Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования «Российская академия народного хозяйства и государственной службы при Президенте Российской Федерации»

Нижегородский институт управления

Кафедра Информатики и информационных технологий

ОТЧЕТ

Задания лекции №2

Выполнил: студент группы:

ИБ-321

Савицкий Руслан Станиславович

Преподаватель:

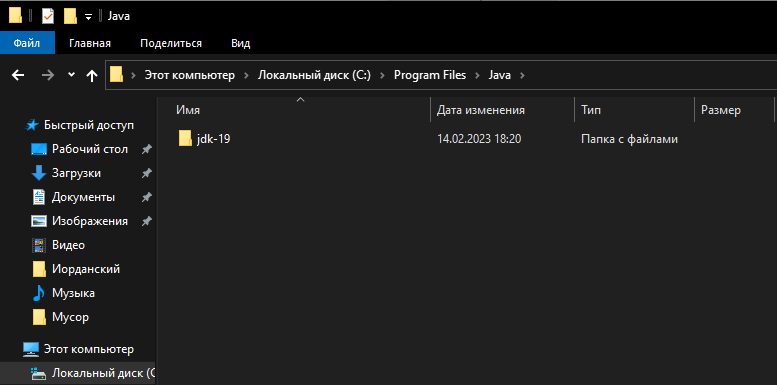
Окулич Евгения Викторовна

Нижний Новгород

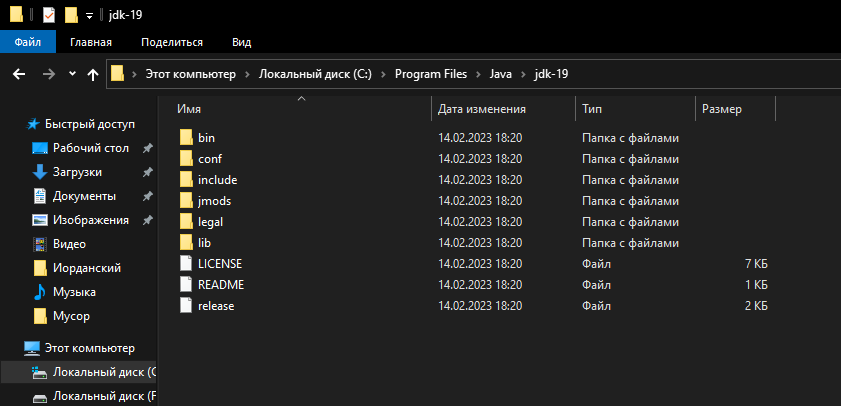
2022 г.

**Задание 1**

**Шаг 1. Установка jdk и работа с содержанием соответствующих каталогов**



Раскрываем папку jdk, чтобы узнать её состав:

 Рисунок 2.2. Совокупность папок первого уровня, содержащих файлы java

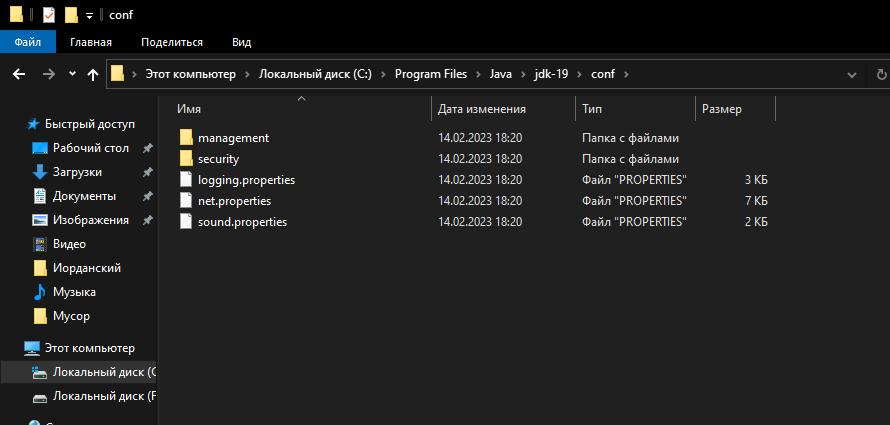


Рисунок 2.3. Содержимое папки conf

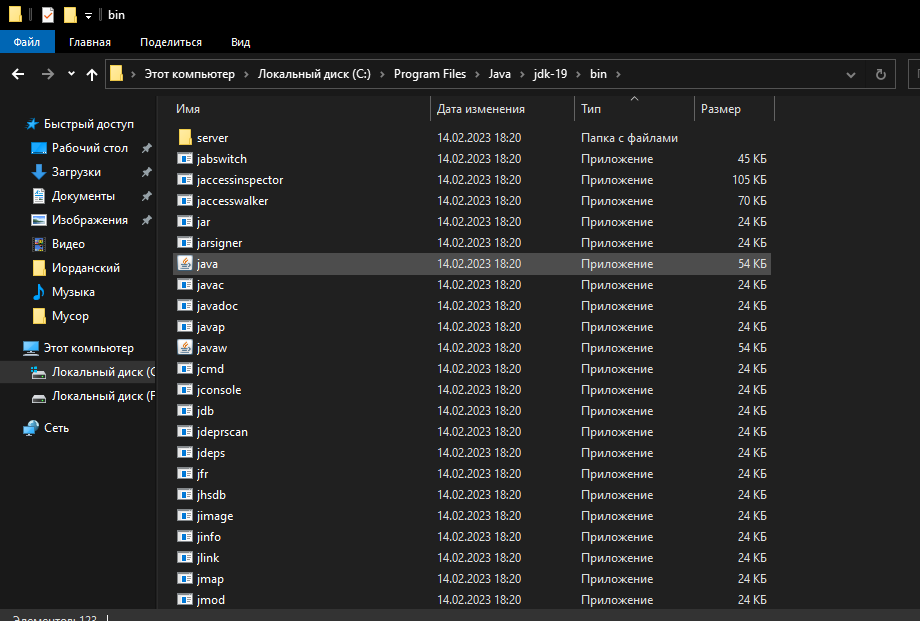
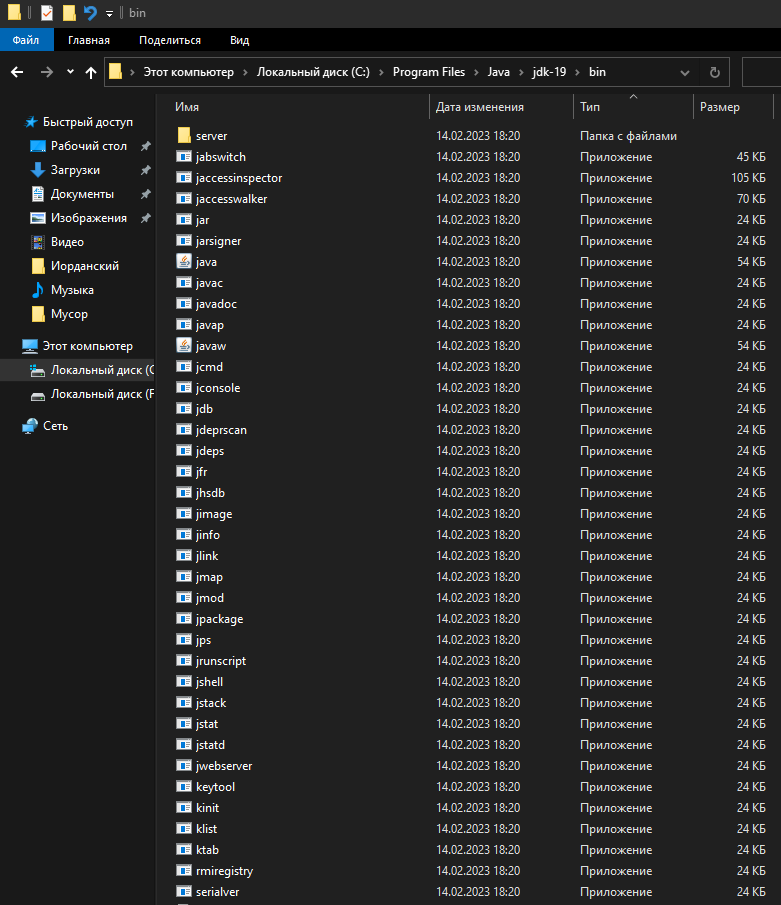


Рисунок **2.4. Содержимое папки bin.**

**Задание 2**

**Шаг 1. Узнать о назначении базовых исполняемых файлов (утилит) с расширением .exe.**



**Шаг 2. Создание первого приложения на языке Java**

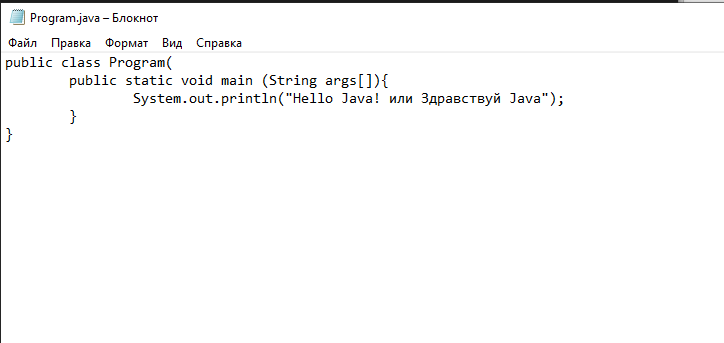


Рисунок 2.5. Код первой программы на Java в Блокноте

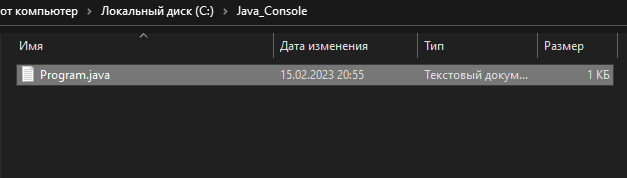


Рисунок 2.6. Вид программы в каталоге

**Шаг 3. Объяснение кода программы**

Java является объектно-ориентированным языком, поэтому вся программа представляется в виде набора взаимодействующих классов**. В данном случае определен один класс Program.**

При определении класса вначале идет модификатор доступа **public**, который указывает, что данный класс будет доступен всем, то есть мы сможем его запустить из командной строки. Далее идет ключевое слово class, а затем название класса. То есть класс называется Program. После названия в фигурных скобках расположено содержимое класса.

Класс может содержать различные переменные и методы. В данном случае у нас объявлен **один метод main**.

**Это главный метод в любой программе на Java, он является входной точкой программы и с него начинается все управление. Он обязательно должен присутствовать в программе**.

Метод main также имеет модификатор **public**. Слово **static** указывает, что метод main - статический, а слово **void** - что он не возвращает никакого значения. Позже мы подробнее разберем, что все это значит.

Далее в скобках у нас идут **параметры метода** - **String args[]** - это массив args, который хранит значения типа String, то есть строки.

В данном случае ни нам пока не нужны, но в реальной программе это те строковые параметры, которые передаются при запуске программы из командной строки.

После списка параметров в **фигурных скобках** идет **тело метода** - это те инструкции, которые и будет выполнять метод. В данном случае фактически определена только одна **инструкция** - вывод на консоль некоторой строки.

Для вывода на консоль используется **встроенный** метод System.out.println(). В этот метод передается выводимая строка. Каждая инструкция завершается точкой с запятой.

**Шаг 4. Компиляция и выполнение файла java через консоль**



Рисунок 2.7.

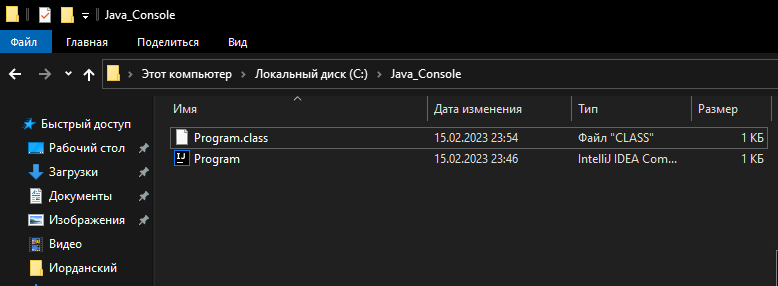


Рисунок 2.7. Содержимое папки после компиляции кода программы.

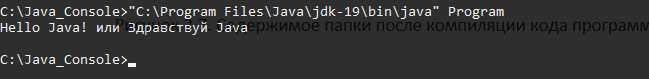


Рисунок 2.8. Команда на выполнение откомпилированной программы java

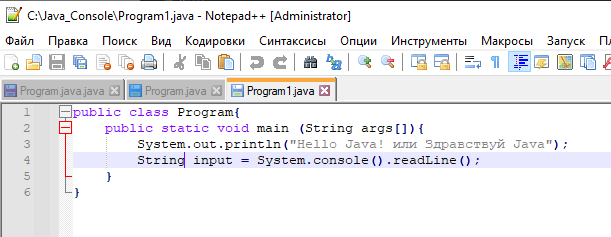


Рисунок 2.9. Программа с вариантом ввода с консоли и клавиатуры

Проделаем те же, что и с программой Program.java, манипуляции с компиляцией и запуском программыProgram1 (рисунок 2.10).

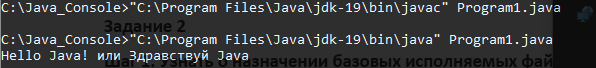


Рисунок 2.10

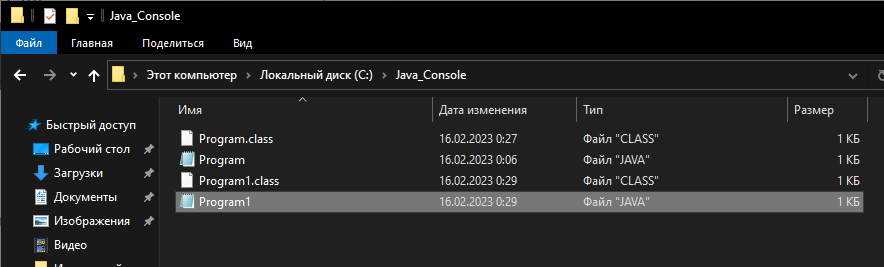
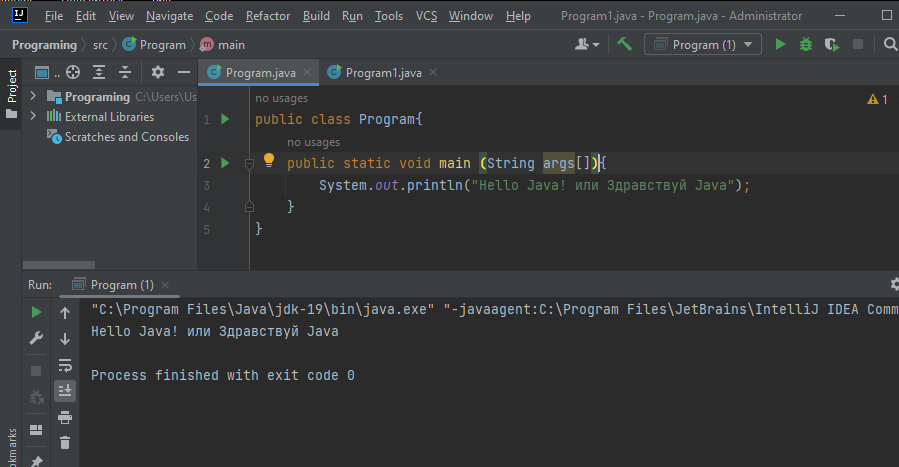


Рисунок 2.11. Содержимое папки после компиляции программы Program1.

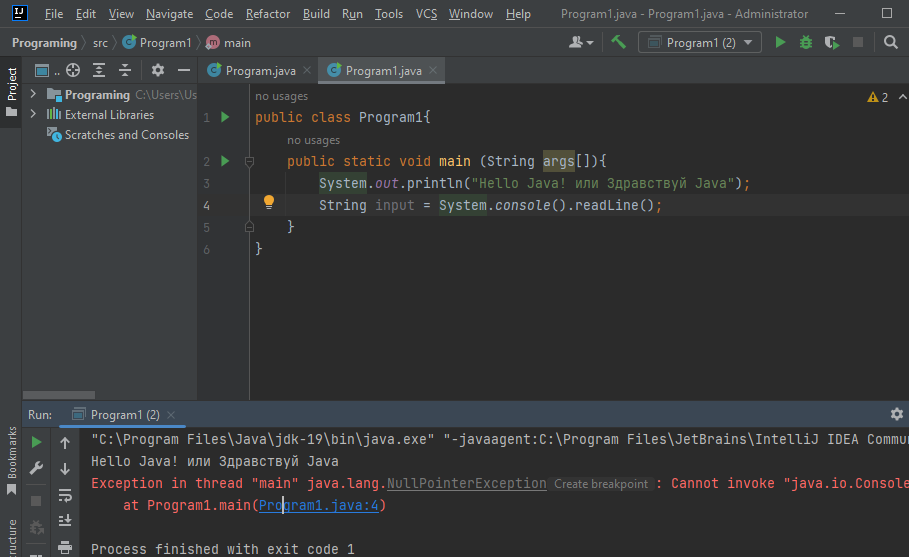
***В моём случае программа запускалась непосредственно из консоли и соответственно выполнялась в ней же. Таким образом независимо от того прекращена программа или нет, результат ее выполнения сохраняется.***

**Задание 3. Проверить версии программ 1 и 2.**

**2.2 Запуск программы в IntelliJ Idea**



Первая программа проверена и успешно запущена



Появляется ошибка которая не устраняется встроенными инструментами.