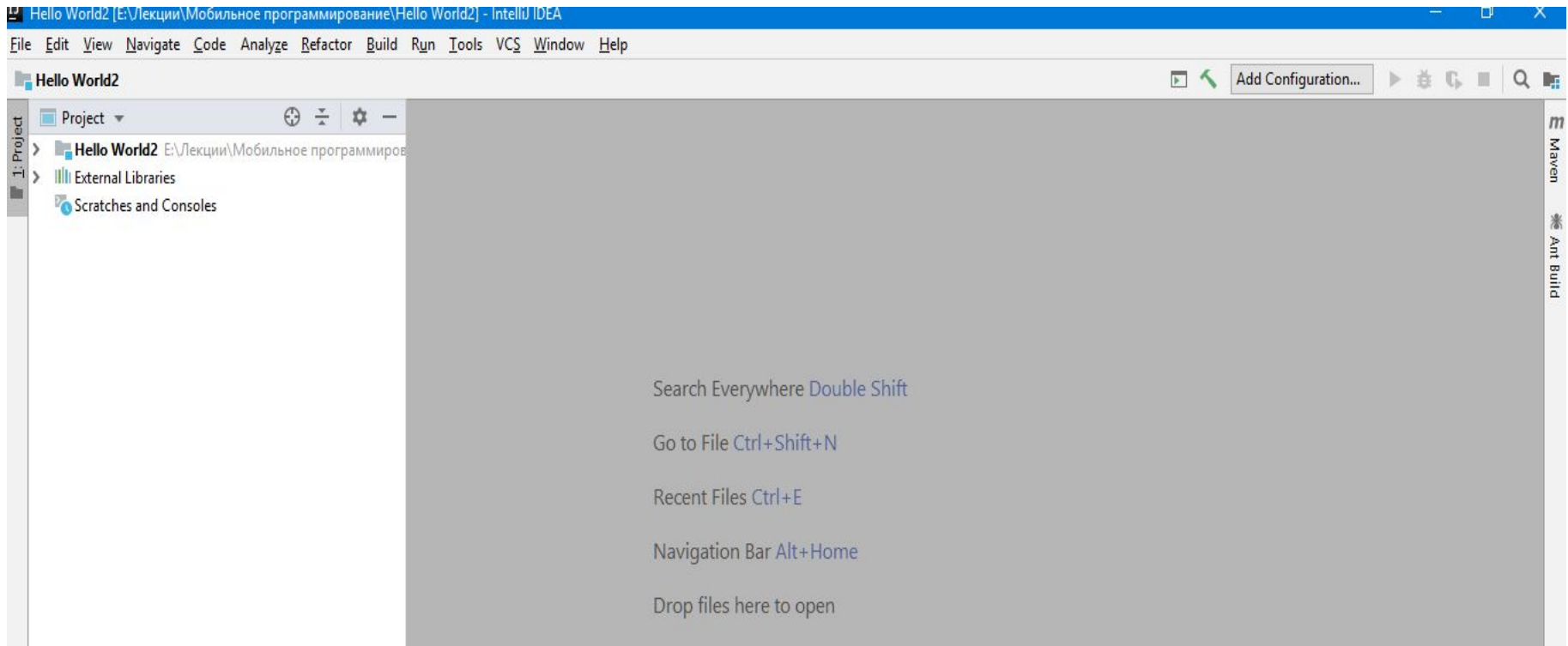
IntelliJ IDEA



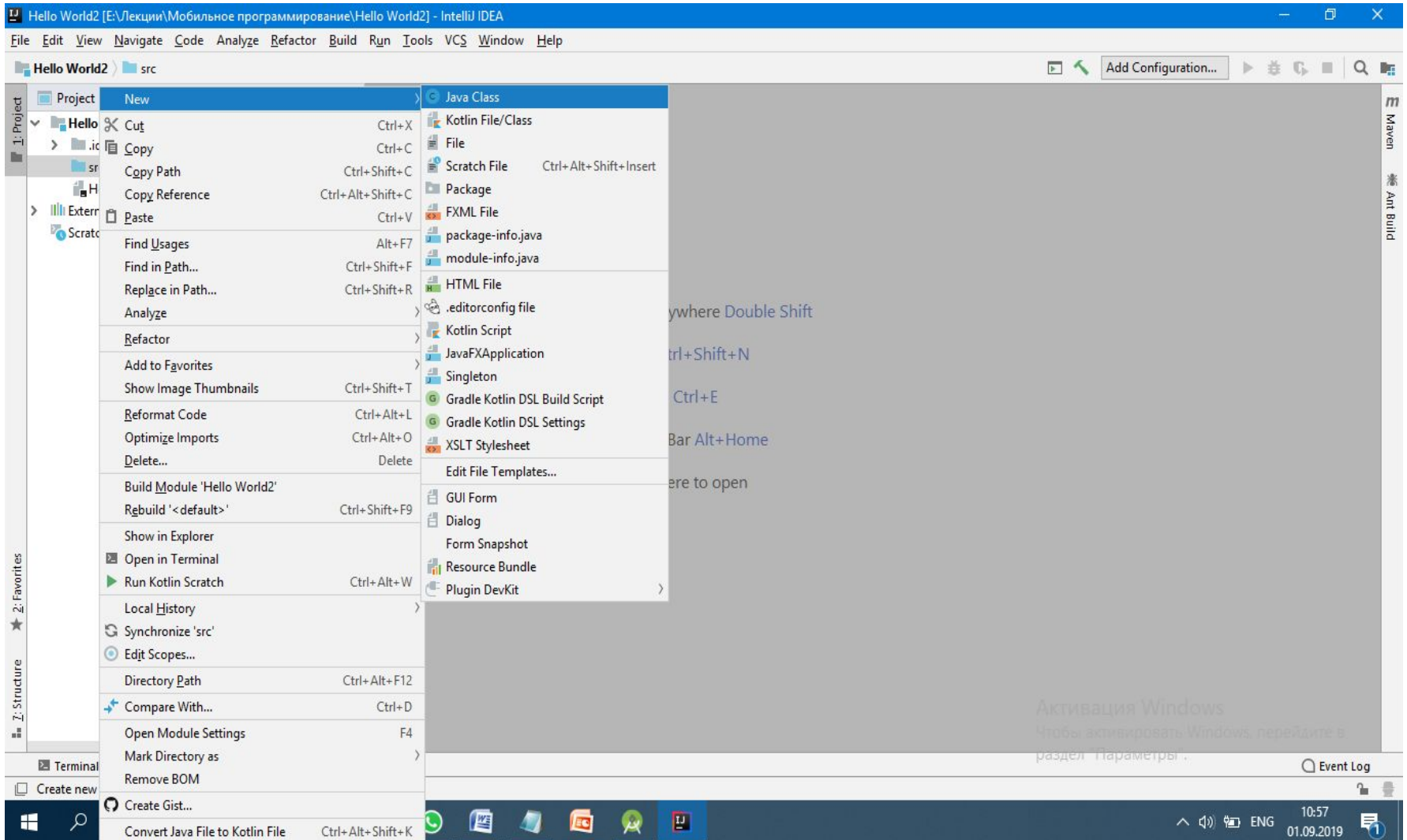
IntelliJ IDEA



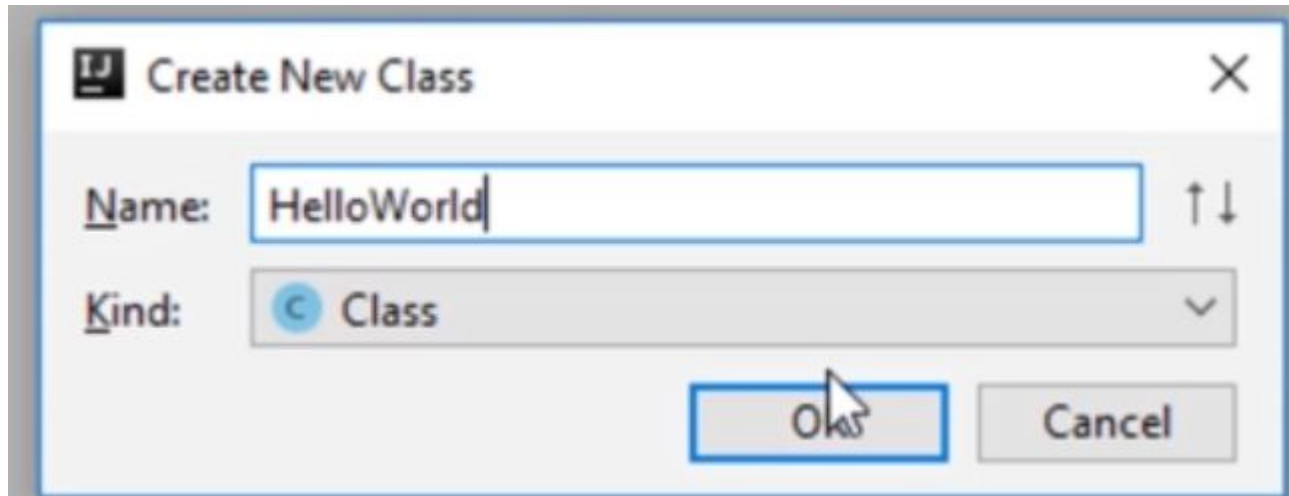
Первый проект в IntelliJ IDEA



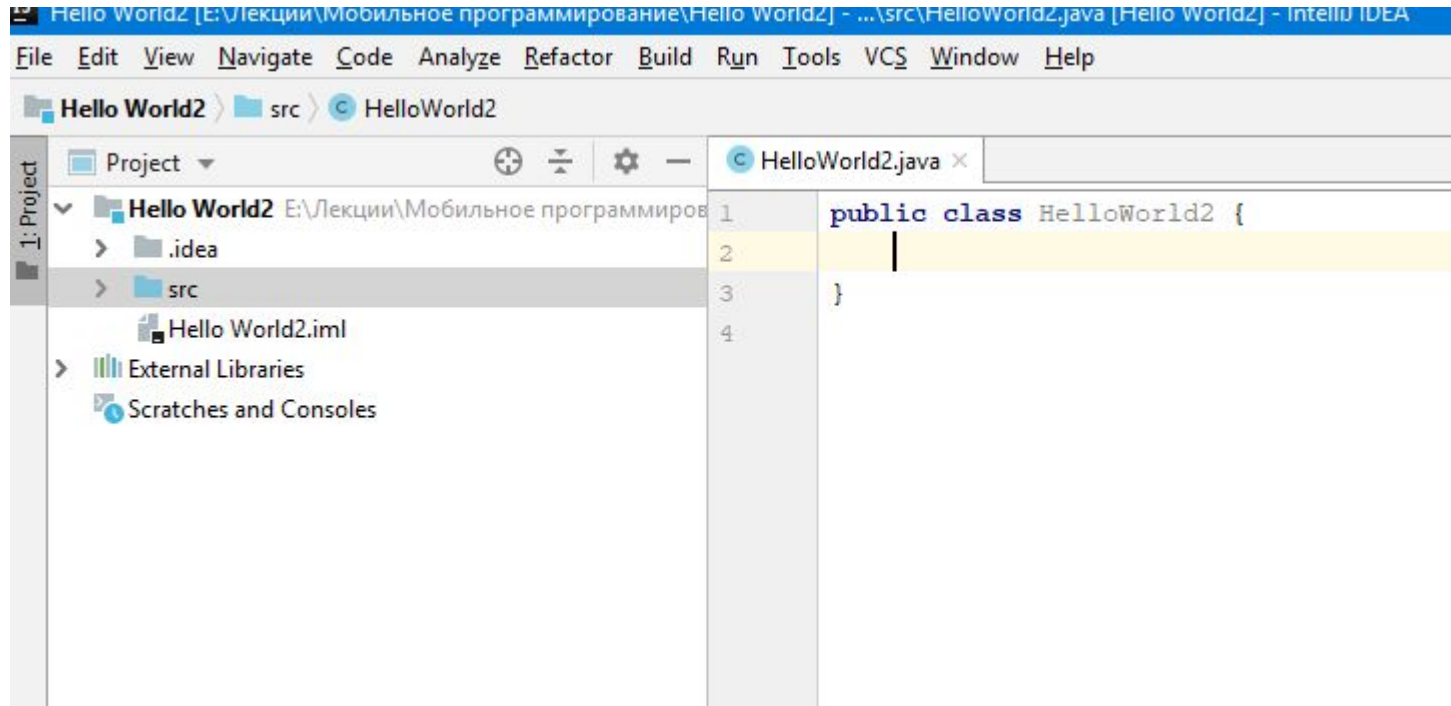
Первый проект в IntelliJ IDEA



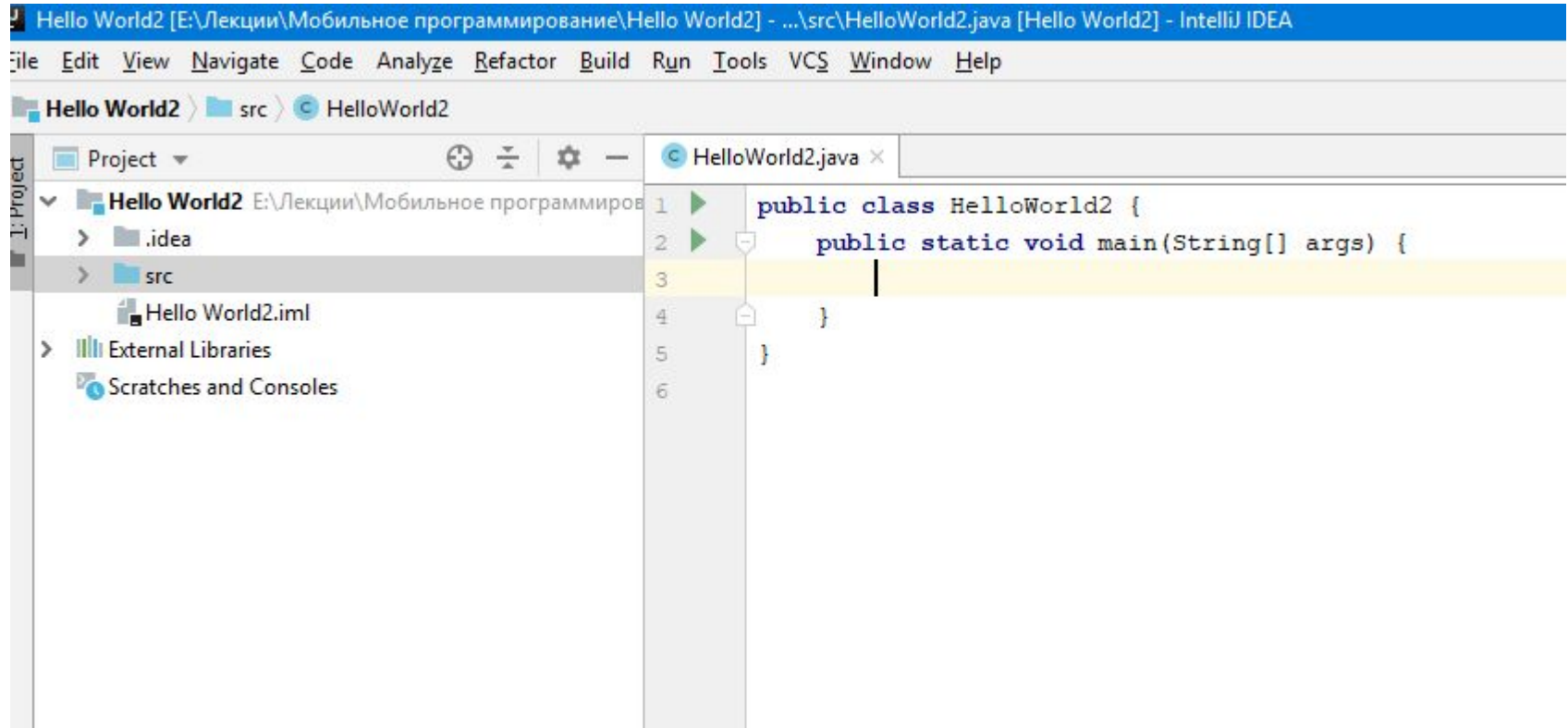
Разделы курса



Первый проект в IntelliJ IDEA



Первый проект в IntelliJ IDEA



Первый проект в IntelliJ IDEA

Весь код на Java пишется внутри классов { между фигурными скобками; }

Точка входа в программу:
`public static void main(String... args){}`

Имена классов с заглавной буквы (HelloWorld)

Имена методов с прописной буквы (main())

Вывести строку в консоль `System.out.println("Строка");`

Активация Windows
Чтобы активировать Windows, перейдите в раздел Параметры .

Условный оператор

The screenshot displays the IntelliJ IDEA IDE interface. The main editor window shows the following Java code:

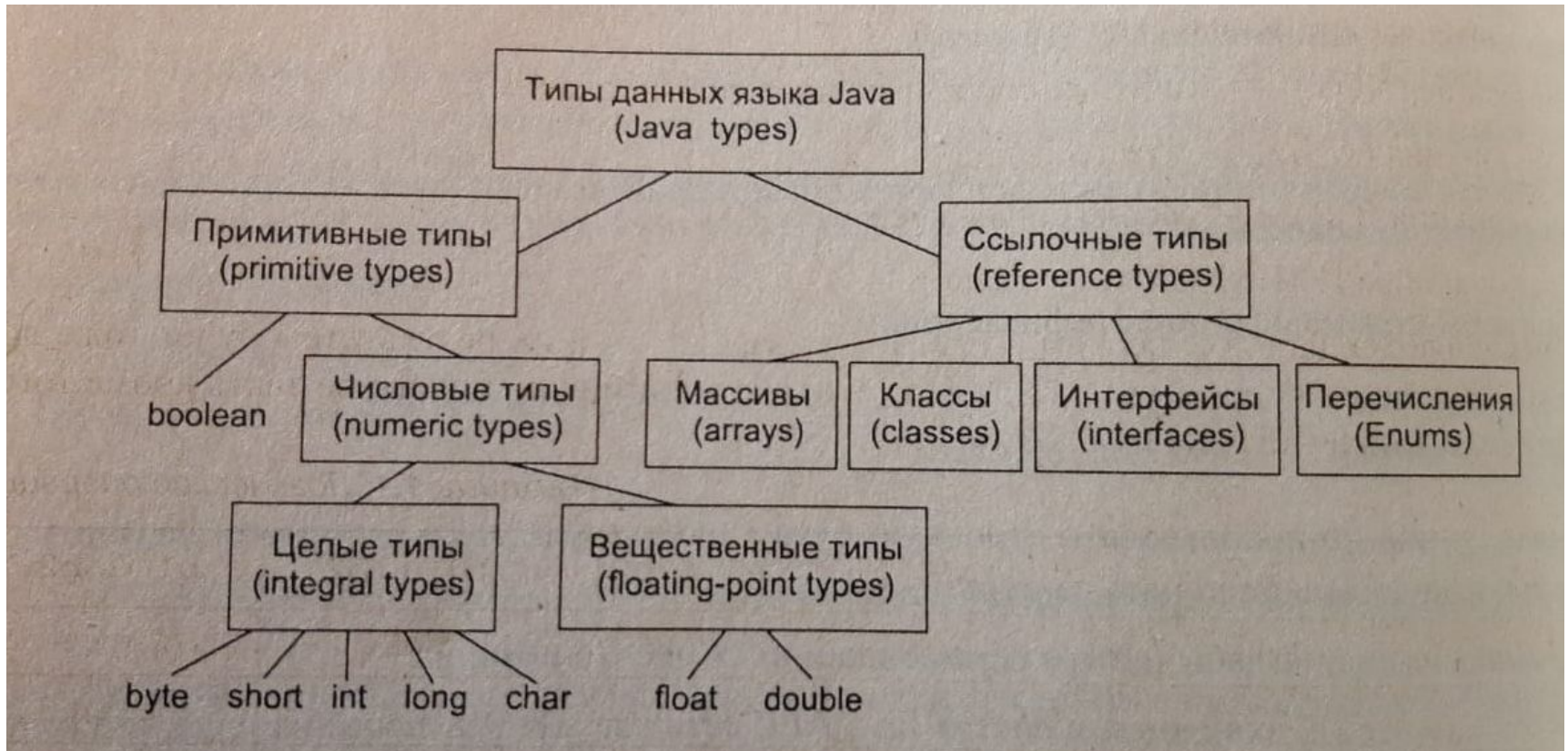
```
1 public class HelloWorld2 {  
2     public static void main(String[] args) {  
3         System.out.println("Hello world2");  
4         float pi = 3.14f;  
5         double pid = 3.14;  
6         if (pi>5) {  
7             System.out.println("Pi>5");  
8         }  
9         else  
10        {  
11            System.out.println("ddd2");  
12        }  
13    }  
14 }  
15
```

The output window at the bottom shows the execution results:

```
Run: HelloWorld2 x  
"C:\Program Files\Java\jdk-9.0.4\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2018.3.5\lib\idea_rt.jar=14246:C:\Prog  
Hello world2  
ddd2  
Process finished with exit code 0
```

At the bottom of the IDE, there is a Windows activation watermark: "Активация Windows. Чтобы активировать Windows, перейдите в раздел 'Параметры'."

Типы данных



Типы целых чисел

4 категории примитивных (простых) типов

Целые
числа

byte

short

int

long

Дробные

float

double

Символы

char

Логические

boolean

Активация Windows

Чтобы активировать Windows, перейдите в раздел "Параметры".

Типы целых чисел

4 категории примитивных (простых) типов

Целые
числа

byte	1 байт	от -128 до 127
short	2 байта	от -32768 до 32767
int	4 байта	от -2147483648 до 2147483647
long	8 байт	от -9223372036854775808 до 9223372036854775807

byte number; short number = 3500; int number = 950000; long number = 999999999999;
number = 120.

Задача. Дано количество дней 1000, скорость света равна 300 000 км/сек, какое расстояние пройдет свет за это время?

Активация Windows
Чтобы активировать Windows, перейдите в раздел "Параметры".

Условный оператор

If (условие)

{

тело условия

}

Elseif ...

Else ..

&& - логическое И

|| - логическое или

Циклы

Циклы с фиксированным количеством итераций

```
for (int i =0; i<10; i++)
```

```
{
```

```
}
```

While (лог выр) оператор

do оператор while (лог выражение)

```
for (int i:localArr)
```

```
{
```

Тело цикла

```
}
```

Циклы

Вывод числа от 1 до 1000.

```
public class Loop {  
    public static void main(String[] args) {  
        int i = 1;  
        while (i <= 1000) {  
            System.out.println(i);  
            i++;  
        }  
    }  
}
```

Цикл с пост условием.

```
public class Loop {  
    public static void main(String[] args) {  
        int a = 0;  
        do {  
            System.out.println("Привет");  
        } while (a > 0);  
    }  
}
```

break – прерывание
цикла

Цикл с фиксированным количеством итераций

```
public class Loop {  
    public static void main(String[] args) {  
        for(int i = 0; i < 100; i++) {  
            System.out.println(i);  
        }  
    }  
}
```

Задача на циклы

Вывести все четырехзначные числа типа ABCD, где: $A + B = C + D$

```
for (int a = 1; a <= 9; a++)
{
    for (int b = 0; b <= 9; b++)
    {
        for (int c = 0; c <= 9; c++)
        {
            for (int d = 0; d <= 9; d++)
            {
                if ((a + b) == (c + d))
                {
                    System.out.println(
                        " " + a + " " + b + " " + c + " " + d);
                }
            }
        }
    }
}
```

Задача на циклы

Вывести все четырехзначные числа типа ABCD, где: $A + B = C + D$

```
import java.util.Random;
import java.util.Scanner;

public class LektionTwo {
    public static void main(String[] args) {

        Scanner keyboard = new Scanner(System.in);
        int randomNumber = new Random().nextInt(10) + 1;
        while (true) {
            System.out.println("Введите число");
            int number = keyboard.nextInt();
            if (number < randomNumber) System.out.println("<");
            if (number > randomNumber) System.out.println(">");
            if (number == randomNumber) {
                System.out.println("Pobeda!");
                break;
            }
        }
    }
}
```

Циклы

a++ инкремент => a = a + 1

a-- декремент => a = a - 1

a +=5 => a = a + 5

a -=5 => a = a - 5

Цикл while (условие)

```
public class Loop {  
    public static void main(String[] args) {  
        int i = 1;  
        while (i <= 1000) {  
            System.out.println(i);  
            i++;  
        }  
    }  
}
```

МАССИВЫ

№	Объявление массива, Java-синтаксис	Примеры	Комментарий
1.	<code>dataType[] arrayName;</code>	<code>int[] myArray;</code> <code>Object[] arrayOfObjects;</code>	Желательно объявлять массив именно таким способом, это Java-стиль
2.	<code>dataType arrayName[];</code>	<code>int myArray[];</code> <code>Object arrayOfObjects[];</code>	Унаследованный от C/C++ способ объявления массивов, который работает и в Java

Как и любой другой объект, создать массив Java, то есть зарезервировать под него место в памяти, можно с помощью оператора `new`. Делается это так:

```
1 new typeOfArray [length];
```

Где `typeOfArray` — это тип массива, а `length` — его длина (то есть, количество ячеек), выраженная в целых числах (`int`). Однако здесь мы только выделили память под массив, но не связали созданный массив ни с какой объявленной ранее переменной. Обычно массив сначала объявляют, а потом создают, например:

```
1 int[] myArray; // объявление массива
2 myArray = new int[10]; // создание, то есть, выделение памяти для массива на 10 элементов типа int
```

МАССИВЫ

```
int[] myArray = new int[10];
```

получаем массив из десяти целых чисел, и, пока это не изменится в ходе программы, в каждой ячейке записан 0. массив с данными ссылочного типа, то по умолчанию в каждой ячейке записаны null

```
1 String[] seasons = new String[4]; /* объявили и создали массив. Java выделила память под массив из 4
2
3 seasons[0] = "Winter"; /* в первую ячейку, то есть, в ячейку с нулевым номером мы записали строку Wint
4 seasons[1] = "Spring"; // проделываем ту же процедуру с ячейкой номер 1 (второй)
5 seasons[2] = "Summer"; // ...номер 2
6 seasons[3] = "Autumn"; // и с последней, номер 3
```

Теперь во всех четырёх ячейках нашего массива записаны названия сезонов. Инициализацию также можно провести по-другому, совместив с инициализацией и объявлением:

```
1 String[] seasons = new String[] {"Winter", "Spring", "Summer", "Autumn"};
```

Более того, оператор `new` можно опустить:

```
1 String[] seasons = {"Winter", "Spring", "Summer", "Autumn"};
```


Массивы

Тип данных [] arr = new Тип данных [размерность]

```
int[] a = new int[50]
```

или

```
int[] arr = { 1,2,3 }
```

Двумерные:

```
int [][] d = new int [3][4]
```

```
int [][] arr2 = { {1,2,3}, {4,5,6} }
```

Элементами массива – являются массивы

МНОГОМЕРНЫЕ МАССИВЫ

Многомерный массив объявляется и создается следующим образом:

```
1 Int[][] myTwoDimentionalArray = new int [8][8];
```

Для работы с массивами в Java есть класс `java.util.Arrays` (`arrays` на английском и означает “массивы”). В целом с массивами чаще всего проделывают следующие операции: заполнение элементами (инициализация), извлечение элемента (по номеру), сортировка и поиск.

==

!=

`if (!str1.equals("x"))`

{

}

Циклы

```
Scanner keyboard = new Scanner(System.in);
int randomNumber = new Random().nextInt(10) + 1;
while (true) {
    System.out.println("Введите число");
    int number = keyboard.nextInt();
    if (number < randomNumber) System.out.println("<");
    if (number > randomNumber) System.out.println(">");
    if (number == randomNumber) {
        System.out.println("Pobeda!");
        break;
    }
}
```

Классы

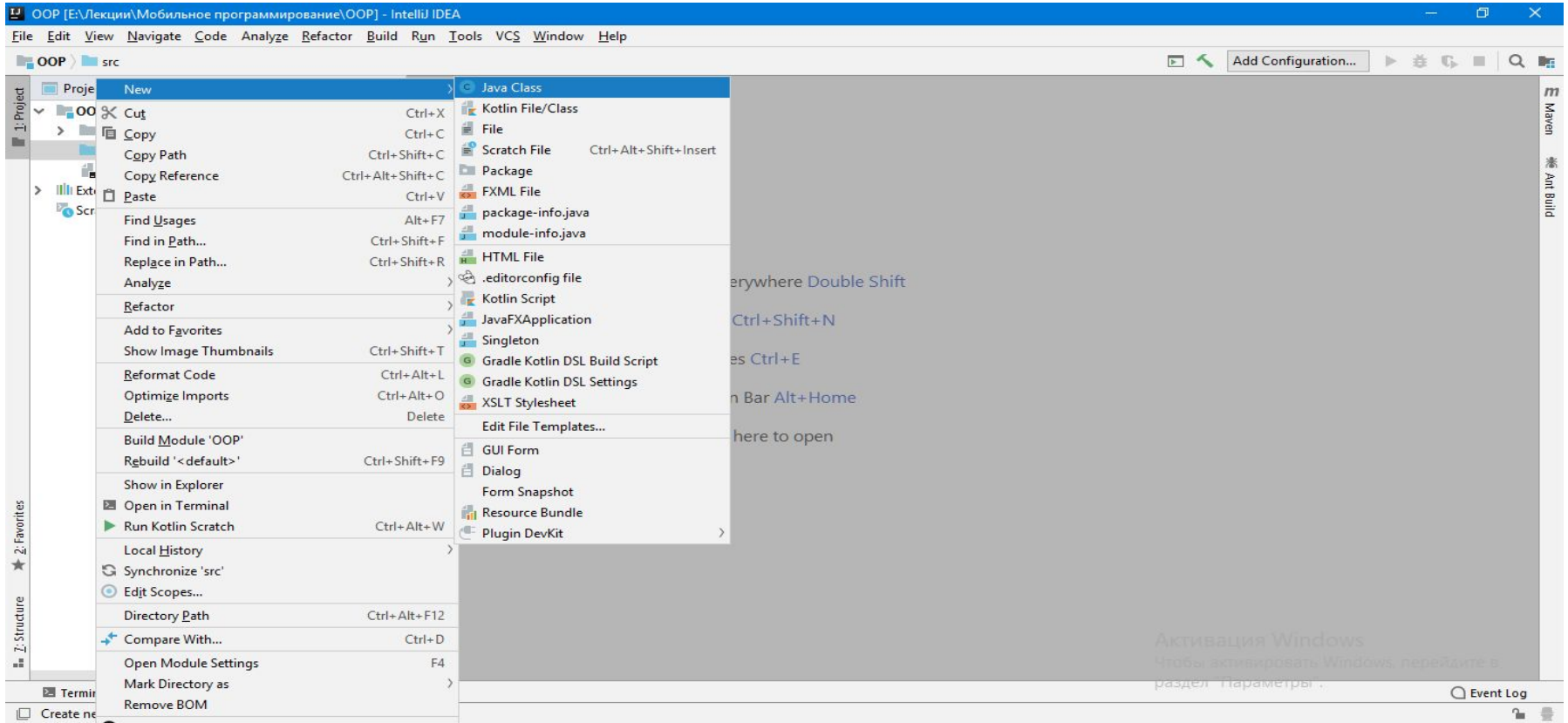
- *Классы* позволяют проводить конструирование из полезных компонентов, обладающих простыми инструментами, что позволяет абстрагироваться от деталей реализации.
- Данные и операции над ними образуют определенную сущность, и они не разносятся по всей программе, как нередко бывает в случае процедурного программирования, а описываются вместе. Локализация кода и данных улучшает наглядность и удобство сопровождения программного обеспечения.
- *Инкапсуляция* позволяет привести свойство *модульности*, что облегчает распараллеливание выполнения задачи между несколькими исполнителями и обновление версий отдельных компонентов.

Полиморфизм оказывается полезным преимущественно в следующих ситуациях.

- Обработка разнородных структур данных. Программы могут работать, не различая вида *объектов*, что существенно упрощает код. Новые виды могут быть добавлены в любой момент.
- Изменение *поведения* во время исполнения. На этапе исполнения один *объект* может быть заменен другим, что позволяет легко, без изменения кода, адаптировать алгоритм в зависимости от того, какой используется *объект*.
- Реализация работы с наследниками. Алгоритмы можно обобщить настолько, что они уже смогут работать более чем с одним видом *объектов*.
- Создание "каркаса" (framework). Независимые от приложения части предметной области могут быть реализованы в виде набора универсальных *классов*, или каркаса (framework), и в дальнейшем расширены за счет добавления частей, специфичных для конкретного приложения.

Активация Windows

Классы



Классы

```
class ИмяКласса {  
    int переменная1;  
    String переменная2;  
    ...  
    void метод1();  
    int метод2();  
    ...  
}
```

```
Main.java x  Box.java x  
1 public class Box {  
2     double width;  
3     double height;  
4     double length;  
5 }  
  
public class Main {  
    public static void main(String[] args) {  
        Box myBox = new Box();  
        myBox.height = 55;  
        Box box4 = myBox;  
    }  
}
```

Конструктор

void - не возвращает никаких данных

```
тип имяМетода(список_параметров) {  
(int, boolean, Box...) тело_метода  
    ...  
}
```

```
public class Box {  
    double width;  
    double height;  
    double length;  
  
    Box() {  
        width = 10;  
        height = 10;  
        length = 10;  
    }  
}
```

Модификаторы доступа

По умолчанию все модификаторы доступа public.

'Generate...' and then 'Getter and Setter'.

Default – это доступ по умолчанию , доступность внутри пакета

```
public class Box {  
    private double width;  
    double height;  
    double length;  
  
    public double getWidth() {  
        return width;  
    }  
  
    public void setWidth(double width) {  
        this.width = width;  
    }  
}
```

public

private

protected

default

Пакеты

Элементами *пакета* являются содержащиеся в нем классы и интерфейсы, а также вложенные *пакеты*. Чтобы получить *составное имя пакета*, необходимо к полному имени *пакета*, в котором он располагается, добавить точку, а затем его собственное *простое имя*. Например, *составное имя* основного *пакета* языка Java – `java.lang` (то есть *простое имя* этого пакета `lang`, и он находится в объемлющем пакете `java`). Внутри него есть вложенный пакет, предназначенный для *типов* технологии reflection, которая упоминалась в предыдущих главах. Простое название пакета `reflect`, а значит, составное – `java.lang.reflect`.

Простое имя классов и интерфейсов дается при объявлении, например, `Object`, `String`, `Point`. Чтобы получить *составное имя* таких *типов*, надо к *составному имени пакета*, в котором находится *тип*, через точку добавить *простое имя типа*. Например, `java.lang.Object`, `java.lang.reflect.Method` или `com.myfirm.MainClass`. Смысл последнего выражения таков: сначала идет обращение к *пакету* `com`, затем к его *элементу* – вложенному *пакету* `myfirm`, а затем к *элементу пакета* `myfirm` – классу `MainClass`. Здесь `com.myfirm` – *составное имя пакета*, где лежит класс `MainClass`, а `MainClass` — *простое имя*. Составляем их и разделяем точкой – получается полное имя класса `com.myfirm.MainClass`.

Пакеты

Программа на Java представляет собой набор *пакетов* (packages). Каждый *пакет* может включать вложенные *пакеты*, то есть они образуют иерархическую систему.

Кроме того, *пакеты* могут содержать классы и интерфейсы и таким образом группируют *типы*. Это необходимо сразу для нескольких целей. Во-первых, чисто физически невозможно работать с большим количеством классов, если они "свалены в кучу". Во-вторых, модульная *декомпозиция* облегчает проектирование системы. К тому же, как будет показано ниже, существует специальный уровень доступа, позволяющий *типам* из одного *пакета* более тесно взаимодействовать друг с другом, чем с классами из других *пакетов*. Таким образом, с помощью *пакетов* производится логическая группировка *типов*. Из ООП известно, что большая *связность* системы, то есть среднее количество классов, с которыми взаимодействует каждый *класс*, заметно усложняет развитие и поддержку такой системы. Используя *пакеты*, гораздо проще организовать эффективное взаимодействие подсистем друг с другом.

Наконец, каждый *пакет* имеет свое *пространство имен*, что позволяет создавать одноименные классы в различных *пакетах*. Таким образом, разработчикам не приходится тратить время на разрешение конфликта имен.

Пакеты

Исходный код располагается в файлах с расширением .java, а бинарный – с расширением .class

исходный код классов Используйте == для сравнения два примитива, или
space.sunsystem.Moon посмотреть,
space.sunsystem.Sun если два ссылки относятся к тот же объект.
space.sunsystem.Test Используйте equals чтобы увидеть если два объекта
хранится в файлах равны.

space\sunsystem\Moon.java
space\sunsystem\Sun.java
space\sunsystem\Test.java

Integer.parseInt("3")
for (int cell : locationCells)

Не принято, чтобы классы находились не внутри пакетов

Пакеты

Package – указывает в каком пакете находится данный класс

```
package Main;

import box.Box;

public class Main {
    = public static void main(String[] args) {
        Box myBox = new Box();

        Box box4 = myBox;
    }
}
```

Отношения между классами

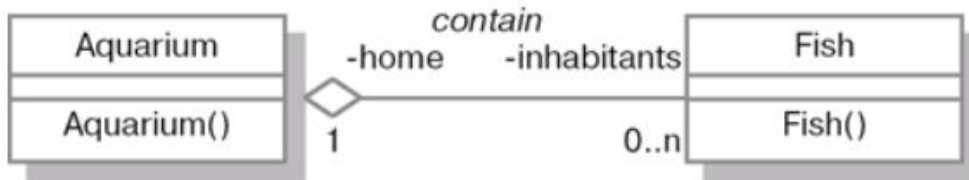
агрегация (Aggregation);
ассоциация (Association);
наследование (Inheritance);
метаклассы (Metaclass).

Приняты также обозначения:

- " 1..n " - от единицы до бесконечности;
- " 0 " - ноль;
- " 1 " - один;
- " n " - фиксированное количество;
- " 0..1 " - ноль или один.

Агрегация

Отношение между *классами* типа "содержит" (contain) или "состоит из" называется агрегацией, или включением. Например, если аквариум наполнен водой и в нем плавают рыбки, то можно сказать, что аквариум агрегирует в себе воду и рыбок.



```
// определение класса Fish
public class Fish {
    // определения поля home
    // (ссылка на объект Aquarium)
    private Aquarium home;

    public Fish() {
    }
}
```

```
// определение класса Aquarium
public class Aquarium {
    // определения поля inhabitants
    // (массив ссылок на объекты Fish)
    private Fish inhabitants[];
    public Aquarium() {
    }
}
```

Задача

Найти наименьшее число имеющие n десятичных разрядов, кратное числам 2,3,5,7.

Указание n – может быть очень большим, например, $n = 1000$.

$$10^k = 10 * 10 * \dots * 10$$

$$(10^k) \bmod p$$

$$(A * B) \bmod p = (A \bmod p) * (B \bmod p)$$

$$5 * 7 \bmod 2 = 1 = 1 * 1 = 1$$