

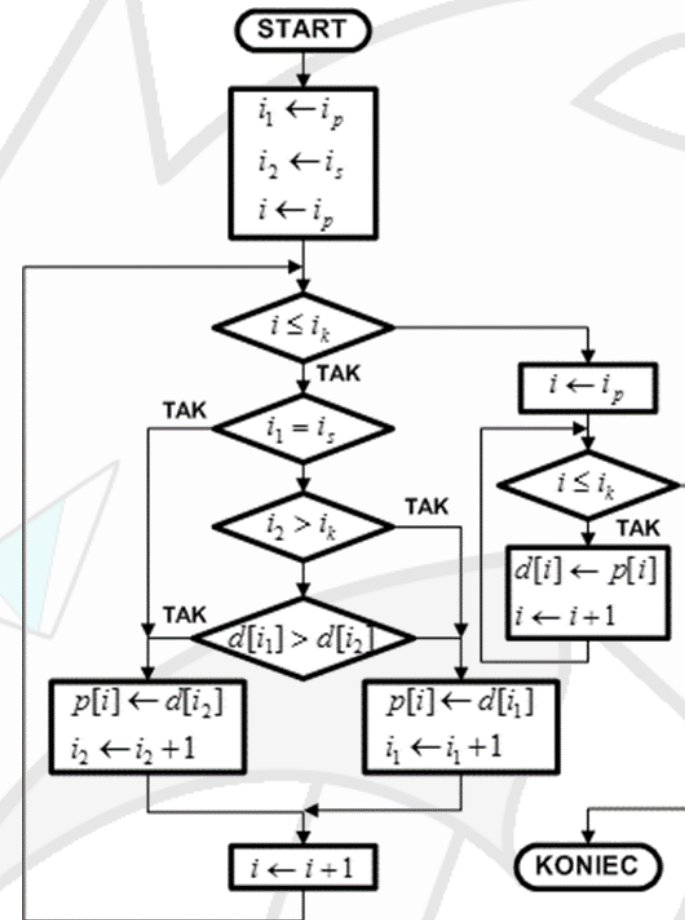
6. Lekcja

Programowanie strukturalne

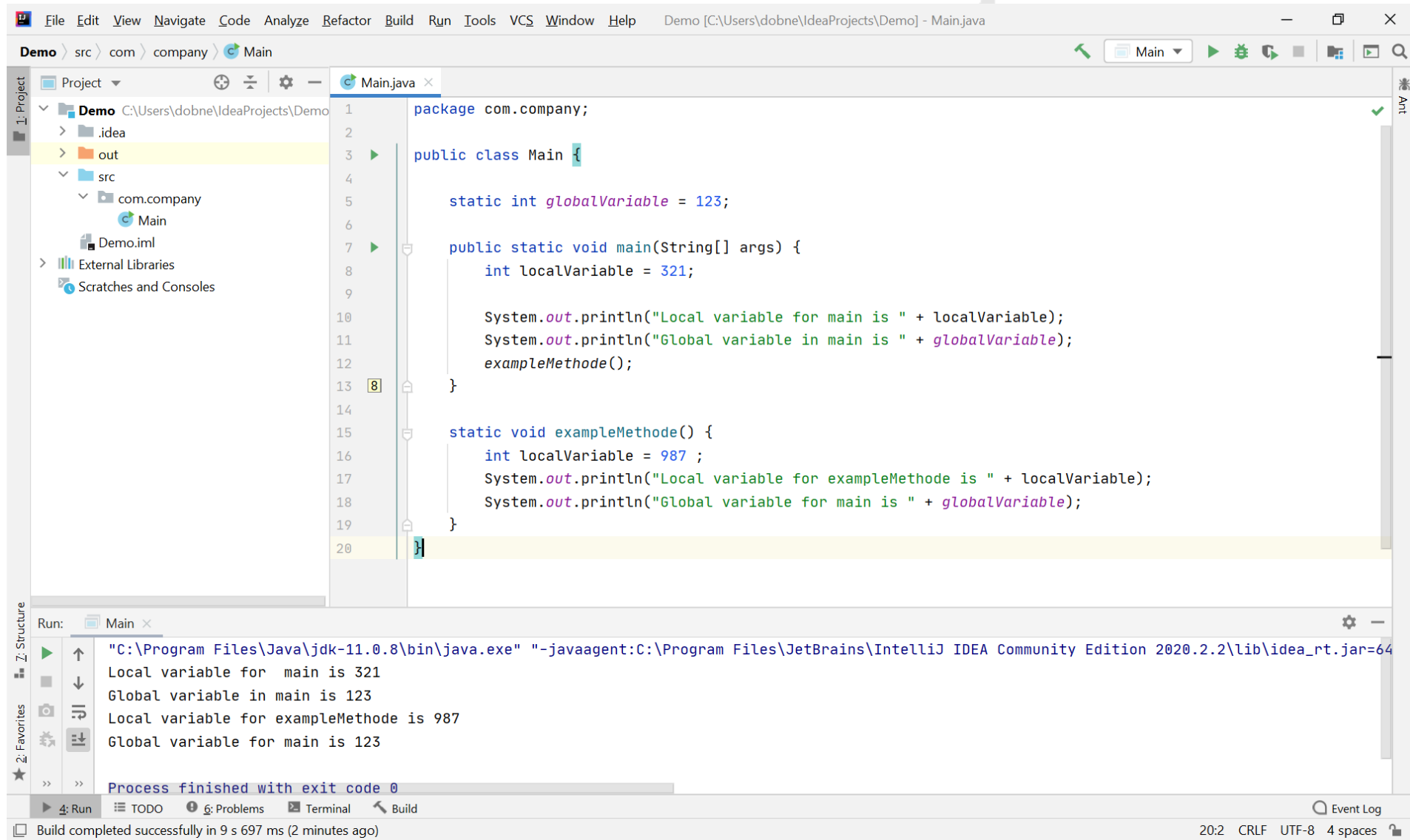


Czyli właściwie co to jest ?

- **programowania** opierający się na podziale kodu źródłowego programu na procedury i hierarchicznie ułożone bloki z wykorzystaniem struktur kontrolnych w postaci instrukcji wyboru i pętli.



Zmienne lokalne i globalne



The screenshot displays the IntelliJ IDEA IDE interface. The main editor window shows a Java file named `Main.java` with the following code:

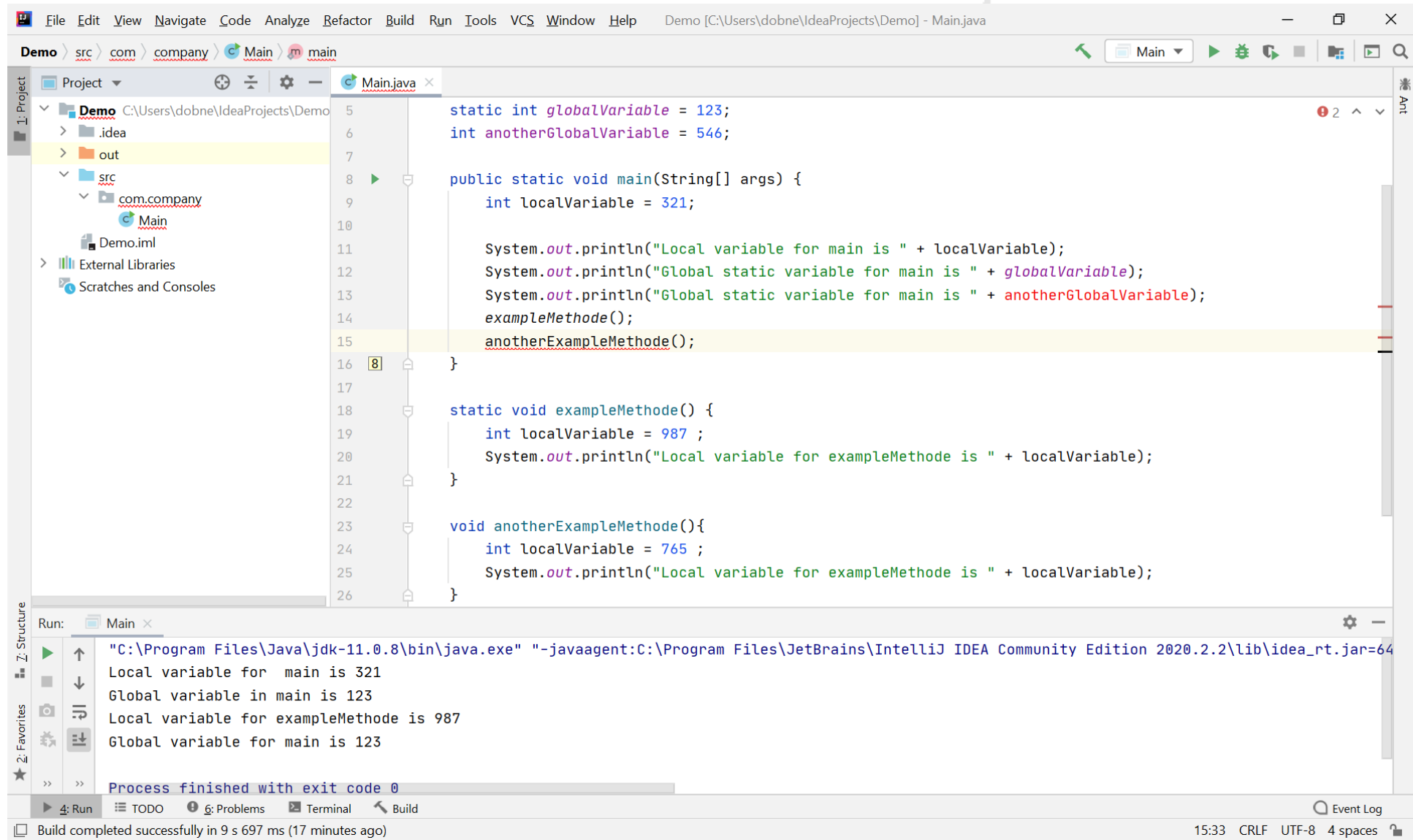
```
1 package com.company;
2
3 public class Main {
4
5     static int globalVariable = 123;
6
7     public static void main(String[] args) {
8         int localVariable = 321;
9
10        System.out.println("Local variable for main is " + localVariable);
11        System.out.println("Global variable in main is " + globalVariable);
12        exampleMethod();
13    }
14
15    static void exampleMethod() {
16        int localVariable = 987;
17        System.out.println("Local variable for exampleMethod is " + localVariable);
18        System.out.println("Global variable for main is " + globalVariable);
19    }
20 }
```

The `main` method is currently selected. Below the editor, the `Run` tab is active, showing the execution output for the `Main` class:

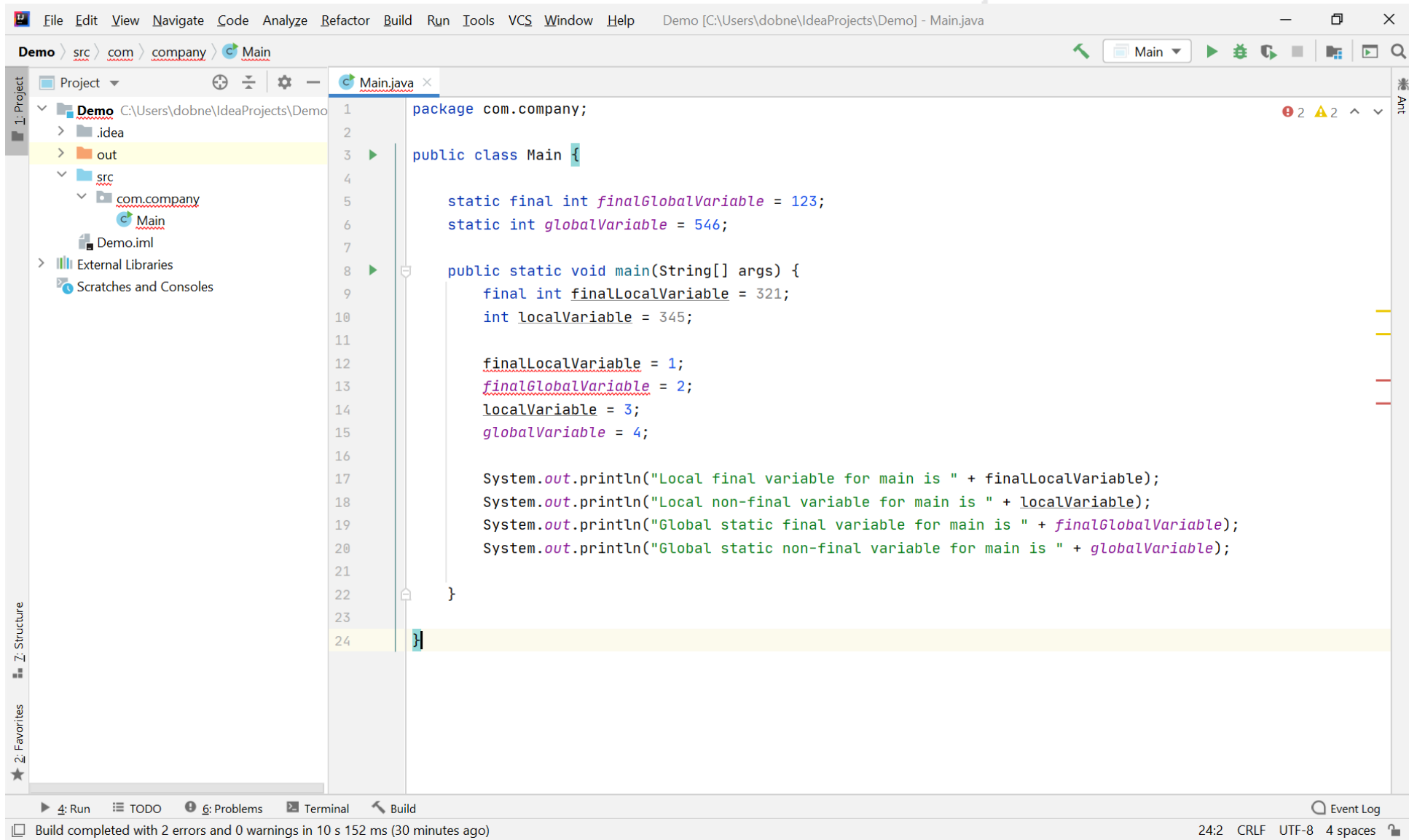
```
Run: Main
"C:\Program Files\Java\jdk-11.0.8\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2020.2.2\lib\idea_rt.jar=64
Local variable for main is 321
Global variable in main is 123
Local variable for exampleMethod is 987
Global variable for main is 123
Process finished with exit code 0
```

The status bar at the bottom indicates that the build completed successfully in 9 s 697 ms (2 minutes ago) and shows the current time as 20:2, along with encoding settings (CRLF, UTF-8, 4 spaces).

static



final



Metody i funkcje

Algorytmiczny przepis na jajecznicę

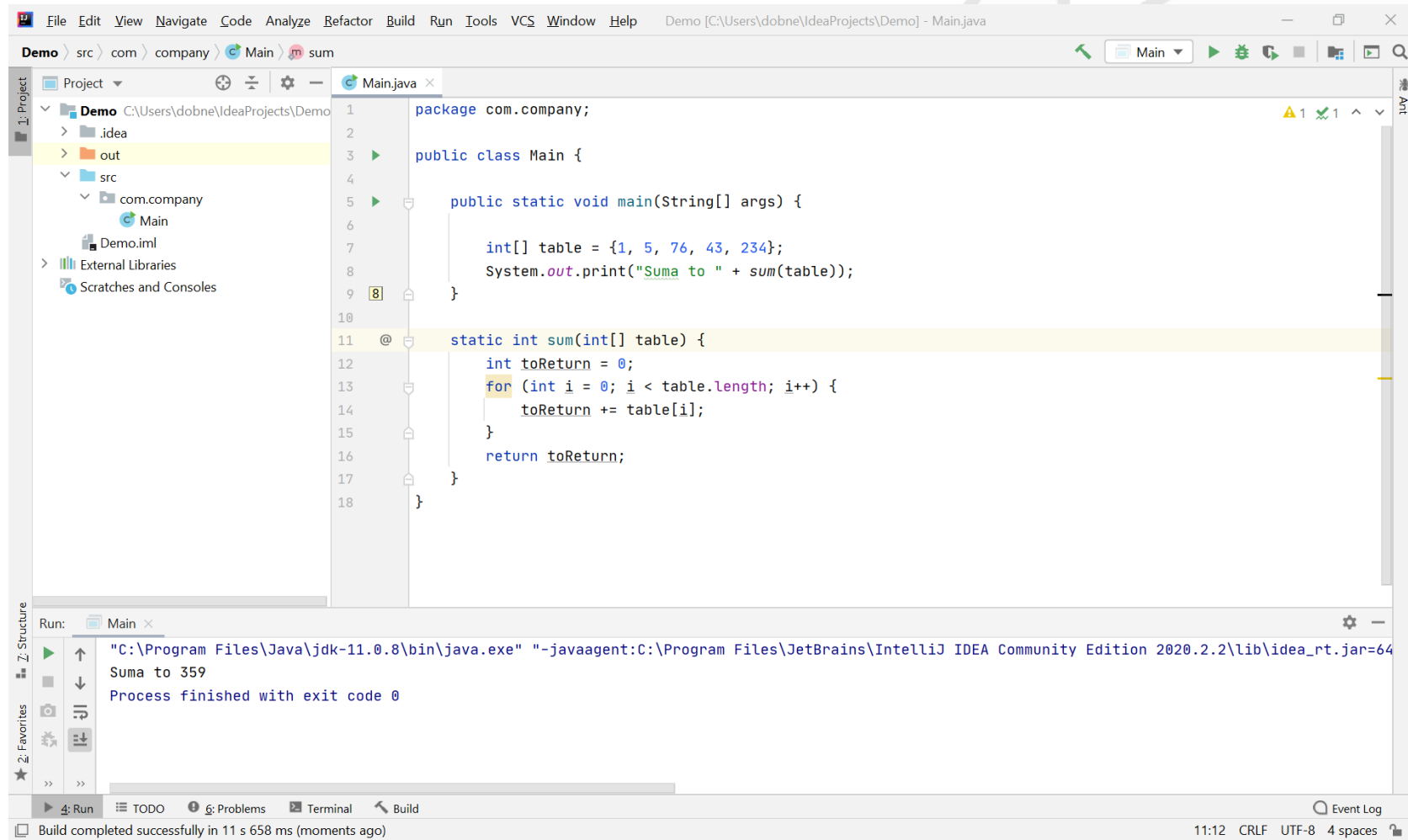


- Funkcja (czasami nazywana podprogramem, rzadziej procedurą) to wydzielona część programu, która przetwarza argumenty i ewentualnie zwraca wartość, która następnie może być wykorzystana, jako argument w innych działaniach lub funkcjach.

- Metoda - podprogram składowy klasy, którego zadaniem jest działanie na rzecz określonych elementów danej klasy lub klas z nią spokrewnionych.

Metoda składnia

typ_zwracany nazwa_metody (lista_argumentów) { operacje }



Argumenty i wartość zwracana metody

- Słowo „void” dla metody oznacza że nie zwraca ona wartości
- Typ przed nazwą metody oznacza typ jaki ma zwracać
- Metody mogą przyjmować dowolną ilość argumentów dowolnego typu lub listę argumentów jednego typu

```
void withoutParameterAndReturn(){  
    System.out.println("withoutParameterAndReturn");  
}  
  
int withoutParameterReturnValue(){  
    return 123;  
}  
  
int oneParameterReturnValue(int parameter1){  
    return parameter1;  
}  
  
int twoParametersReturnValue(int parameter1, int parameter2){  
    return parameter1+parameter2;  
}  
  
int manyParametersReturnValue(int... parameters){  
    return parameters[0]+parameters[parameters.length-1];  
}
```


Metody przeciążone

To metody które posiadają tą samą nazwę ale posiadają inną listę parametrów. Przez podanie odpowiednich parametrów kompilator sam wie której metody użyć.

Nie jest dozwolona ta sama nazwa metody przy różnym typie zwracanym.

```
static void wait(int seconds) {  
    System.out.println("I waiting by " + seconds + " seconds");  
}
```

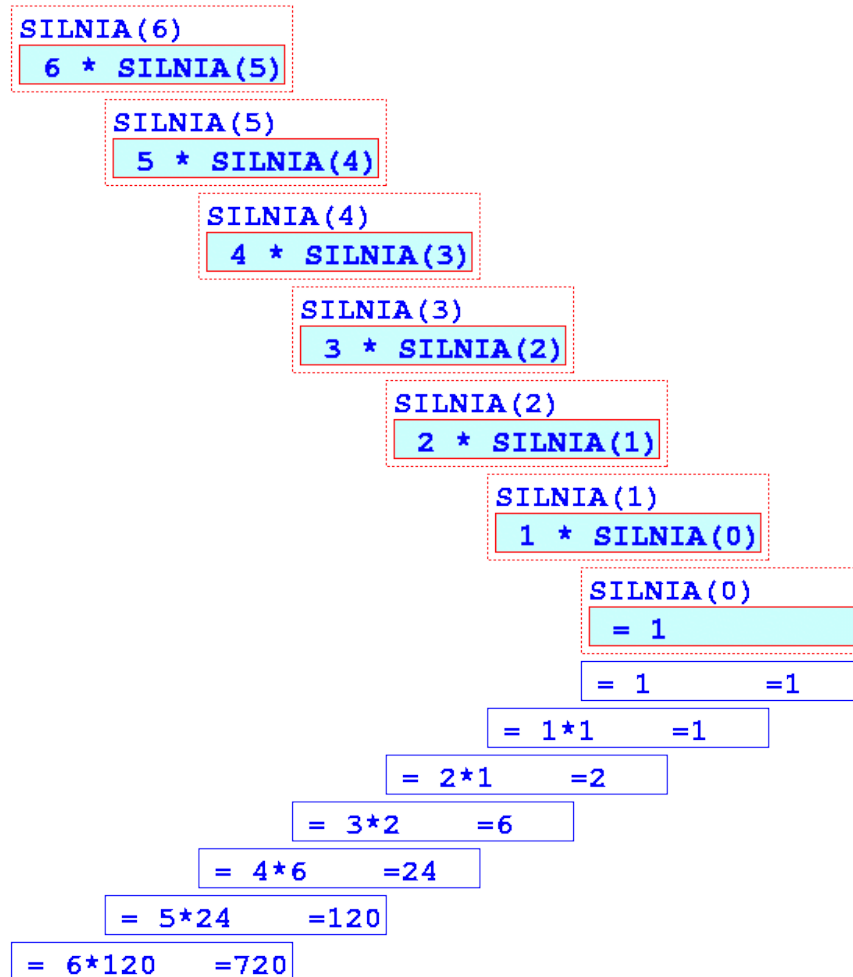
```
static void wait(double seconds) {  
    System.out.println("I waiting by " + seconds + " seconds");  
}
```

```
static void getAge() {  
    System.out.println("I'm 28 years old");  
}
```

```
static int getAge() {  
    return 28;  
}
```

Rekurencja

- polega na wywołaniu przez funkcję samej siebie. Algorytmy rekurencyjne zastępują w pewnym sensie iteracje.



```
1 package com.company;
2
3 public class Main {
4
5
6     public static void main(String[] args) {
7         recuretion( iteration: 10);
8     }
9
10    static void recuretion(int iteration) {
11        if (iteration != 0) {
12            System.out.println("Recurention " + iteration);
13            recuretion(--iteration);
14        }
15    }
16 }
```

Run: Main

"C:\Program Files\Java\jdk-11.0.8\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2020.2.2\lib\idea_rt.jar=65

Recurention 10

Recurention 9

Recurention 8

Recurention 7

Recurention 6

Recurention 5

Recurention 4

Recurention 3

Recurention 2

Recurention 1

Process finished with exit code 0

Zrozumiałe?

