

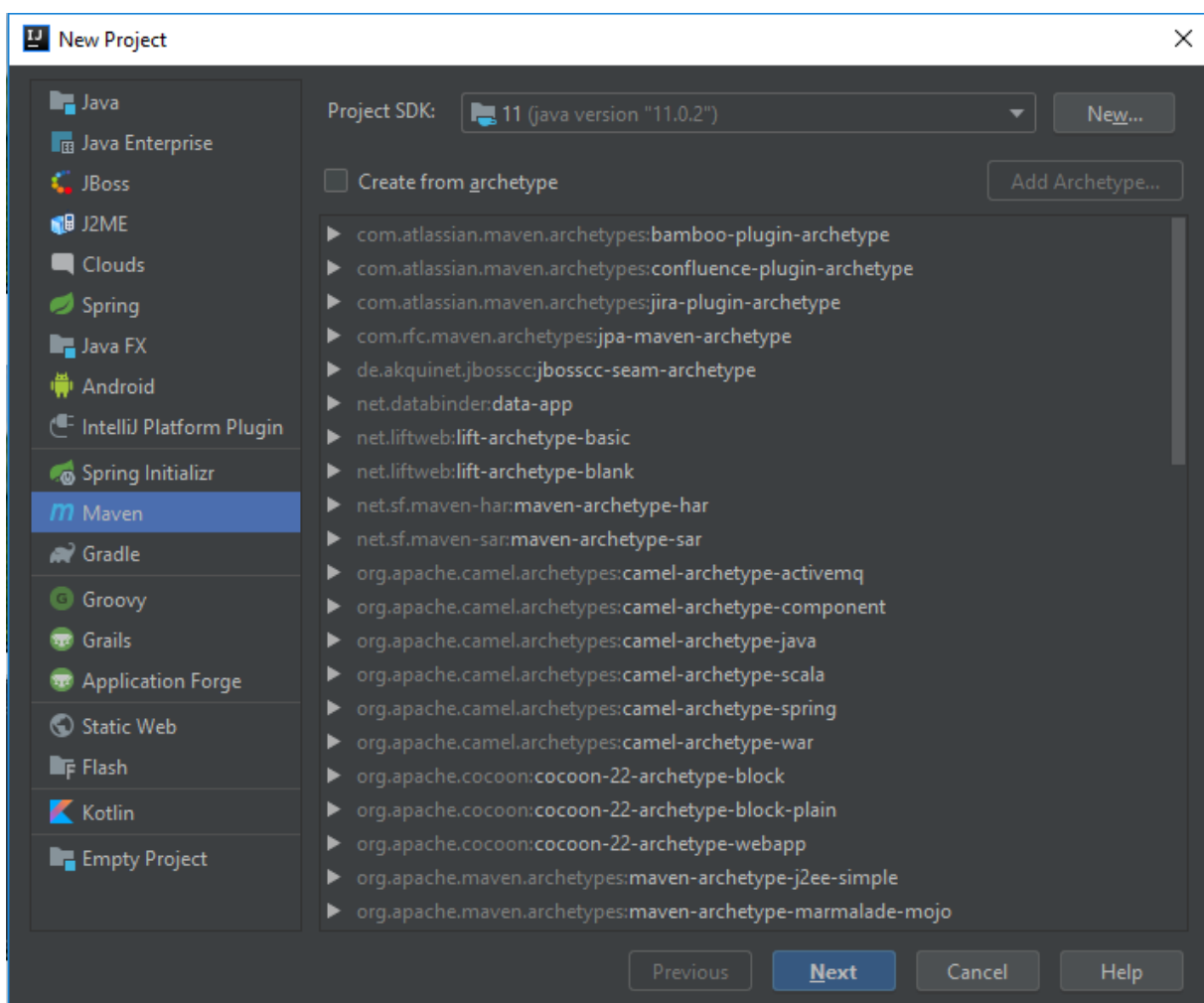
Java – Programowanie Obiektowe – Ćwiczenie 1- wprowadzenie

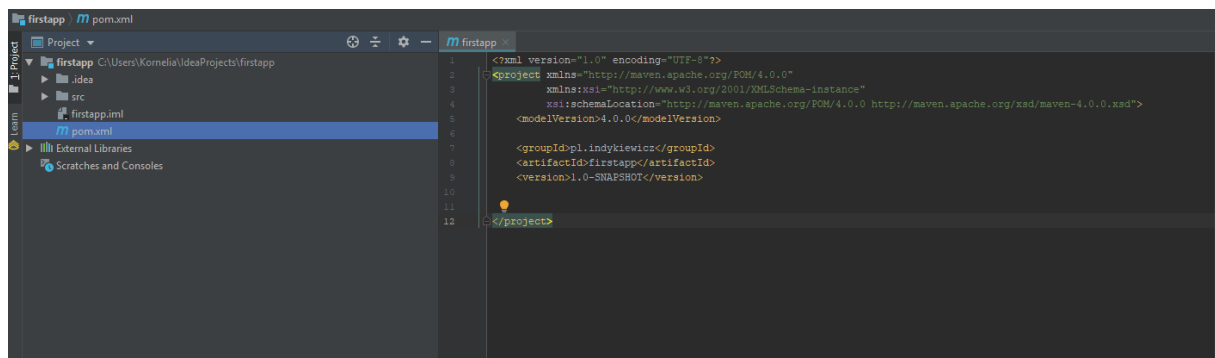
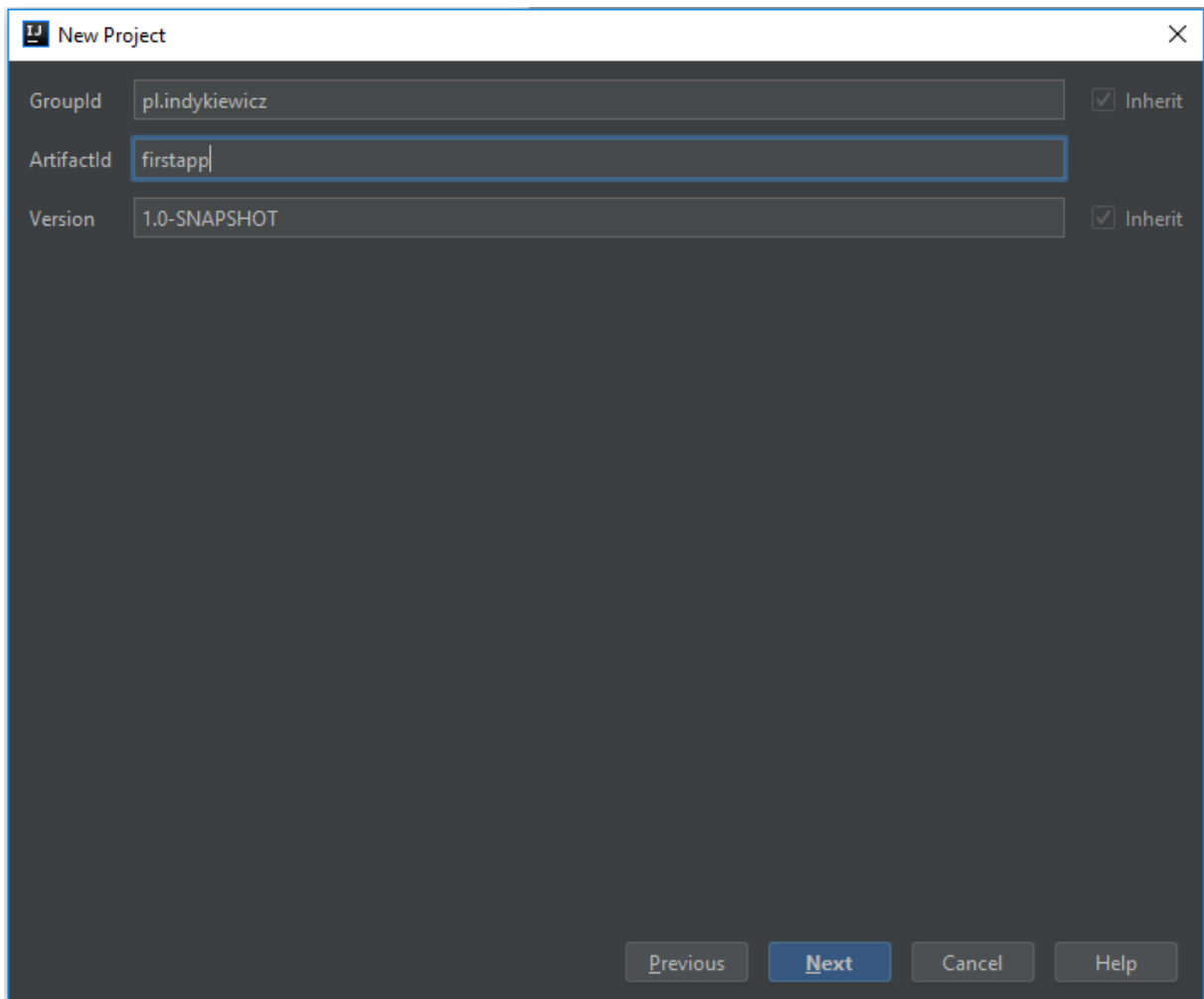
1. Cel ćwiczenia i zakres

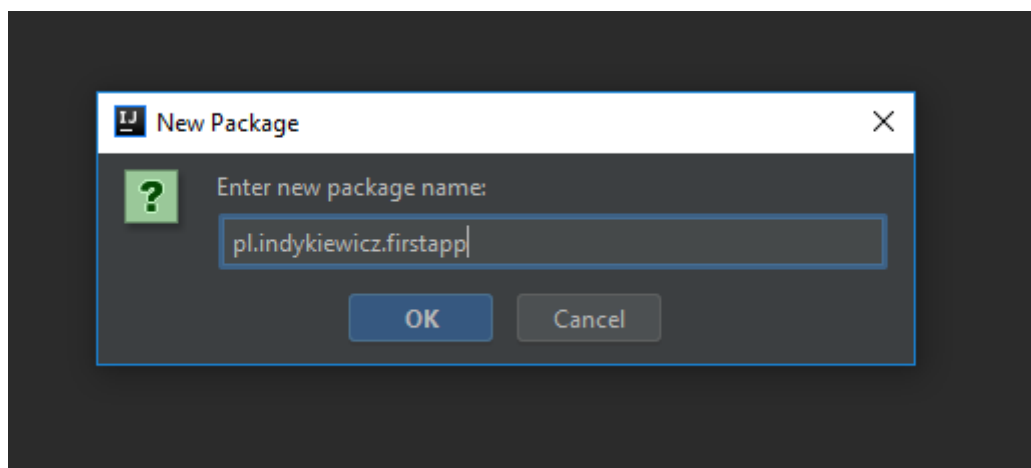
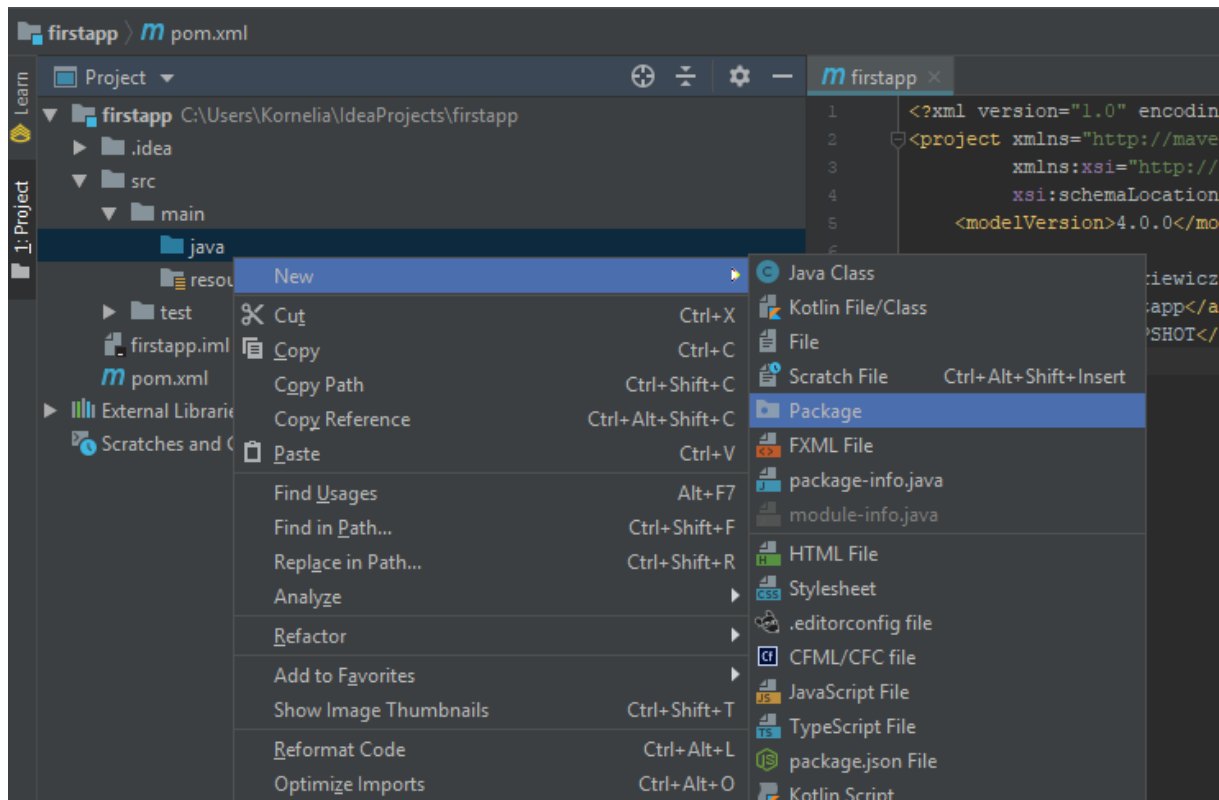
- Poznanie środowiska IntelliJ IDEA
- Maven – narzędzie do budowania projektów
- typy podstawowe i tablice, instrukcje warunkowe, pętle
- klasy, metody, obiekty

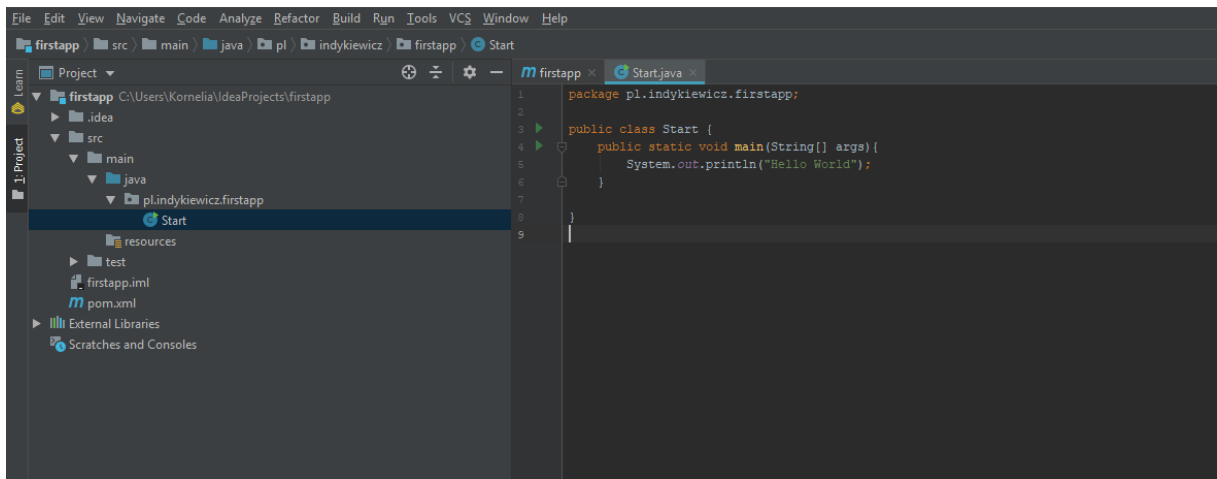
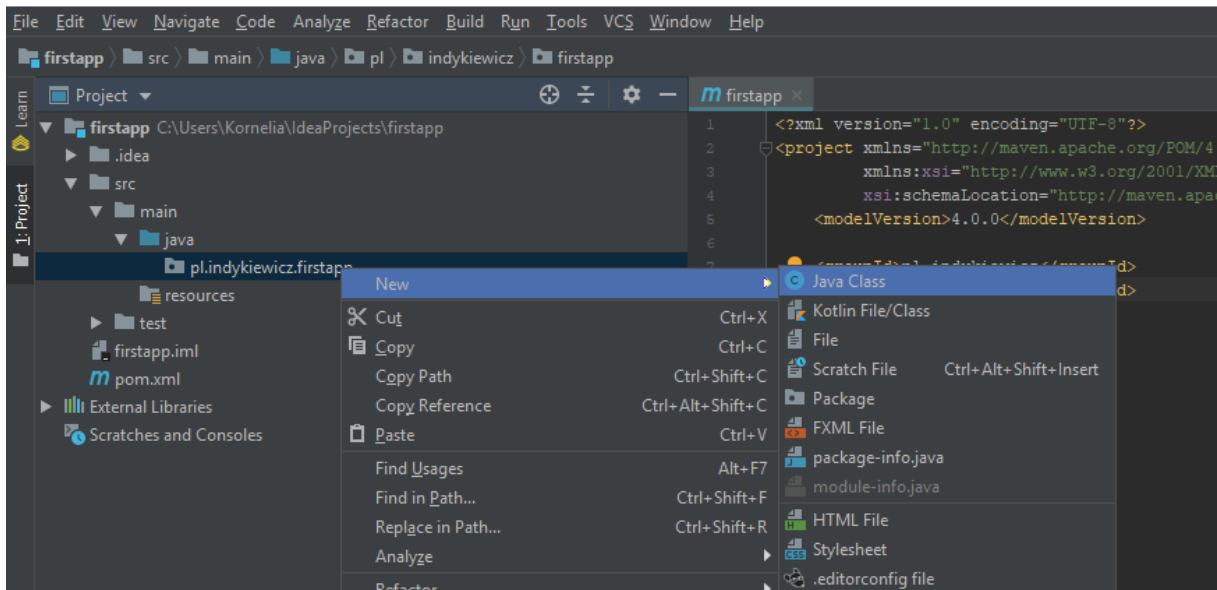
2. Zagadnienia

- Stworzenie projektu



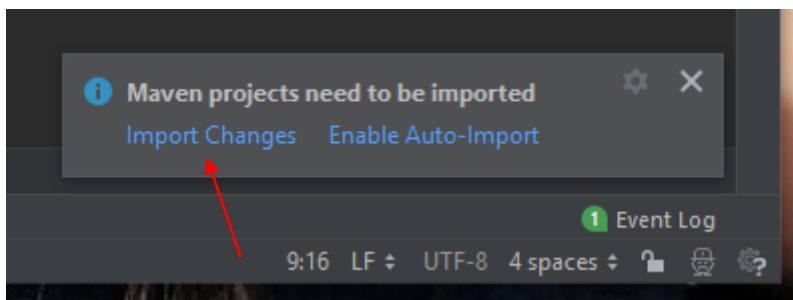






//Dodanie plugina do pom.xml

```
<build>
  <plugins>
    <plugin>
      <artifactId>maven-compiler-plugin</artifactId>
      <version>3.8.0</version>
      <configuration>
        <source>11</source>
        <target>11</target>
      </configuration>
    </plugin>
  </plugins>
</build>
```



- Uruchomienie pierwszego programu – Witaj Świecie!
 - src/main/java + pakiety + jedna publiczna klasa na plik (uwaga! ta sama nazwa co plik)

```
public class Start {
  public static void main(String[] args){
    System.out.println("Hello World");
  }
}
```

Maven: Compile -> install a potem już tylko Run



- Zmienne i typy danych

Typ zmiennych	Domyślna wartość	Domyślna wielkość
byte	0	1 byte
short	0	2 bytes
int	0	4 bytes
long	0L	8 bytes
float	0.0f	4 bytes
double	0.0d	8 bytes
boolean	false	1 bit
char	'\u0000'	2 bytes

```
int first = 5;
int second = 7;

System.out.println(first+ second);
System.out.println(2 + second - first - second);
```

```
class DataTypes {
    public static void main(String[] args) {
        System.out.println('c' + 'C');
        System.out.println(3 + 5);
        System.out.println(1.0 + 2.0);
        System.out.println("apostrof `");
        System.out.println(true);
    }
}
```

- Tablice

```
int[] tablica = new int[10];
```

lub

```
int tablica[] = new int[10];
```

- String

```
String hi = "Hej ";
String name = "Jacek";
String bye = ", and pa!";
String sentence = hi + name + bye;
System.out.println(sentence);
```

```
.length()
.contains()
```

- **ArrayList**

```
import java.util.ArrayList;

public class ListProgram {
    public static void main(String[] args) {
        ArrayList<String> wordList = new ArrayList<String>();
        wordList.add("First");
        wordList.add("Second");
    }
}
```

.add()
.remove()

- **Instrukcje warunkowe: if – else, if else if**

```
int number = 4;

if (number > 5) {
    System.out.println("Your number is greater than five!");
} else {
    System.out.println("Your number is equal to or less than five!");
}
```

lub

```
int number = 4;
String result = "";

if (number > 5) {
    result = "Your number is greater than five!";
} else {
    result = "Your number is equal to or less than five!";
}

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

```
int number = 3;

if (number == 1) {
    System.out.println("The number is one.");
} else if (number == 2) {
    System.out.println("The number is two.");
} else if (number == 3) {
    System.out.println("The number is three!");
} else {
    System.out.println("Quite a lot!");
}
```

- **Pętle: while, do while, for**

```
int number = 0;

while (number < 11) {
    System.out.println(number);
    number++;
}
```

```
int number = 0;

do{
    System.out.println(number);
    number++;
}
while(number < 11);
```

```
for (int number = 0; number < 11; number++){
    System.out.println(number);
}
```

- **Podstawowe wejście**

```
import java.util.Scanner;

public class Start {
    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        System.out.println("Jak masz na imię?");
        String firstName = scan.nextLine();
        System.out.println("Witaj " + firstName);
    }
}
```

.nextLine() wstrzymuje działanie programu do momentu, w którym użytkownik wpisze coś w konsoli i zatwierdzi enterem

.nextInt() - odczytuje kolejną liczbę całkowitą

- **Klasy metody obiekty**

```
//Foo.java

public class Foo {
    public void method() {
        System.out.println("Ta metoda nic nie zwraca");
    }

    public inttwo() {
        return 2;
    }
}
```



```
public int subtract(int a, int b) {
    return a - b;
}
```

```
// Start.java

public class Start {

    public static void main(String[] args) {
        Foo tab = new Foo();
        int x = tab.two();
        System.out.println(x);
        tab.method ();
        int y = tab.subtract(2, 4);
        System.out.println(y);
    }
}
```

3. Zadania do wykonania

- Napisz program, który zwróci prawdę, jeśli jedna z dwóch temperatur jest mniejsza niż 100
- Napisz program, który zwróci prawdę, jeśli sekwencja liczb 1, 2, 3 pojawi się gdzieś w tablicy liczb całkowitych,
- Napisz program, który sprawdza dla trzech boków trójkąta a, b i c wprowadzonych z klawiatury, czy tworzą one trójkąt prostokątny (zakładamy, że $a > 0$, $b > 0$, $c > 0$).
- Użyj pętli do wyświetlenia następującego wzoru dla zadanej liczby wierszy

```
*
**
***
****
*****
```

- Napisz program, który zwróci połowę ciągu znaków ze Stringa
- Napisz program, który zwróci nową tablicę z elementami w odwrotnej kolejności, np. {1, 2, 3} stanie się {3, 2, 1}.
- Napisz klasę Pracownik, która przechowuje trzy pola:
 - Imię
 - Nazwisko
 - Wiek

Następnie utwórz klasę Firma, w której wykorzystasz klasę pracownik do utworzenia dwóch obiektów przechowujących dane pracowników (wymyśl sobie jakieś) i później wyświetlasz je na ekran.