

Отчет по лабораторной работе 2

Студентов группы ПИМ-21 Бубенцова С.А. Носкова И.А.

1 Постановка задачи

В процессе выполнения лабораторной работы необходимо выполнить следующие задачи:

1. Сделать проект с помощью maven. Добавить в проект зависимость и продемонстрировать ее работу.
 2. Добавить jar библиотеку в локальный репозиторий maven. Подключить ее как зависимость и продемонстрировать работу.
 3. Сделать родительский проект и два дочерних с помощью maven. Один дочерний зависим от другого.
-

2 Разработка

2.1 Структура проекта

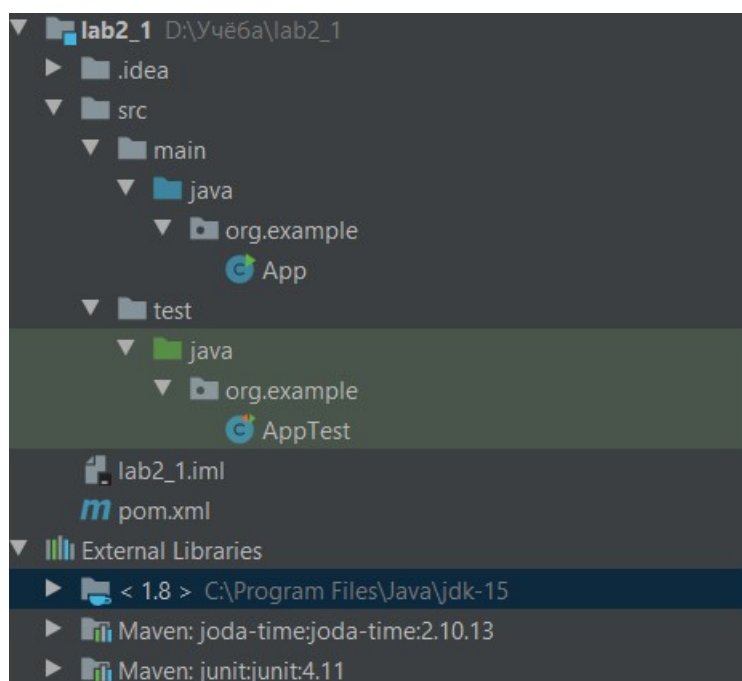


Рисунок 1. Структура проекта первого задания

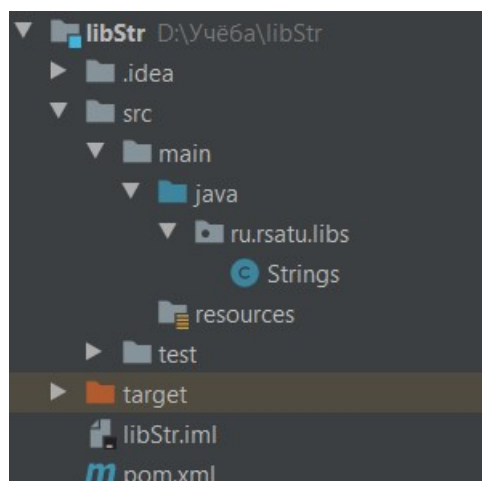


Рисунок 2. Структура проекта создания jar в локальном репозитории

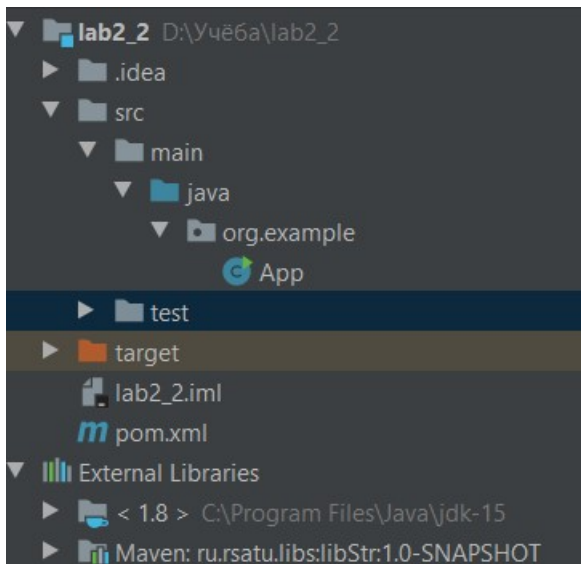


Рисунок 3. Структура проекта второго задания

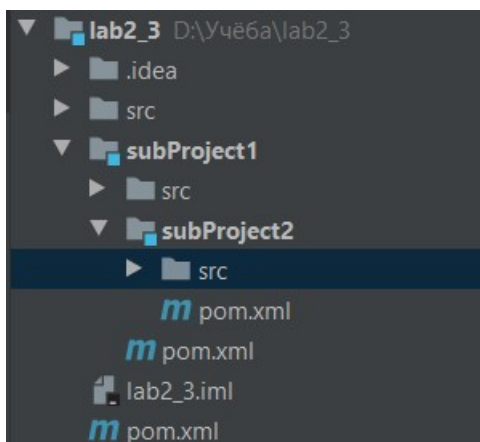


Рисунок 4. Структура проекта третьего задания

2.2 Задание 1

При создании проекта была проведена идентификация.

Листинг 1. Идентификация пакета

```
<groupId>org.example</groupId>
<artifactId>lab2_1</artifactId>
<version>1.0-SNAPSHOT</version>
```

Далее в проект добавлены класс App и зависимости: JUnit (библиотека для модульного тестирования) и joda-time (библиотека для работы с датой и временем).

Листинг 2. Листинг класса App

```
package org.example;

import org.joda.time.LocalDateTime;

public class App
{
    public static void main( String[] args )
    {
        LocalDateTime now = new LocalDateTime();
        System.out.println( "Current time: " + now);
    }
}
```

Для демонстрации работы библиотеки JUnit, был добавлен модульный тест AppTest.

Листинг 3. Листинг класса AppTest

```

package org.example;

import static org.junit.Assert.assertTrue;

import org.junit.Test;

/**
 * Unit test for simple App.
 */
public class AppTest
{
    @Test
    public void shouldAnswerWithTrue()
    {
        assertTrue( true );
    }
}

```

Для добавления зависимостей в файл `pom` добавлена секция `dependencies`.

Листинг 4. Секция `dependencies`

```

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

  <dependency>
    <groupId>joda-time</groupId>
    <artifactId>joda-time</artifactId>
    <version>2.10.13</version>
  </dependency>
</dependencies>

```

2.3 Задание 2

В локальном репозитории создана `jar` библиотека с классом `Strings`.





Локальный диск (C:) > Пользователи > buben > .m2 > repository > ru > rsatu > libs > libStr > 1.0-SNAPSHOT				
Имя	Дата изменения	Тип	Размер	
 <code>_remote.repositories</code>	19.09.2021 9:11	Файл "REPOSITOR...	1 КБ	
 <code>libStr-1.0-SNAPSHOT.jar</code>	19.09.2021 9:11	Executable Jar File	3 КБ	
 <code>libStr-1.0-SNAPSHOT.pom</code>	19.09.2021 9:11	Файл "POM"	1 КБ	
 <code>maven-metadata-local.xml</code>	19.09.2021 9:11	Файл "XML"	1 КБ	

Рисунок 5. Локальный репозиторий

Листинг 5. Листинг `Strings`

```

package ru.rsatu.libs;

public class Strings {

    public static String str = "lib1";

}

```

Далее создан проект с классом `App`.

Листинг 6. Листинг класса `App`

```

package org.example;

import ru.rsatu.libs.Strings;

public class App
{
    public static void main( String[] args )
    {
        System.out.println("Библиотека Strings: " + Strings.str);
    }
}

```

К проекту в секции `dependencies` была подключена зависимость из локального репозитория.

Листинг 7. Секция `dependencies`

```

<dependencies>
  <dependency>
    <groupId>ru.rsatu.libs</groupId>
    <artifactId>libStr</artifactId>
    <version>1.0-SNAPSHOT</version>
  </dependency>
</dependencies>

```

2.4 Задание 3

Был создан родительский проект. Далее создан первый дочерний проект, к которому подключена зависимость `joda-time`.

Потом создан второй дочерний проект, в котором используется `joda-time`.

В `parent` родительского проекта добавлена секция `modules` для подключения первого дочернего проекта.

Листинг 8. Секция `modules` родительского проекта

```

<modules>
  <module>subProject1</module>
</modules>

```

В первый дочерний проект добавлены `modules` и `parent`.

Листинг 9. `Modules` и `parent` `lab2_3`

```

<parent>
  <artifactId>lab2_3</artifactId>
  <groupId>org.example</groupId>
  <version>1.0-SNAPSHOT</version>
</parent>

<artifactId>subProject1</artifactId>
<version>1.0-SNAPSHOT</version>

<modules>
  <module>subProject3</module>
</modules>

```

Ко второму дочернему проекту добавлена секция `parent`.

Листинг 10. Секция `parent`

```

<parent>
  <groupId>org.example</groupId>
  <artifactId>subProject1</artifactId>
  <version>1.0-SNAPSHOT</version>
</parent>

```

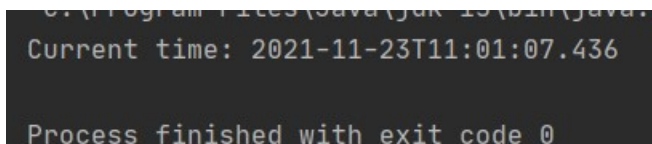
Листинг 11. Листинг `subProject3`

```
import org.joda.time.LocalDateTime;

public class SubProject3 {
    public static void main( String[] args )
    {
        LocalDateTime time = new LocalDateTime();
        System.out.println("Проверка работы библиотеки joda-time из родительского проекта");
        System.out.println(time);
    }
}
```

3 Результат разработки

В результате выполнения первого задания с помощью joda-time выведены текущие дата и время, также был запущен простой тест для демонстрации работы подключенной библиотеки JUnit.



```
C:\Program Files\Java\jdk-15\bin\java.exe
Current time: 2021-11-23T11:01:07.436
Process finished with exit code 0
```

Рисунок 6. Результат выполнения первого задания (Вывод даты и времени)

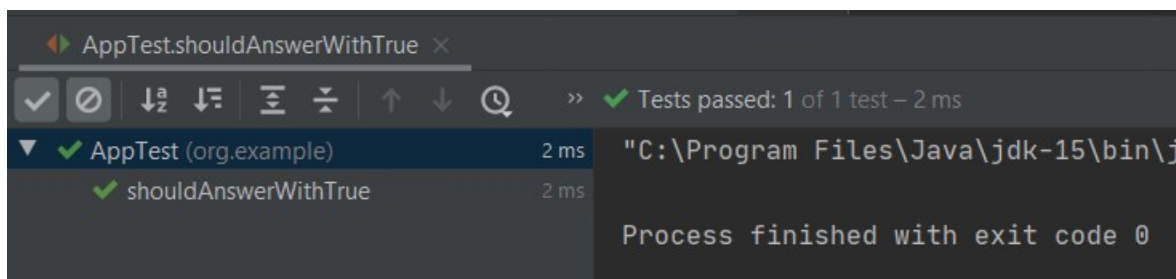
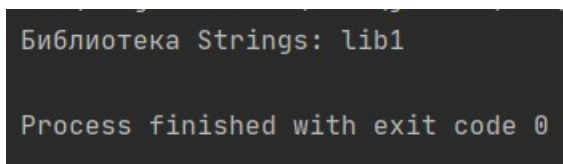


Рисунок 7. Результат выполнения первого задания (Результат выполнения теста)

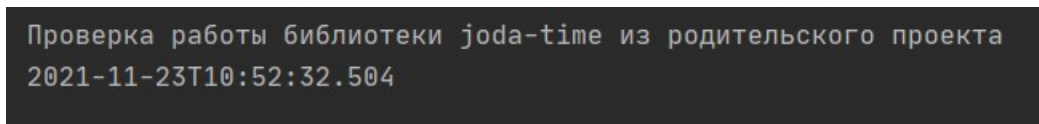
В результате выполнения второго задания было получено содержимое библиотеки, созданной в локальном репозитории.



```
Библиотека Strings: lib1
Process finished with exit code 0
```

Рисунок 8. Результат выполнения второго задания

В результате выполнения третьего задания с помощью joda-time (добавленного к первому дочернему проекту) во втором дочернем выведены дата и время.



```
Проверка работы библиотеки joda-time из родительского проекта
2021-11-23T10:52:32.504
```

Рисунок 9. Результат выполнения третьего задания

Вывод

В результате выполнения лабораторной работы мы познакомились с maven. Научились создавать проект с помощью maven, подключать локальные и внешние зависимости, а также создавать зависимости между родительским и дочерними проектами.

