

ФЕДЕРАЛЬНОЕ АГЕНТСТВО СВЯЗИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ТЕЛЕКОММУНИКАЦИЙ ИМ. ПРОФ. М.А. БОНЧ-БРУЕВИЧА»
(СПбГУТ)

Институт магистратуры
Кафедра информационных управляющих систем
Дисциплина «Аспектно-ориентированное программирование»

Отчет
к лабораторной работе № 1

Выполнил:
Студент группы ИСТ-012м
Тарасенко А.А.

Принял:
Ст. преподаватель Антонов В.В.

Санкт-Петербург
2020

Задание: изучить следующие способы создания проектов с фреймворком Maven.

1. Создание проекта вручную;
2. Создание проекта при помощи spring.io;
3. Загрузка репозитория готового проекта при помощи Git.

Ход работы:

1 Создание проекта вручную

На данном и последующих этапах в качестве среды разработки используется IntelliJ IDEA, но аналогичные действия могут быть выполнены и в любой среде разработки, поддерживающей Java.

Для создания Maven-проекта достаточно просто выбрать соответствующий пункт в меню создания проекта (рисунок 1):

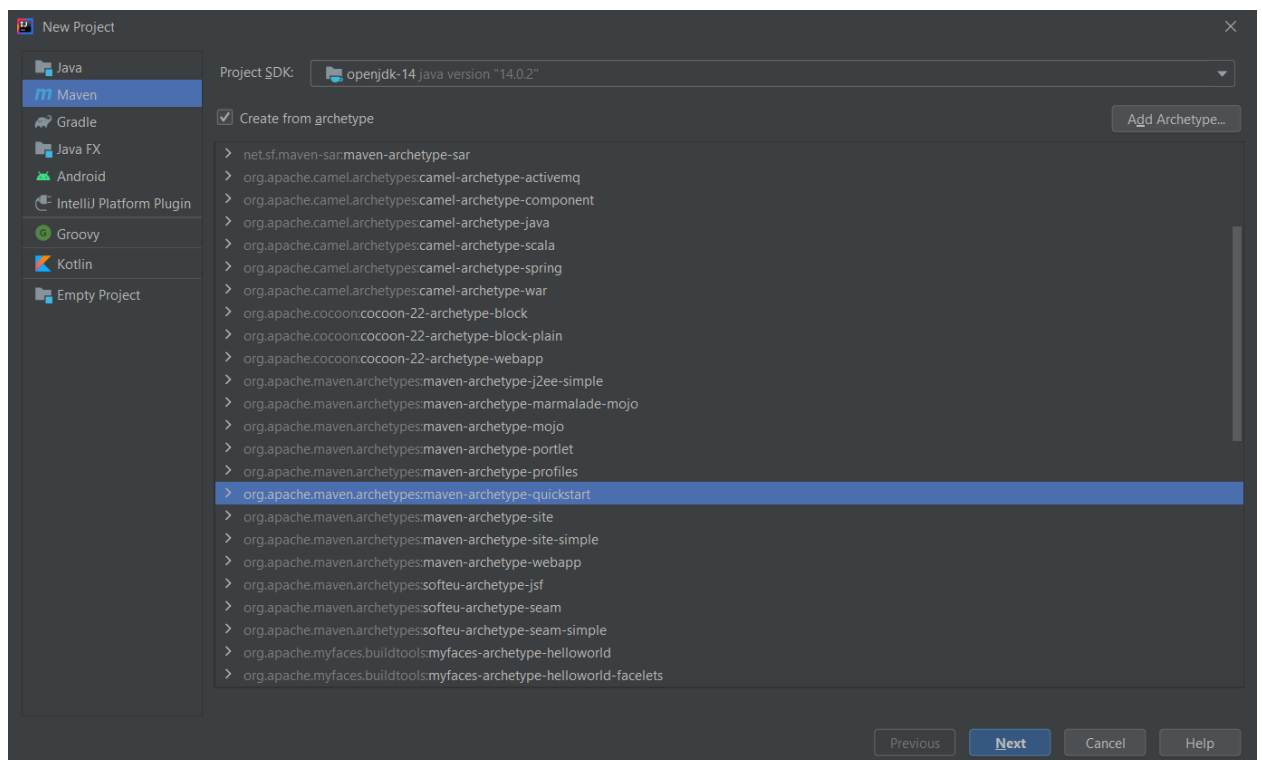
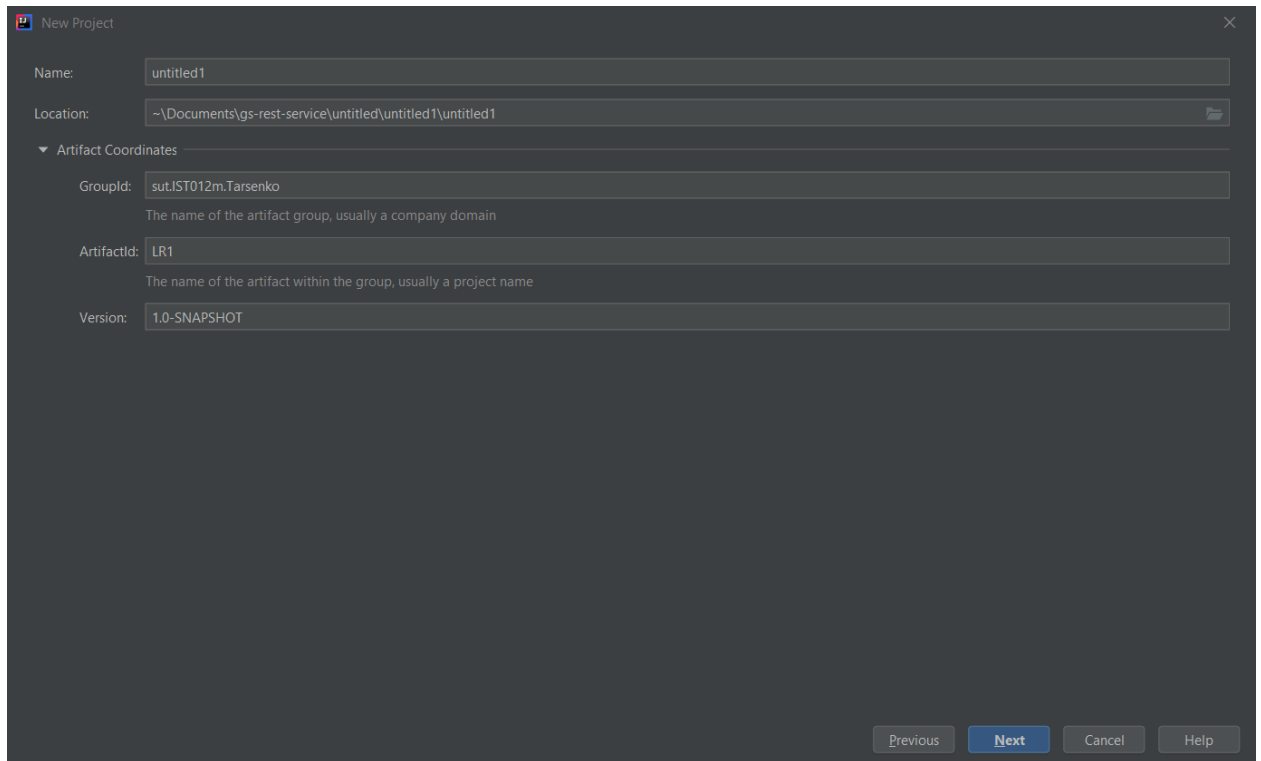


Рисунок 1 – выбор типа проекта

Также для удобства можно выбрать архетип, на основе которого будет создан проект. Для этого достаточно выбрать опцию “Create from archetype”, после чего выбрать нужный архетип.

Далее на рисунке 2 представлено описание создаваемого проекта, в котором можно указать название проекта, путь, по которому он будет располагаться, а также идентификаторы GroupId и ArtifactId.



The screenshot shows a 'New Project' dialog box with the following fields and values:

- Name: untitled1
- Location: ~\Documents\gs-rest-service\untitled\untitled1\untitled1
- Artifact Coordinates section:
 - GroupId: sut.IST012m.Tarsenko (with a tooltip: 'The name of the artifact group, usually a company domain')
 - ArtifactId: LR1 (with a tooltip: 'The name of the artifact within the group, usually a project name')
 - Version: 1.0-SNAPSHOT

At the bottom right, there are four buttons: 'Previous' (disabled), 'Next' (active), 'Cancel', and 'Help'.

Рисунок 2 – описание проекта

После чего проект создан и готов к работе.

В проекте уже содержится pom-файл:

```
<?xml version="1.0" encoding="UTF-8"?>

<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>

  <groupId>sut.IST012M.Tarasenko</groupId>
  <artifactId>LR1</artifactId>
  <version>1.0-SNAPSHOT</version>

  <name>LR1</name>
  <!-- FIXME change it to the project's website -->
  <url>http://www.example.com</url>

  <properties>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    <maven.compiler.source>1.7</maven.compiler.source>
    <maven.compiler.target>1.7</maven.compiler.target>
  </properties>

  <dependencies>
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>4.11</version>
      <scope>test</scope>
    </dependency>
  </dependencies>

  <build>
    <pluginManagement><!-- lock down plugins versions to avoid using Maven defaults (may be moved to parent pom) -->
      <plugins>
```

Рисунок 3 – pom-файл

2 Создание проекта при помощи spring.io

Проект создается после заполнения шаблона на сайте:

The screenshot shows the Spring Boot project creation wizard. The 'Project' section has 'Maven Project' selected. The 'Language' section has 'Java' selected. The 'Spring Boot' section has '2.3.3' selected. The 'Project Metadata' section has the following values: Group: sut.IST012M, Artifact: MavenFromWeb, Name: MavenFromWeb, Description: Demo project for Spring Boot, Package name: sut.IST012M.MavenFromWeb. The 'Packaging' section has 'Jar' selected. At the bottom, there are buttons for 'GENERATE', 'EXPLORE', and 'SHARE'.

Рисунок 4 – шаблон создания проекта

Здесь можно выбрать тип проекта: Maven или Gradle; язык, версию Spring, указать данные проекта и указать зависимости.

После заполнения необходимых полей и нажатия кнопки Generate, скачивается архив с собранным проектом.

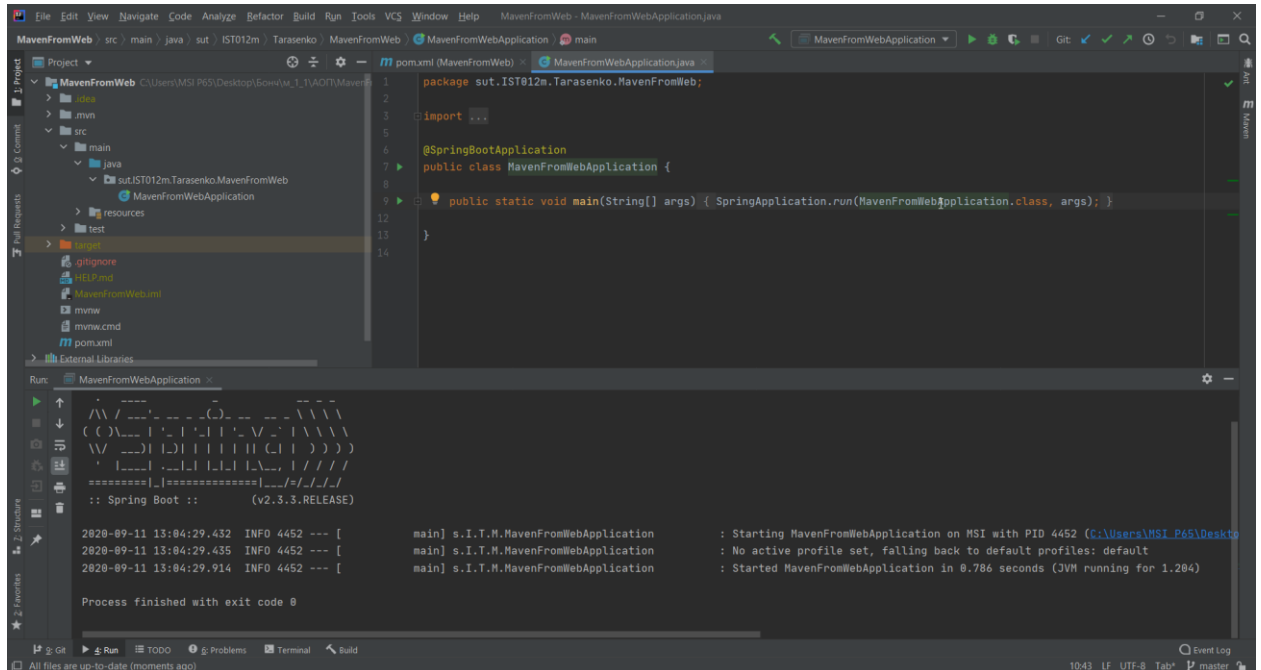


Рисунок 5 – запуск сгенерированного проекта

3 Загрузка репозитория готового проекта при помощи Git

Скачаем проект из открытого репозитория на GitHub:

```
MSI P65@MSI MINGW64 ~/Desktop/Бонч/м_1_1/AOP
$

MSI P65@MSI MINGW64 ~/Desktop/Бонч/м_1_1/AOP
$

MSI P65@MSI MINGW64 ~/Desktop/Бонч/м_1_1/AOP
$ mkdir git_test

MSI P65@MSI MINGW64 ~/Desktop/Бонч/м_1_1/AOP
$ cd git_test/

MSI P65@MSI MINGW64 ~/Desktop/Бонч/м_1_1/AOP/git_test
$ git clone https://github.com/Yarr8/LR1_1
Cloning into 'LR1_1'...
remote: Enumerating objects: 18, done.
remote: Counting objects: 100% (18/18), done.
remote: Compressing objects: 100% (7/7), done.
remote: Total 18 (delta 0), reused 18 (delta 0), pack-reused 0
Unpacking objects: 100% (18/18), 1.94 KiB | 19.00 KiB/s, done.

MSI P65@MSI MINGW64 ~/Desktop/Бонч/м_1_1/AOP/git_test
$ |
```

Рисунок 6 – загрузка проекта из репозитория

После чего проект можно открыть в среде разработки и работать с ним как обычно.

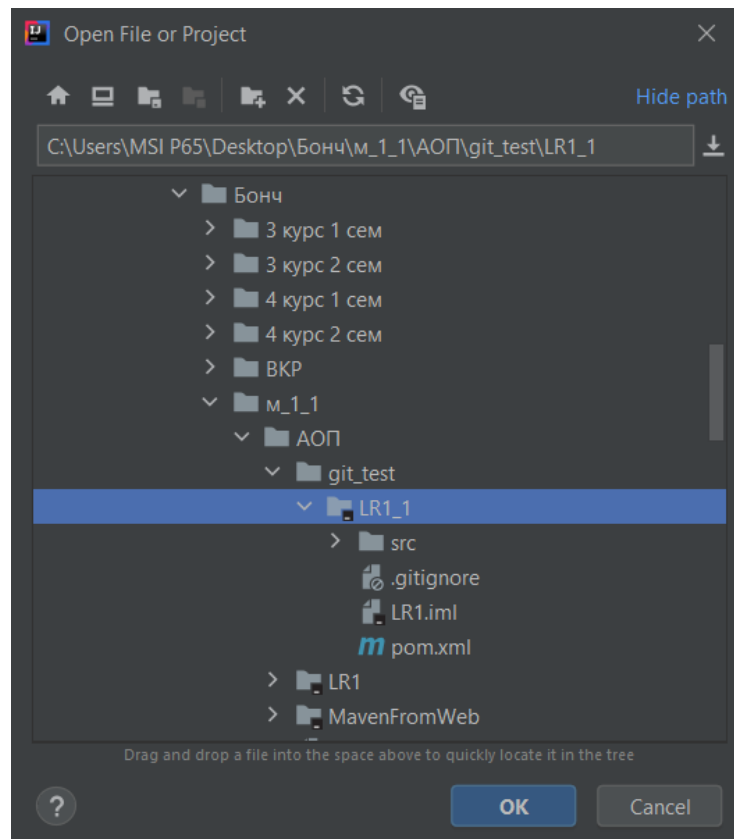


Рисунок 7 – выбор папки с проектом

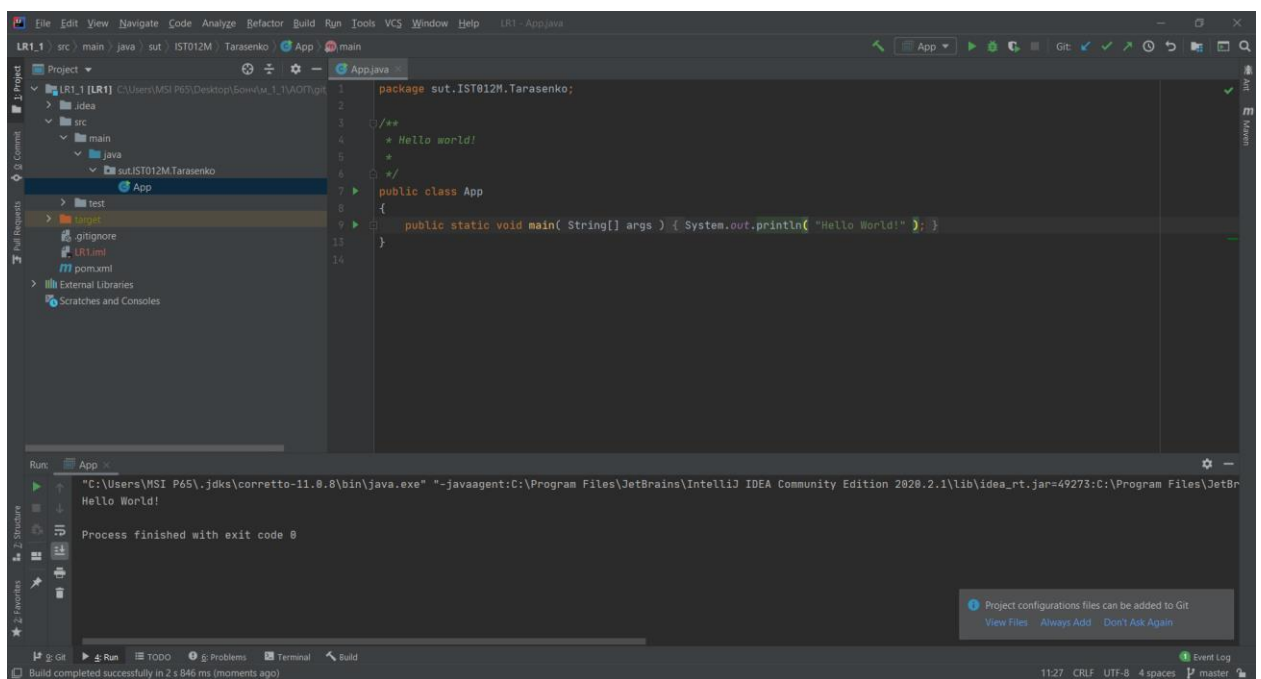


Рисунок 8 – успешный запуск проекта