

# Лабораторная работа №1

## «Базовые средства разработки для языка Java»

Скоробогатов С.Ю.

11 февраля 2016 г.

### 1 Цель работы

Целью данной работы является формирование комфортного окружения для разработки программного обеспечения на языке Java.

### 2 Исходные данные

Для выполнения лабораторной работы потребуются дистрибутивы Java Development Kit (JDK) версии 1.8, а также IntelliJ IDEA 15 Community Edition.

Подходящий для целевой платформы вариант JDK можно скачать на сайте корпорации Oracle:

[www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html](http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html)

Дистрибутив интегрированной среды разработки IntelliJ IDEA нужно взять со страницы [www.jetbrains.com/idea/download/](http://www.jetbrains.com/idea/download/)

### 3 Задание

Лабораторная работа состоит из четырёх этапов, которые должны выполняться последовательно. Выполнение лабораторной работы возможно на любом компьютере с любой операционной системой, но в описании каждого этапа подразумевается наличие 64-разрядного компьютера под управлением ОС Linux.

#### 3.1 Установка Java Development Kit

Мы будем устанавливать JDK в домашний каталог пользователя. Этот способ не требует административных привилегий и не зависит от наличия нужной версии JDK в репозиториях используемого дистрибутива Linux.

Скачайте файл `jdk-8u73-linux-x64.tar.gz` с сайта Oracle и разархивируйте его в домашний каталог. При этом в домашнем каталоге должен появиться подкаталог с именем `jdk1.8.0_73`.

Стандартные средства разработки языка Java находятся в подкаталоге `jdk1.8.0_73/bin`. Чтобы они могли правильно работать, им нужно знать, в каком каталоге находится JDK. Для этого мы должны присвоить путь к каталогу с JDK переменной окружения `JAVA_HOME`.

Кроме того, нам потребуется запускать эти средства из любого рабочего каталога, поэтому нужно добавить путь к каталогу `bin` в переменную `PATH`.

Присваивание значений переменным окружения выполняется командой `export` командного интерпретатора `bash`. При этом эту команду удобно вызывать автоматически из скрипта `.profile` (или `.bash_profile`, или `.bashrc`), находящегося в корне домашнего каталога.

Отредактируйте файл `.profile` (или `.bash_profile`, или `.bashrc`), добавив в его конец две строчки:

```
export JAVA_HOME=$HOME/jdk1.8.0_73
export PATH=$JAVA_HOME/bin:$PATH
```

Если вы редактировали файлы `.profile` или `.bash_profile`, вам придётся выйти из системы и повторно в неё войти (перелогиниться), потому что эти скрипты запускаются однократно при входе в систему. В случае, если вы изменяли файл `.bashrc`, достаточно просто перезапустить терминал, так как он выполняется при каждом запуске `bash`.

Для проверки правильности установки JDK запустите терминал и введите команду

```
javac -version
```

В результате вы должны увидеть сообщение

```
javac 1.8.0_73
```

## 3.2 Компиляция и запуск тестовой программы с помощью базовых средств JDK

Создайте файл `Factorial.java` следующего содержания:

```
1 import java.util.stream.IntStream;
2
3 public class Factorial {
4     public static void main(String[] args) {
5         if (args.length == 0) {
6             System.out.println("usage: _java_Factorial_x");
7         } else {
8             int n = Integer.parseInt(args[0]);
9             int f = IntStream.range(1, n+1).reduce(1, (r,x) -> r*x);
10            System.out.println(f);
11        }
12    }
13 }
```

Откомпилируйте его с помощью команды

```
javac Factorial.java
```

Убедитесь, что в текущем каталоге появился файл `Factorial.class` и запустите его командой

```
java Factorial 3
```

Программа должна вывести число 6.

### 3.3 Установка IntelliJ IDEA

Скачайте с сайта компании JetBrains дистрибутив IntelliJ IDEA Community Edition, который представляет собой архивный файл `ideaIC-15.0.3.tar.gz`.

Разархивируйте дистрибутив в домашний каталог. При этом в домашнем каталоге появится подкаталог `idea-IC-143.1821.5`.

Отредактируйте скрипт `.profile` (или `.bash_profile`, или `.bashrc`), добавив каталог `idea-IC-143.1821.5/bin`, в котором расположены исполняемые файлы IntelliJ IDEA, в переменную `PATH`:

```
export PATH=$HOME/idea-IC-143.1821.5/bin:$PATH
```

### 3.4 Создание проекта в IntelliJ IDEA, его сборка и запуск

Запустите IntelliJ IDEA, выполнив в терминале команду

```
idea.sh
```

Так как IDEA запускается первый раз, появится диалоговое окно, в котором нужно выбрать пункт `Create New Project`, после чего откроется окно создания нового проекта. Создание проекта – необходимое условие работы с интегрированной средой разработки, потому что проект, объединяющий набор файлов с исходным текстом программы, является в ней единицей сборки и выполнения.

В окне создания нового проекта, прежде всего, укажите путь к каталогу, в который вы установили JDK. Для этого нажмите кнопку `New`, в появившемся выпадающем списке нажмите на пункт `JDK` и в окне выбора каталога перейдите на каталог `jdk1.8.0_73`, расположенный в вашем домашнем каталоге. IDEA будет использовать именно указанную вами версию JDK, даже если на вашем компьютере присутствуют другие версии.

Два раза нажмите на кнопку `Next` и введите название вашего проекта – `FactorialProject`. Отметим, что IDEA создаст для вашего проекта каталог `~/IdeaProjects/FactorialProject`. Именно там нужно будет, в случае чего, искать ваши файлы.

Для создания проекта нажмите кнопку `Finish`, в результате чего откроется главное окно среды разработки. В левой части окна располагается дерево наших проектов, содержащее проект `FactorialProject`. Так как мы собираемся использовать возможности Java 1.8, а именно – замыкания, мы должны включить их поддержку для нашего проекта. Для этого выберите пункт меню `File|Project Structure`, выберите «8 – Lambdas, type annotations etc» в выпадающем списке `Project Language Level` и нажмите `OK`.

Настало время для добавления файлов в наш проект. Для этого раскройте его в дереве проектов, перейдите на пункт `src` и в контекстном меню выберите `New|Java Class`. Появится диалоговое окно с предложением ввести имя класса. Введите `Factorial` и нажмите `OK`. После этого IDEA создаст файл с именем `Factorial.java` и откроет его в окне редактора. Замените в редакторе «заготовку» класса `Factorial`, которую создала для нас IDEA, на код из пункта 3.2.

Для сборки нашего проекта выберите пункт меню `Build|Make Project`. Если проект не содержит ошибок, в левом нижнем углу главного окна IDEA появится надпись «`Compilation completed successfully in x seconds`». В противном случае откроется окно сообщений компилятора с описанием ошибок.

Параметры запуска нашего проекта в IDEA определяются так называемыми *конфигурациями*. Диалговое окно для работы с конфигурациями открывается путём выбор пункта меню Run|Edit Configurations. Для создания новой конфигурации нажмите на кнопку с изображением знака «+» и в выпадающем списке выберите Application. После этого IDEA создаст новую безымянную конфигурацию и даст нам возможность редактировать параметры этой конфигурации.

Дайте новой конфигурации имя FactorialApp, в качестве главного класса (Main class) укажите Factorial и задайте число, факториал которого хотите посчитать, в аргументах командной строки (Program arguments). Затем нажмите ОК.

Теперь можно запустить проект, выбрав в меню Run|FactorialApp. В нижней части главного окна IDEA покажет содержимое стандартного потока вывода нашей программы.