

Лабораторная работа №1

Задание на лабораторную работу

В процессе написания тестовых заданий ознакомиться со структурой исходного кода для Java, изучить особенности областей видимости и использования пакетов.

Каждое следующее задание надо выполнять в новом каталоге (например, Task1, Task2 и так далее).

Эта лабораторная работа выполняется без применения среды разработки.

Задание -1

Внимательно выслушать информацию, предоставленную преподавателем. В процессе выполнения лабораторной работы понадобятся некоторые знания, изложенные в первых лекциях: JVM, исходный код, байт-код, объекты и классы, поля и методы (в том числе статические), конструкторы и создание объектов, модификаторы доступа, пакеты. Примеры соответствующих синтаксических конструкций также имеются в лекциях.

Задание 0

Запустите компилятор `javac` без параметров и ознакомьтесь с форматом задания параметров компилятора.

Запустите программу `java` без параметров и ознакомьтесь с форматом задания параметров запуска виртуальной машины Java (JVM).

Задание 1

Создайте файл `MyFirstProgram.java`, содержащий исходный код одного пустого класса с именем `MyFirstClass`:

--- `MyFirstProgram.java` -----

```
class MyFirstClass {  
  
}
```

Откомпилируйте его с помощью компилятора `javac`. Для этого необходимо выполнить команду `"javac MyFirstProgram.java"`. Запустите полученный файл класса на выполнение с помощью команды `"java <Имя класса>"`.

Обратите внимание на то, что на вход компилятора необходимо подавать имя файла с расширением, а на вход JVM – без расширения.

Также обратите внимание на то, что в операционных системах, в которых имена файлов чувствительны к регистру, имена файлов для компиляции и запуска следует указывать с учетом регистра. При этом расширение файла с исходным кодом должно быть `".java"` (маленькими буквами).

Добавьте в класс метод `main()`:

```
--- MyFirstProgram.java -----  
  
class MyFirstClass {  
  
    void main(String[] s) {  
  
        System.out.println("Hello world!!!");  
  
    }  
  
}
```

Точкой входа программы является метод `main()` с параметрами `String[] s`.

Для вывода текстовой информации на экран в данном случае используется функция `println()` класса `PrintStream`. Поле `out` этого типа является статическим полем класса `System`, поэтому может использоваться без создания объекта типа `System`.

Откомпилируйте и запустите программу на выполнение.

Логично предположить, что метод `main()` должен быть статическим, т.к. на момент запуска программы ни одного объекта типа `MyFirstClass` не существует. Сделайте метод статическим, снова откомпилируйте и запустите программу.

Внесите в текст программы необходимые для ее запуска изменения, откомпилируйте и запустите программу.

Задание 2

Замените текст метода `main()` на следующий:

--- MyFirstClass.main -----

```
for (int i = 0; i < s.length; i++) {  
  
    System.out.println(s[i]);  
  
}
```

Откомпилируйте и запустите программу, добавив в командную строку ряд аргументов. Например, следующим образом: "java MyFirstClass arg1 arg2 arg3 arg4 arg5".

Задание 3

В том же файле MyFirstProgram.java после описания класса MyFirstClass добавьте описание второго класса MySecondClass, реализующего следующую функциональность:

- имеет два приватных поля типа int;
- методы для получения и модификации их значений;
- конструктор, создающий объект и инициализирующий значения полей;
- метод с возвращаемым типом int, реализующий над этими числами действие в соответствии с вариантом.

Код метода MyFirstClass.main() при этом следует заменить на следующий (в угловых скобках указаны элементы, имена которых следует подставить при написании программы):

--- MyFirstClass.main -----

<Создание и инициализация объекта "o" типа MySecondClass>;

System.out.println(o.<Метод, реализующий действие над числами>());

for (int i = 1; i <= 8; i++) {

for (int j = 1; j <= 8; j++) {

o.<Метод установки значения первого числового поля>(i);

o.<Метод установки значения второго числового поля>(j);

System.out.print(o.<Метод, реализующий действие над числами>());

System.out.print(" ");

```
    }  
  
    System.out.println();  
}
```

Откомпилируйте и запустите программу.

Задание 4

Удалите все откомпилированные байт-коды классов. Вынесите код класса `MySecondClass` без изменений в отдельный файл с именем `MyFirstPackage.java`, и поместите его в поддиректорию `myfirstpackage`, откомпилируйте. Попробуйте откомпилировать файл `MyFirstProgram.java`.

Добавьте в начало исходного кода в файле `MyFirstProgram.java` следующий код:

--- `MyFirstProgram.java` -----

```
import myfirstpackage.*;
```

Снова попробуйте откомпилировать `MyFirstProgram.java`. Далее, следуя сообщениям компилятора и изменяя исходный код программы, добейтесь ее работоспособности.

Задание 5

Запустите программу `jar`, предназначенную для создания архивов, и ознакомьтесь с форматом задания ключей для формирования архивов.

Скопируйте в рабочую папку, сохранив структуру каталогов, только файлы с расширением `class`, полученные в результате выполнения задания 4.

Создайте файл `manifest.mf`, содержащий следующий код:

--- `manifest.mf` -----

Manifest-Version: 1.0

Created-By: <Ваши фамилии>

Main-Class: `MyFirstClass`

Обратите внимание на то, что после имени класса надо обязательно поставить символ новой строки.

Создайте архив myfirst.jar, включив в него полученные ранее файлы классов и указав созданный вами манифест-файл. Переместите полученный файл в другую директорию (например, поддиректорию MyJar) и запустите его на выполнение.

Варианты для задания 3

1	Сложение
2	Вычитание
3	Умножение
4	Целочисленное деление
5	Остаток от деления
6	Минимум
7	Максимум
8	Битовое «И»
9	Битовое «ИЛИ»
10	Битовое исключающее «ИЛИ»
11	Битовый сдвиг влево

