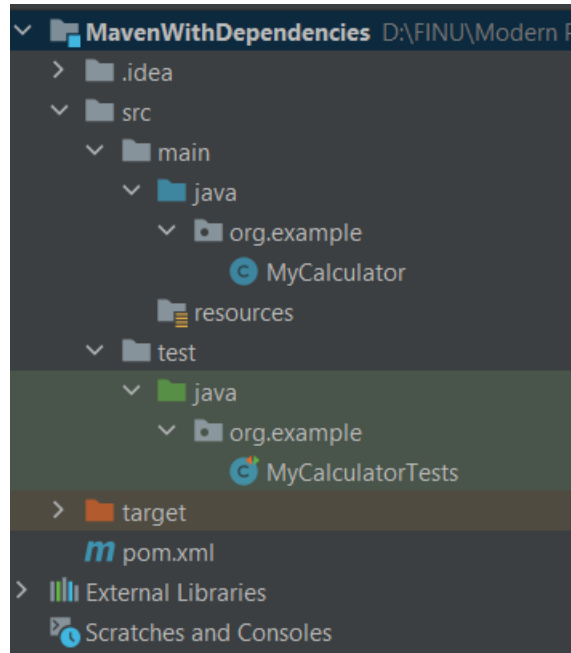


Lab2- Введение в язык программирования Java (часть 2)

Работа с зависимостями и плагинами Maven

1- Создать проект Maven в IntelliJ IDEA со структурой, показанной на рисунке ниже.



2- Изменить файл pom.xml.

```
<?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>org.example</groupId>

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

  <properties>
    <maven.compiler.source>18</maven.compiler.source>
    <maven.compiler.target>18</maven.compiler.target>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    <project.reporting.outputEncoding>UTF-8</project.reporting.outputEncoding>
  </properties>
  <build>
    <plugins>
      <plugin>
        <groupId>org.apache.maven.plugins</groupId>
        <artifactId>maven-surefire-plugin</artifactId>
        <version>3.0.0-M7</version>
      </plugin>
    </plugins>
  </build>
  <dependencies>
    <dependency>
      <groupId>org.junit.jupiter</groupId>
```

```
        <artifactId>junit-jupiter-engine</artifactId>
        <version>5.9.0</version>
        <scope>test</scope>
    </dependency>
</dependencies>

</project>
```

3- Добавить в файл MyCalculator.java следующий код:

```
package org.example;

public class MyCalculator {
    public static double multiply(double x, double y){return x*y;}
    public static double divide(double x, double y){return x/y;}
    public static double add(double x, double y){return x+y;}
    public static double subtract(double x, double y){return x-y;}
}
```

4- Добавить в файл MyCalculatorTests.java следующий код:

```
package org.example;
import org.junit.jupiter.api.DisplayName;
import org.junit.jupiter.api.Test;

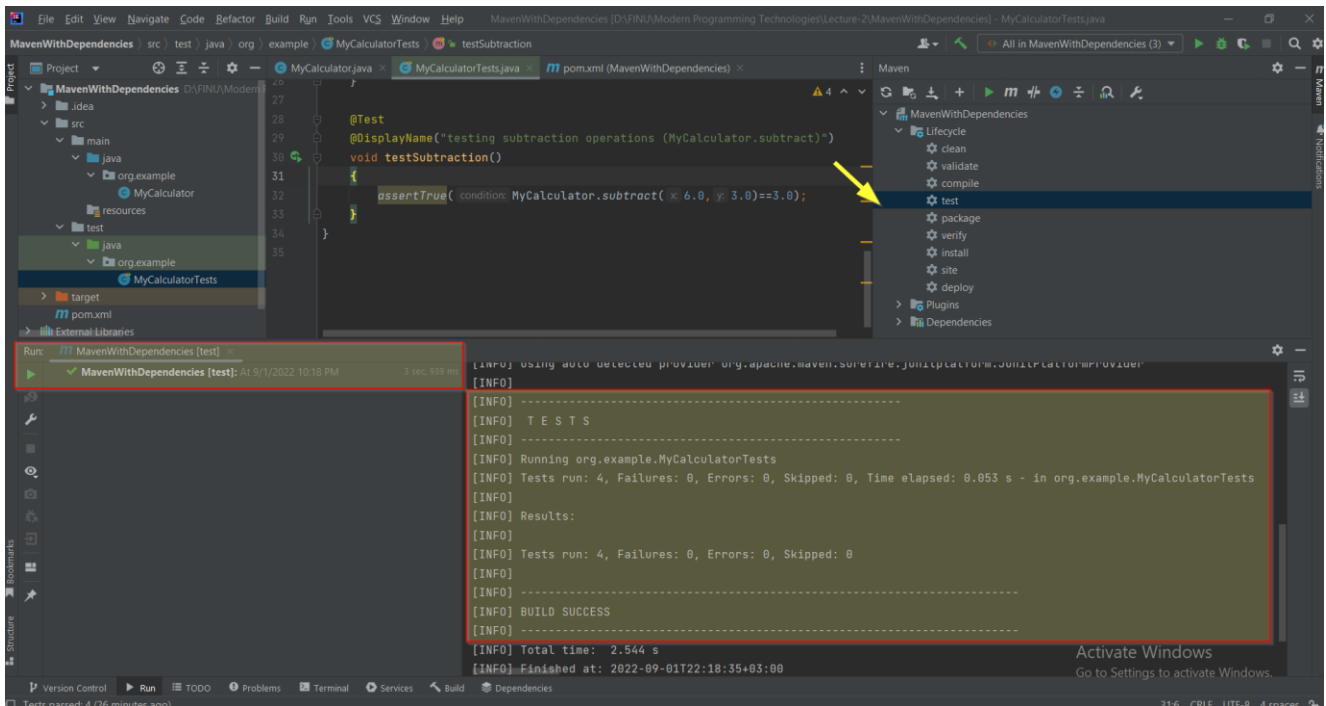
import static org.junit.jupiter.api.Assertions.assertTrue;

public class MyCalculatorTests {
    @Test
    @DisplayName("testing multiplication operations (MyCalculator.multiply)")
    void testMultiplication()
    {
        assertTrue(MyCalculator.multiply(2.0,3.0)==6.0);
    }
    @Test
    @DisplayName("testing division operations (MyCalculator.divide)")
    void testDivision()
    {
        assertTrue(MyCalculator.divide(6.0,3.0)==2.0);
    }

    @Test
    @DisplayName("testing addition operations (MyCalculator.add)")
    void testAddition()
    {
        assertTrue(MyCalculator.add(6.0,3.0)==9.0);
    }

    @Test
    @DisplayName("testing subtraction operations (MyCalculator.subtract)")
    void testSubtraction()
    {
        assertTrue(MyCalculator.subtract(6.0,3.0)==3.0);
    }
}
```

5- Запустить фазу Maven test и обратить внимание на результаты выполнения.



6- Щелкнуть правой кнопкой мыши на (Run 'All Tests') и обратить внимание на результаты в нижней части окна программы.

Переменные в Java

Переменные в Java делятся на:

- Примитивы: примитивные типы хранят значение (см. таблицу ниже).
- Ссылки: ссылочные типы хранят ссылку на объект, или же тип данных null, то есть нулевую (пустую) ссылку. Переменные включают классы (class types), интерфейсы (interface types), переменные типов (type variables), и массивы (array types).

		ТИП ДАННЫХ	ДИАПАЗОН ДОПУСТИМЫХ ЗНАЧЕНИЙ	ОБЪЁМ ЗАНИМАЕМОЙ ПАМЯТИ
целочисленные	[byte	от -128 до 127	1 байт
		short	от -32768 до 32767	2 байта
		int	от -2147483648 до 2147483647	4 байта
		long	от -9223372036854775808 до 9223372036854775807	8 байт
с плавающей точкой	[float	от -3.4E+38 до 3.4E+38	4 байта
		double	от -1.7E+308 до 1.7E+308	8 байт
символы	[char	от 0 до 65536	2 байта
логические	[boolean	true или false	Для хранения значения этого типа достаточно 1 бита, но в реальности память такими порциями не выделяется, поэтому переменные этого типа могут быть по-разному упакованы виртуальной машиной

Пример ниже показывает:

- 1- как создать простой класс (мы подробно изучим классы в следующих уроках).
- 2- как объявлять переменные и вводить и выводить их в командной строке.
- 3- как создавать простые массивы.
- 4- как использовать некоторые языковые конструкции (for, while, do while, switch,if).

```
package org.example;

import java.util.Scanner;

class Person
{
    /*
    статическая переменная который существуют в одном экземпляре
    и обращаться к ним нужно по имени класса
    (внутри класса к статической переменной можно обращаться просто по имени).
    В этом случае переменная подсчитывает количество экземпляров, созданных из этого
    класса.
    */
    private static int counter=0;
    private String name="";
    private int age;

    public Person()
    {
        counter++;
    }
    public Person(String name)
    {
        counter++;
        this.setName(name);
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public int getAge() {
        return age;
    }

    public void setAge(int age) {
        this.age = age;
    }

    public static int getCounter() {
        return counter;
    }

    public void print()
    {
        System.out.print("Name: "+name+"\n"+"Age: "+age+"\n");
    }
}
```

```
public class Main {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);

        System.out.print("Printing person data: ");
        Person yuri=new Person("Yuri");
        yuri.setAge(25);
        yuri.print();

        System.out.print("Printing person data: ");
        Person alexey=new Person("Alexey");
        alexey.setAge(50);
        alexey.print();

        System.out.println("Number of people so far: "+Person.getCounter());

        byte varByte=55;
        int varInt=10;
        long varLong=789;

        double varDouble=1.675;
        float varFloat= 265.36f;

        char varChar='W';

        boolean varBoolean=false;

        String varString="";

        System.out.print("\n\nCurrent value of varByte is: "+varByte);
        System.out.print("\n\nEnter new value for varByte: ");
        varByte=in.nextByte();
        System.out.print("New value of varByte is: "+varByte);

        System.out.print("\n\nCurrent value of varInt is: "+varInt);
        System.out.print("\n\nEnter new value for varInt: ");
        varInt=in.nextInt();
        System.out.print("New value of varInt is: "+varInt);

        System.out.print("\n\nCurrent value of varLong is: "+varLong);
        System.out.print("\n\nEnter new value for varLong: ");
        varLong=in.nextLong();
        System.out.print("New value of varLong is: "+varLong);

        System.out.print("\n\nCurrent value of varDouble is: "+varDouble);
        System.out.print("\n\nEnter new value for varDouble: ");
        varDouble=in.nextDouble();
        System.out.print("New value of varDouble is: "+varDouble);

        System.out.print("\n\nCurrent value of varFloat is: "+varFloat);
        System.out.print("\n\nEnter new value for varFloat: ");
        varFloat=in.nextFloat();
        System.out.print("New value of varFloat is: "+varFloat);

        System.out.print("\n\nCurrent value of varChar is: "+varChar);
        System.out.print("\n\nEnter new value for varChar: ");
        varChar=in.next().charAt(0);
        System.out.print("New value of varChar is: "+varChar);

        System.out.print("\n\nCurrent value of varBoolean is: "+varBoolean);
        System.out.print("\n\nEnter new value for varBoolean: ");
        varBoolean=in.nextBoolean();
        System.out.print("New value of varBoolean is: "+varBoolean);
    }
}
```

```

System.out.print("\n\nCurrent value of varString is: "+varString);
System.out.print("\nEnter new value for varString: ");
varString=in.next();
System.out.print("New value of varString is: "+varString);

System.out.print("\n\nEntering 5 integers\n");
int [] array=new int[5];
for(int i=0;i<array.length;i++) {
    System.out.print("Enter value: ");
    array[i] = in.nextInt();
}

System.out.print("\n\nEnter 5 names\n");
String [] names=new String[5];
int i=0;
while (i<names.length)
{
    System.out.print("Enter name: ");
    names[i]=in.next();
    i++;
}

do {
    System.out.println("square of ["+array[i]+" is "+Math.pow(array[i],2));
    System.out.println("square root of ["+array[i]+" is "+Math.sqrt(array[i]));
}while (i< array.length);

int choice=-1;
while (choice!=4) {
    System.out.print(
        """
                =====Main Menu Example =====
                1- print Hello message
                2- print integer array
                3- print names array
                4- exit
                """);
    choice = in.nextInt();
    switch (choice) {
        case 1: {
            System.out.print(
                """
                Choose language
                1- Russian
                2- English
                """);
            int lang = in.nextInt();
            if (lang == 1) {
                System.out.println("Добро пожаловать!");
            } else if (lang == 2) {
                System.out.println("Welcome!");
            } else {
                System.out.println("invalid input!");
            }

            break;
        }
        case 2:
        {
            for(int k=0;k<array.length;k++)
            {
                System.out.print(array[k]+" ");
            }
        }
    }
}

```


4- угол между двумя векторами

Напишите тесты для проверки корректности вашей программы.