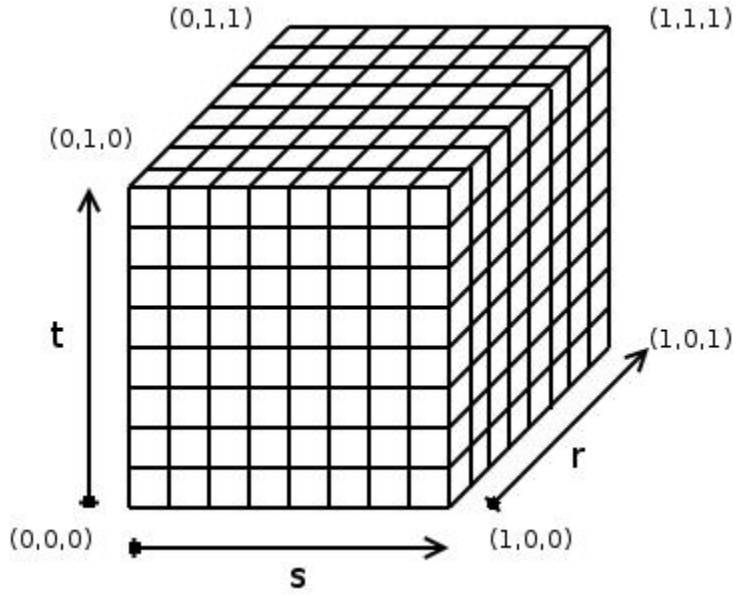
5. Lekcja

Tablice n-wymiarowe i pętle

Tablice n-wymiarowe

Tablica trójwymiarowa



Tablica jednowymiarowa

Tablica dwuwymiarowa

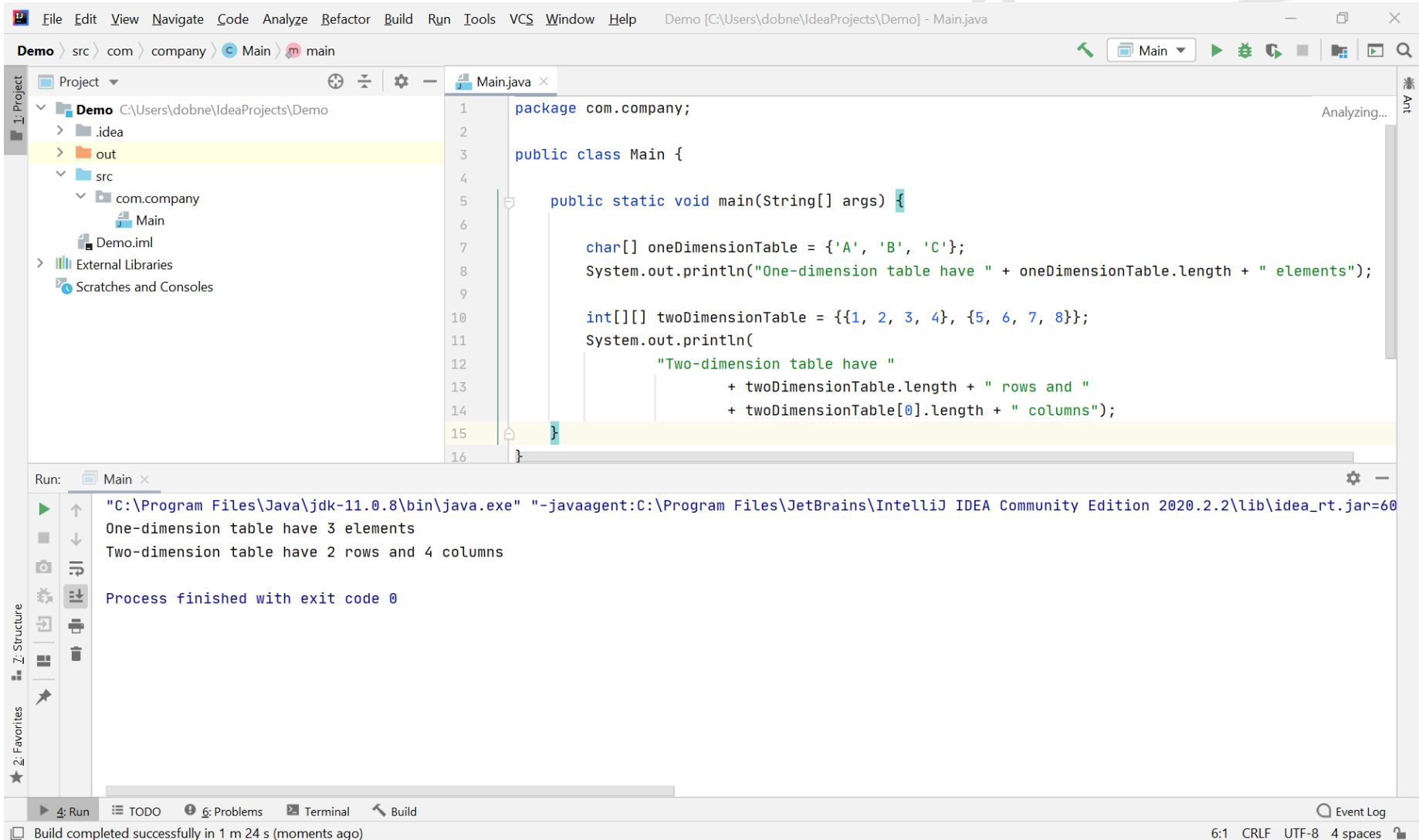
[0][0]	[0][1]	[0][2]	[0][3]	[0][4]
[1][0]	[1][1]	[1][2]	[1][3]	[1][4]
[2][0]	[2][1]	[2][2]	[2][3]	[2][4]
[3][0]	[3][1]	[3][2]	[3][3]	[3][4]

Inicjalizacja tablic n-wymiarowych

```java
//One dimensional table
int[] oneDimensional = new int[5];

//Two dimensional table
char[][] twoDimensional = new char[5][7];

//Three dimensional table
boolean[][][] threeDimensional = new boolean[2][3][4];
```

```java
int[] oneDimensional = {1, 2, 3, 4, 5};

char[][] twoDimensional = {
        {'A', 'B', 'C'},
        {'a', 'b', 'c'},
        {'1', '2', '3'}
};

boolean[][][] threeDimensional = {
        {
                {true, false},
                {false, true}
        },
        {
                {true, false},
                {false, true}
        }
};
```

Sprawdzanie długości tablicy



Pętla for

for (zmienna_startowa; warunek_wykonywalności; modyfikacja_zmiennej) { operacje }

Zalety:

- Możliwość wykorzystania iteratora do obliczeń.
- Idealna do iteracji po tablicach n-wymiarowych

Wady:

- Skomplikowana budowa

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help Demo [C:\Users\dobne\IdeaProjects\Demo] - Main.java

Demo > src > com > company > Main > main

Main

Project

Demo C:\Users\dobne\IdeaProjects\Demo
.idea
out
src
com.company
Main
Demo.iml
External Libraries
Scratches and Consoles

Main.java

```
1  package com.company;
2
3  public class Main {
4
5      public static void main(String[] args) {
6
7          for (int i = 0; i < 10; i++) {
8              System.out.println(i + " number of loop");
9          }
10
11     }
12 }
13
```

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=56
0 number of loop
1 number of loop
2 number of loop
3 number of loop
4 number of loop
5 number of loop
6 number of loop
7 number of loop
8 number of loop
9 number of loop

Process finished with exit code 0
```

4: Run TODO 6: Problems Terminal Build Event Log

Build completed successfully in 8 s 144 ms (moments ago) 10:1 CRLF UTF-8 4 spaces

Pętla foreach

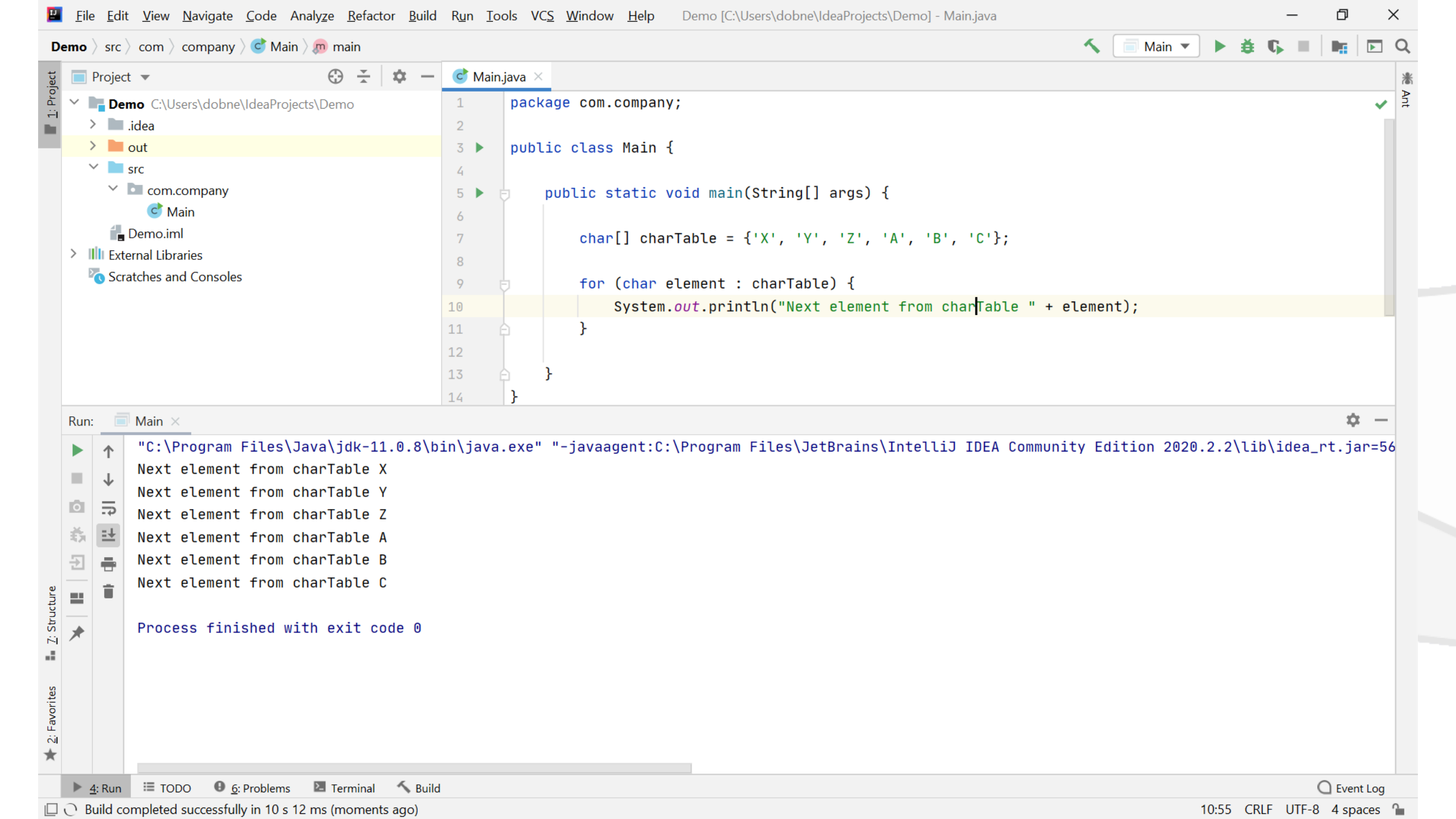
for (określenie_zmiennej: kolekcja) { operacje }

Zalety:

- Idealna do operacji na tablicach jednowymiarowych
- Łatwa budowa

Wady:

- Brak wbudowanego iteratora


File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
Demo [C:\Users\dobne\IdeaProjects\Demo] - Main.java
Demo src com company Main main
Main
Project
Demo C:\Users\dobne\IdeaProjects\Demo
.idea
out
src
com.company
Main
Demo.iml
External Libraries
Scratches and Consoles
Main.java
1 package com.company;
2
3 public class Main {
4
5 public static void main(String[] args) {
6
7 char[] charTable = {'X', 'Y', 'Z', 'A', 'B', 'C'};
8
9 for (char element : charTable) {
10 System.out.println("Next element from charTable " + element);
11 }
12
13 }
14 }
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=56
Next element from charTable X
Next element from charTable Y
Next element from charTable Z
Next element from charTable A
Next element from charTable B
Next element from charTable C
Process finished with exit code 0
1: Project
2: Favorites
7: Structure
Ant
4: Run
TODO
6: Problems
Terminal
Build
Event Log
Build completed successfully in 10 s 12 ms (moments ago)
10:55 CRLF UTF-8 4 spaces

Pętla while

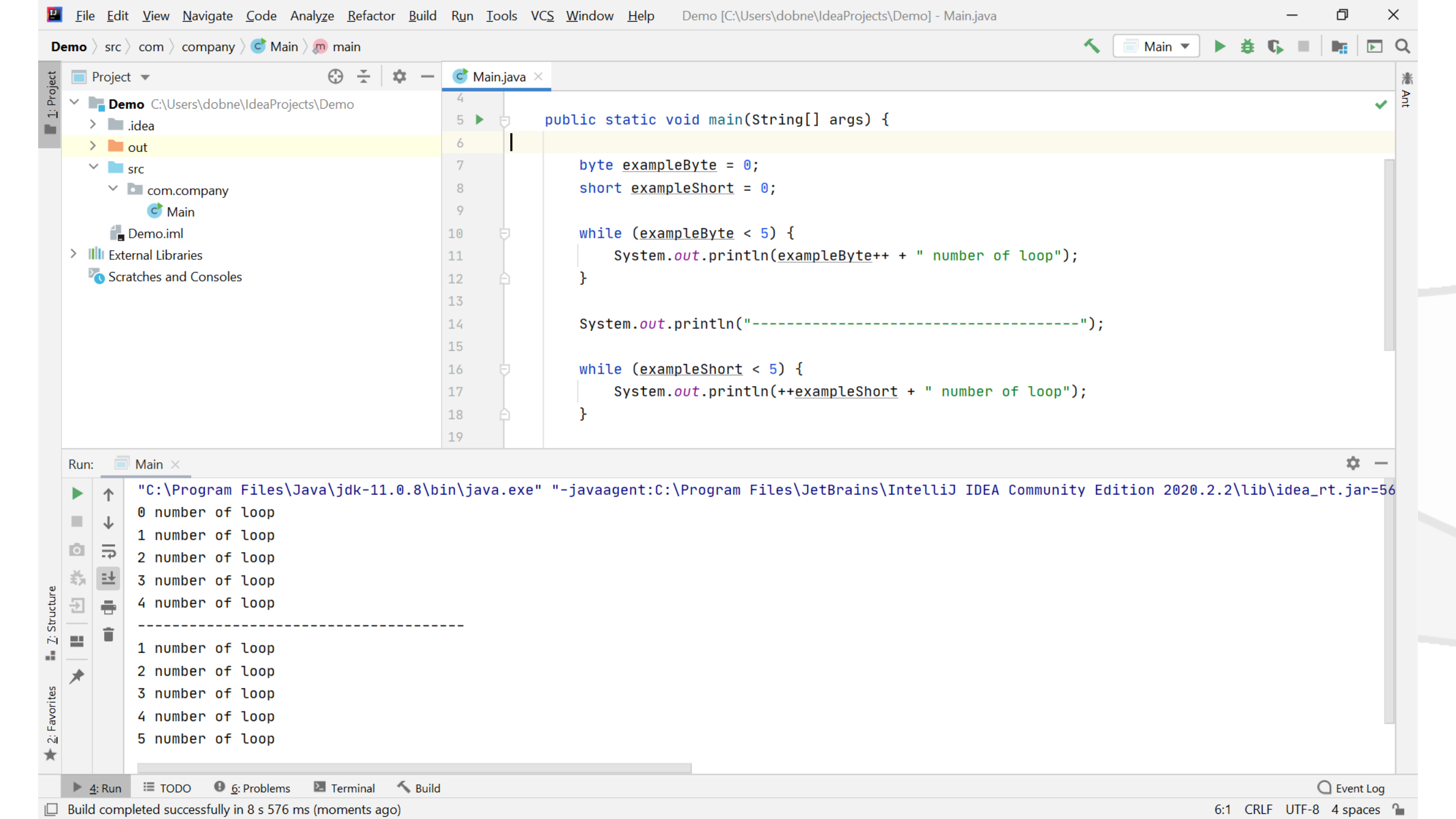
while (warunek_stopu) { operacje }

Zalety:

- Świetnie sprawdza się przy operacjach na plikach
- Łatwa budowa

Wady:

- Możliwość wystąpienia pętli nieskończonej


File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
Demo [C:\Users\dobne\IdeaProjects\Demo] - Main.java
Demo src com company Main main
Main
Project
Demo C:\Users\dobne\IdeaProjects\Demo
.idea
out
src
com.company
Main
Demo.iml
External Libraries
Scratches and Consoles
Main.java
4
5 public static void main(String[] args) {
6
7 byte exampleByte = 0;
8 short exampleShort = 0;
9
10 while (exampleByte < 5) {
11 System.out.println(exampleByte++ + " number of loop");
12 }
13
14 System.out.println("----------");
15
16 while (exampleShort < 5) {
17 System.out.println(++exampleShort + " number of loop");
18 }
19
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=56
0 number of loop
1 number of loop
2 number of loop
3 number of loop
4 number of loop
----
1 number of loop
2 number of loop
3 number of loop
4 number of loop
5 number of loop
1: Project
7: Structure
2: Favorites
Ant
4: Run
TODO
6: Problems
Terminal
Build
Event Log
Build completed successfully in 8 s 576 ms (moments ago)
6:1 CRLF UTF-8 4 spaces

Pętla do while

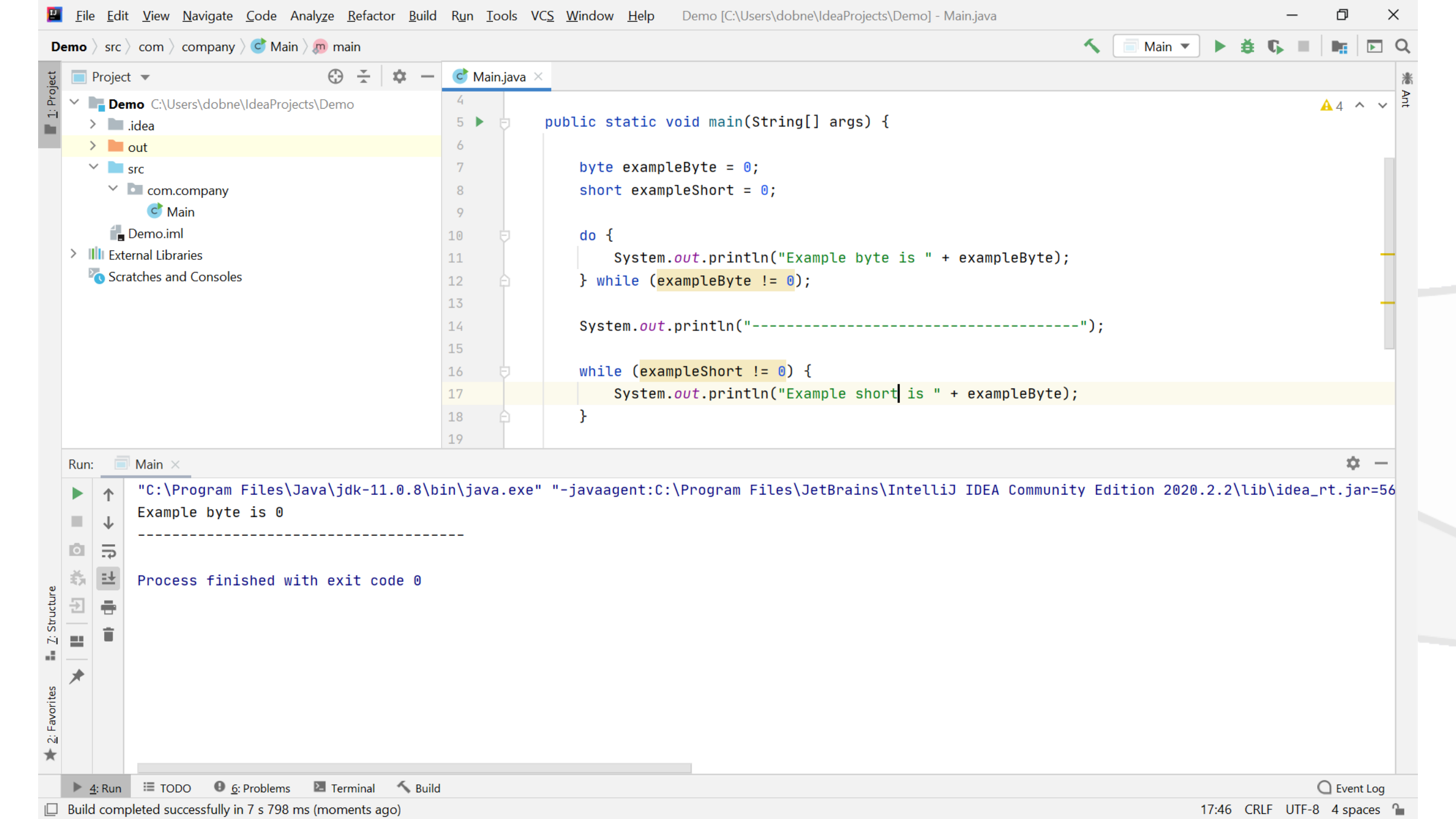
do { operacje } while(warunek_stopu)

Zalety:

- Używana gdy przynajmniej jeden przebieg pętli powinien się wykonać

Wady:

- Możliwość wystąpienia pętli nieskończonej


File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
Demo [C:\Users\dobne\IdeaProjects\Demo] - Main.java
Demo src com company Main main
Main
Project
Demo C:\Users\dobne\IdeaProjects\Demo
.idea
out
src
com.company
Main
Demo.iml
External Libraries
Scratches and Consoles
Main.java
4
public static void main(String[] args) {
byte exampleByte = 0;
short exampleShort = 0;
do {
System.out.println("Example byte is " + exampleByte);
} while (exampleByte != 0);
System.out.println("----------");
while (exampleShort != 0) {
System.out.println("Example short is " + exampleByte);
}
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=56
Example byte is 0

Process finished with exit code 0
4: Run
TODO
6: Problems
Terminal
Build
Event Log
Build completed successfully in 7 s 798 ms (moments ago)
17:46
CRLF
UTF-8
4 spaces

Zadania?