



# Funkcje i LINQ

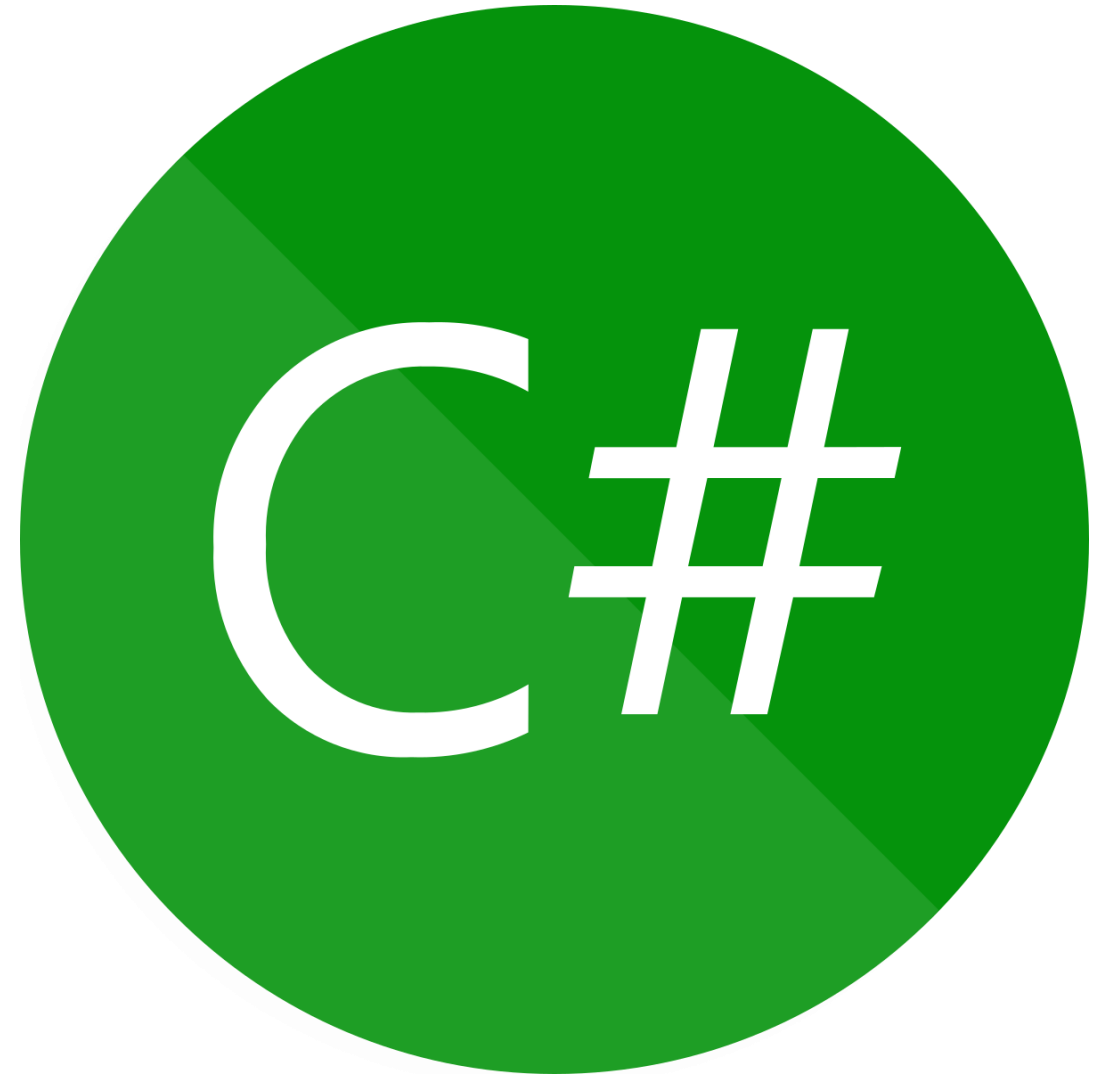
Realizacja podprogramów oraz  
wykonywanie zapytań na obiektach

Maciej Kisielewicz

Politechnika Wrocławska

Koło Naukowe Kredek

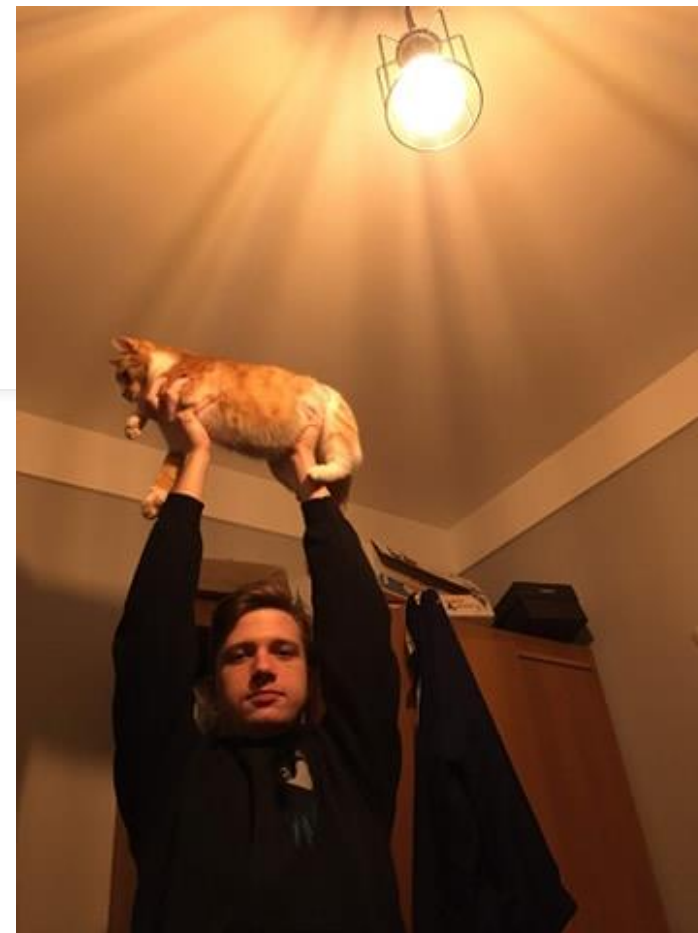
INIT 2021-2



# O mnie

## Maciej Kisielewicz

- Uczestnik CPC 2020-2
- Członek KN Kredek od 2021
- Wiceprezes KN Kredek ds. kursów i warsztatów
- Technologie i umiejętności:
  - C#, .NET Core, .NET Framework, WebApi, MSSQL
  - Python / Pentesting
- Inżynier oprogramowania / inżynier bezpieczeństwa od 2021
- Studiuję cyberbezpieczeństwo na W4 (inż)



<http://maciejkislewicz.pl/>



<https://github.com/Kisielos10>

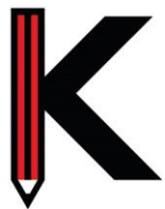
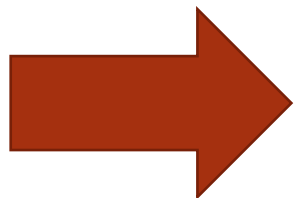


<https://www.linkedin.com/in/maciejkislewicz/>



# Agenda

- 10.11 - Zajęcia wprowadzające
- 17.11 - GIT
- 24.11 - If, pętle, struktury danych
- 01.12 - Funkcje, LINQ
- 08.12 - Programowanie obiektowe
- 22.12 - Windows Forms
- 12.01 - Wzorce projektowe



**FUNCTIONAL**



**PROGRAMMING**

memegenerator.net

**Funkcje w  
języku C#**

# Co to funkcja/metoda?

Metoda jest blokiem kodu, który zawiera serie instrukcji. Program powoduje wykonanie instrukcji przez wywołanie metody.

# Sygnatura funkcji

- Opcjonalny poziom dostępu, taki jak *public* lub *private* . Wartość domyślna to *private*.
- Wartość zwracana lub *void*.
- Nazwa metody.
- Dowolne parametry metody. Parametry metody są ujęte w nawiasy i są rozdzielone przecinkami. Puste nawiasy wskazują, że metoda nie wymaga żadnych parametrów.
- Metody deklarowane są w klasach.



# Przykład

1 reference

```
static int Add(int x, int y)
{
    return x + y;
}
```

# Wywołanie metody

Metody statyczne (czyli takie używane w klasie) możemy wywołać samą jego nazwą i przekazaniem parametru.

```
Congratulations();  
Program.Congratulations();
```





# Rekurencja

Funkcje rekursywne to takie, które wywołują same siebie. Cechują się dużą złożonością obliczeniową ale są bardzo czytelne.



# Przeciążenia

Polega na tworzeniu nowych wersji tych samych metod poprzez zmianę ich sygnatur (lista parametrów, wartość zwracana).

```
public long Add(long a, long b, long c)
{
    long sum = a + b + c;
    return sum;
}

0 references
public double Add(double a, double b, double c)
{
    double sum = a + b + c;
    return sum;
}
```





# Co to jest LINQ?

---





# LINQ

Language-Integrated Query (LINQ) to technologia która pozwala na zapytania bezpośrednio wewnątrz języka C#.



```
static void Main()
{
    // Specify the data source.
    int[] scores = new int[] { 97, 92, 81, 60 };

    // Define the query expression.
    IEnumerable<int> scoreQuery =
        from score in scores
        where score > 80
        select score;

    // Execute the query.
    foreach (int i in scoreQuery)
    {
        Console.Write(i + " ");
    }
}
```

# Właściwości

- Zapytanie nie jest wykonane dopóki nie jest wykonana iteracja zmiennej zapytania np. *foreach*. *Chyba, że wcześniej rzutujemy wynik zapytania na np. listę.*
- Zapytania można wykonywać poprzez ekspresję i systemem metodycznym (funkcyjnym).



# Wyrażenie lambda

Wyrażenie lambda to specjalna metoda używana do tworzenia delegatów w LINQ.

*Parametr ==> kod do wykonania*

*$n ==> n \% 2 == 1$*

# Dodatkowe materiały

<https://github.com/dotnet/try-samples/blob/main/101-linq-samples/readme.md>

<https://linqsamples.com/linq-to-objects>

[https://www.w3schools.com/cs/cs\\_methods.asp](https://www.w3schools.com/cs/cs_methods.asp)

<https://docs.microsoft.com/pl-pl/dotnet/csharp/methods>

