



Kontrola przebiegu programu

Instrukcje warunkowe, pętle, struktury danych

Michał Najwer

Politechnika Wrocławska

Koło Naukowe Kredek

INIT 2021-1



O mnie - Michał Najwer

- Studiuję informatykę stosowaną na W4N, 3 rok
- Członek KN "Kredek"
- Pracuję na Politechnice Wrocławskiej
- Technologie:
 - C#,
 - MSSQL,
 - .NET Core,
 - .NET Framework,
 - Entity Framework,
 - Python, Java, C++



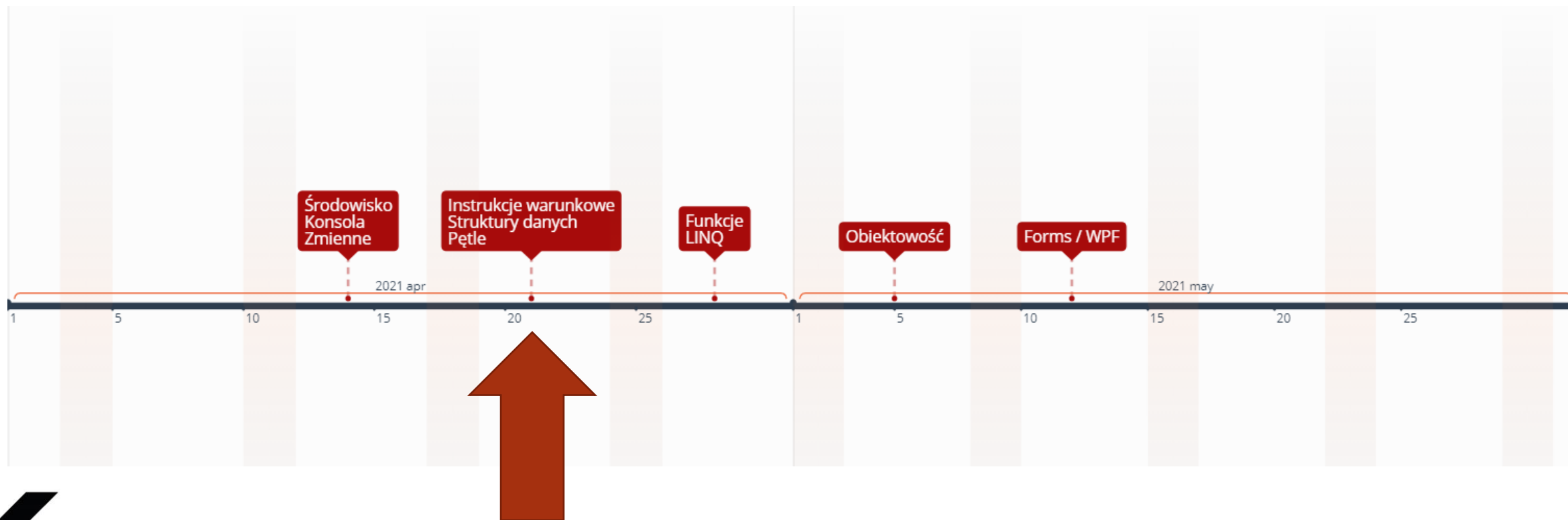
<https://github.com/najwi/>

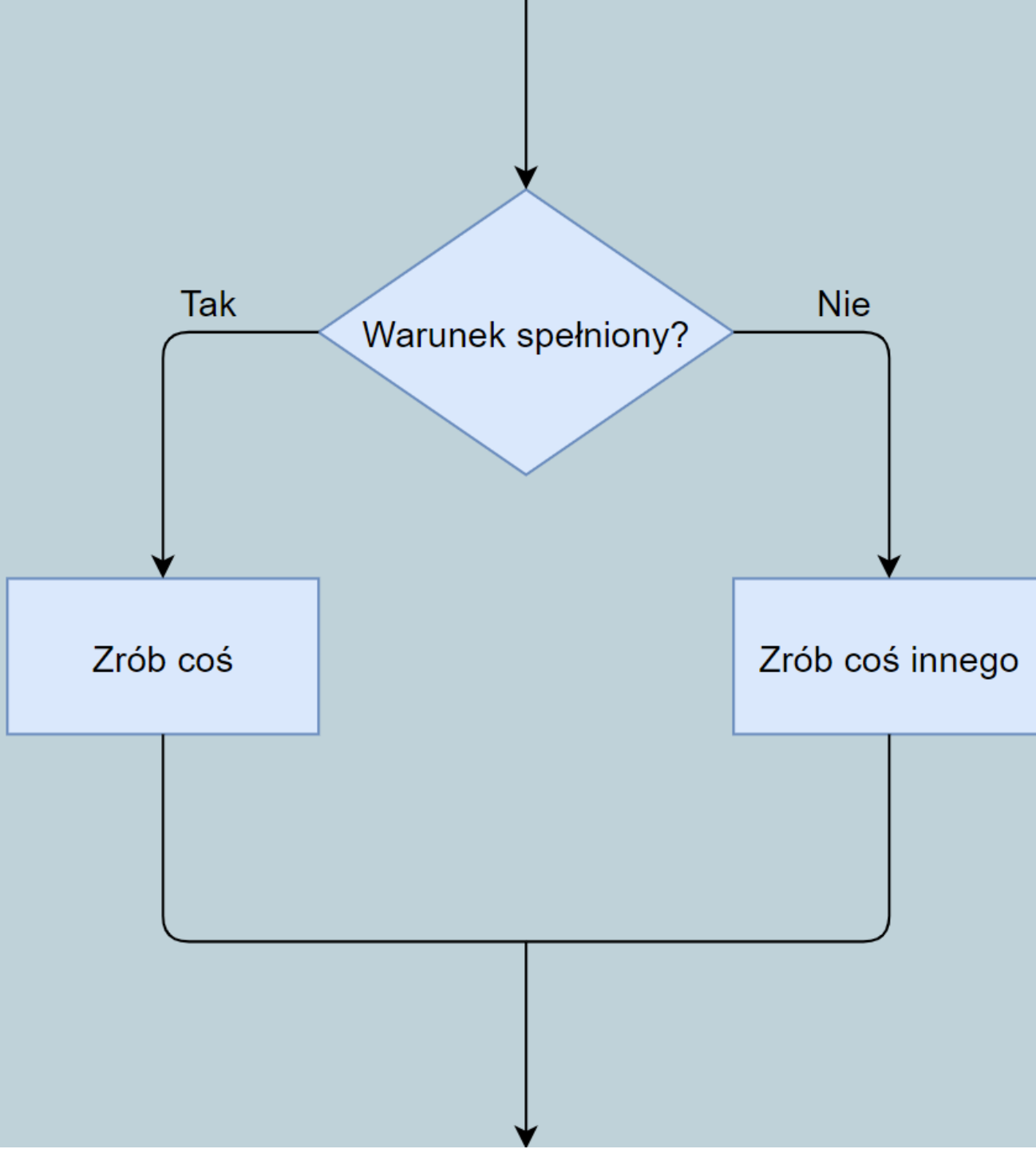


<https://www.linkedin.com/in/michal-najwer/>



Agenda





Instrukcje warunkowe



Definicja

Instrukcja warunkowa – element języka programowania, który pozwala na wykonanie różnych instrukcji w zależności od tego czy zdefiniowane przez programistę wyrażenie logiczne jest prawdziwe, czy fałszywe.



Przykładowy kod

```
1  using System;
2
3  0 references
4  public class Program
5  {
6      0 references
7      static void Main(string[] args)
8      {
9          if(5 > 2)
10             Console.WriteLine("5 jest większe od 2.");
11     }
12 }
13
```

```
1  using System;
2
3  0 references
4  public class Program
5  {
6      0 references
7      static void Main(string[] args)
8      {
9          int x = 1;
10         switch (x)
11         {
12             case 0:
13                 Console.WriteLine("x = 0");
14                 break;
15             case 1:
16                 Console.WriteLine("x = 1");
17                 break;
18         }
19     }
20 }
```

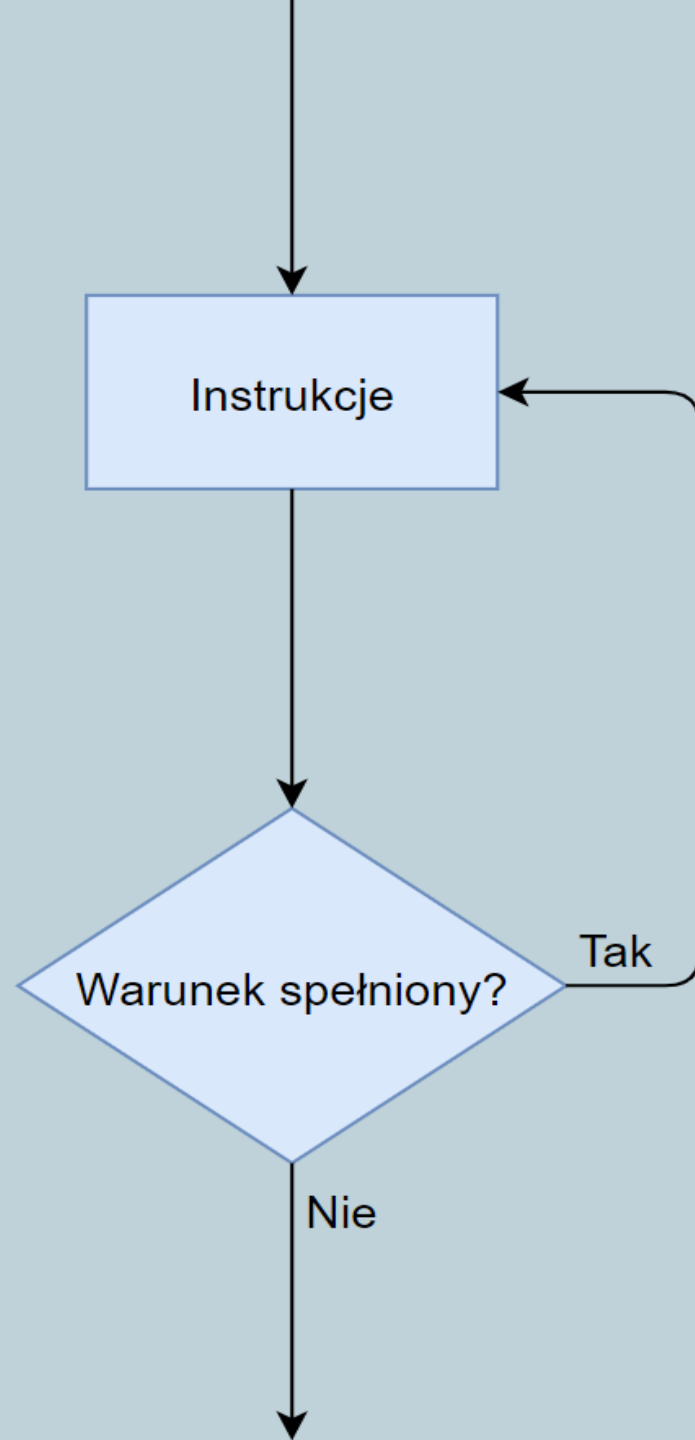


Operatory relacyjne

W poniższej tabeli wszystkie operatory relacyjne wspierane przez język C#. Dla ułatwienia założmy, że $A = 10$ oraz $B = 20$.

Operator	Opis	Przykład
<code>==</code>	sprawdza czy wartości argumentów są równe, jeżeli tak zwraca flagę <code>true</code>	$(A == B)$ <code>false</code> .
<code>!=</code>	sprawdza czy wartości argumentów są równe, jeżeli wartości nie są równe zwraca <code>true</code>	$(A != B)$ <code>true</code>
<code>></code>	sprawdza czy wartość lewego argumentu jest większa od prawego, jeżeli tak zwraca flagę <code>true</code>	$(A > B)$ <code>false</code>
<code><</code>	sprawdza czy wartość lewego argumentu jest mniejsza od prawego, jeżeli tak zwraca flagę <code>true</code>	$(A < B)$ <code>true</code>
<code>>=</code>	sprawdza czy argument po lewej stronie jest większy bądź równy argumentowi po prawej stronie, jeżeli tak zwraca flagę <code>true</code>	$(A >= B)$ <code>false</code>
<code><=</code>	sprawdza czy argument po lewej stronie jest mniejszy bądź równy argumentowi po prawej stronie, jeżeli tak zwraca flagę <code>true</code>	$(A <= B)$ <code>true</code>





Pętle



Definicja

Pętla – jedna z trzech podstawowych konstrukcji programowania strukturalnego. Umożliwia cykliczne wykonywanie ciągu instrukcji określoną liczbę razy, do momentu zajścia pewnych warunków, dla każdego elementu kolekcji lub w nieskończoność.



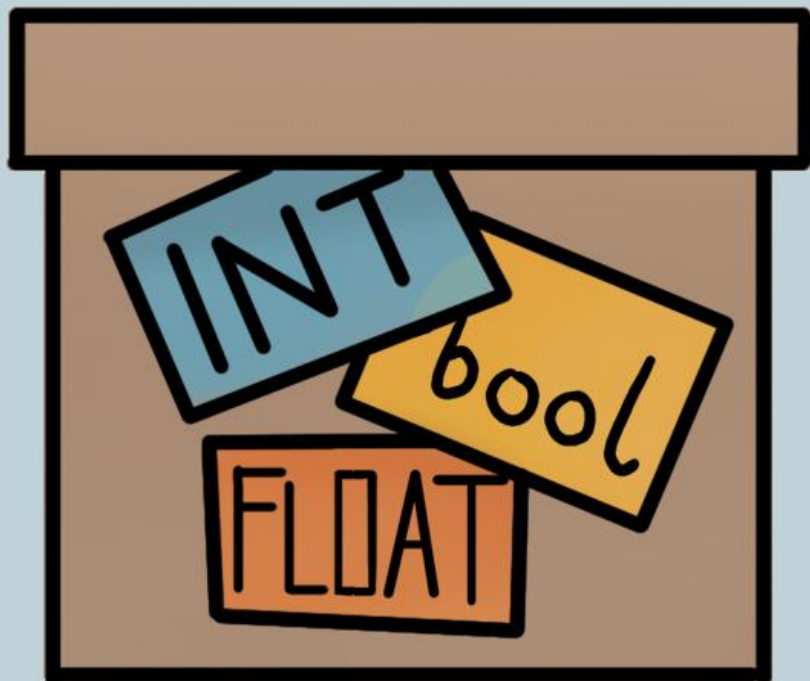
Przykładowy kod

```
1 using System;
2
3 0 references
4 public class Program
5 {
6     0 references
7     static void Main(string[] args)
8     {
9         int x = 5;
10        while (x > 0)
11        {
12            Console.WriteLine(x);
13            x--;
14        }
15    }
16 }
```

```
1 using System;
2
3 0 references
4 public class Program
5 {
6     0 references
7     static void Main(string[] args)
8     {
9         int x = 4;
10        do
11        {
12            Console.WriteLine(x);
13            x--;
14        } while (x > 0);
15    }
16 }
```

```
1 using System;
2
3 0 references
4 public class Program
5 {
6     0 references
7     static void Main(string[] args)
8     {
9         for (int i = 0; i < 5; i++)
10        {
11            Console.WriteLine(i);
12        }
13    }
14 }
```





Struktury danych



Definicja

Struktura danych – sposób przechowywania danych w pamięci komputera. Na strukturach danych operują algorytmy. Podczas implementacji programu programista często staje przed wyborem między różnymi strukturami danych, aby uzyskać pożądaný efekt.



Popularne struktury danych

- Tablica - *object[] array*
- Lista - *List<object> list*
- Kolejka - *Queue<object> queue*
- Stos - *Stack<object> stack*
- Słownik - *Dictionary<object, object> dictionary*





Kontrola przebiegu programu

Instrukcje warunkowe, pętle, struktury danych

Michał Najwer

Politechnika Wrocławska

Koło Naukowe Kredek

INIT 2021-1

