

## **ZASADY DZIAŁANIA PROGRAMU ORAZ INSTRUKCJE, JAK NALEŻY KORZYSTAĆ Z PROGRAMU W CELU PRAWIDŁOWEGO DZIAŁANIA OPERACJI WIELOWĄTKOWOŚCI [ 3 wątki jednocześnie ]**

Niniejsze programy / aplikacje zostały napisane w języku C# .NET oraz demonstrują zasadę działania wielowątkowości [na przykładzie trzech wątków] i wyjaśniają czym jest wielowątkowość.

Niniejsze programy zostały wykonane na dwa różne sposoby, gdzie pierwszy program jest aplikacją konsolową [wszystko wyświetla się w konsoli], natomiast drugą jest aplikacja okienkowa [gdzie prezentowane jest działanie wielowątkowości na przykładzie trzech, wybranych wątków, wykonujących ustalone wcześniej poszczególne działania matematyczne].

W aplikacji konsolowej zostało stworzone 11 metod, z czego 4 metody prezentują działanie wielowątkowości, jedna wyjaśnia od strony teoretycznej, zaś reszta jest dodatkiem wzbogacającym działanie całego programu. W niniejszej aplikacji metody są następujące:

- => 'static void Main(string[] args)',
- => 'public static void JakieJestDzialanieProgramu()',
- => 'public static void DemonstracjaDzialaniaWielowatkowosci()',
- => 'static void WielowatkowoscObliczeniaNr1()',
- => 'static void WielowatkowoscObliczeniaNr2()',
- => 'static void WielowatkowoscObliczeniaNr3()',
- => 'public static void KontynuacjaProgramu()',
- => 'public static void DodatkoweOddzielenieA()',
- => 'public static void DodatkoweOddzielenieB()',
- => 'public static void WprowadzenieDoDefinicjiWielowatkowosci()',
- => 'public static void PrzedstawienieDefinicjiWielowatkowosci()'.

Metoda 'public static void DemonstracjaDzialaniaWielowatkowosci()' jest metodą, która jest wywoływana w metodzie Main oraz jest metodą, gdzie jest inicjowanie trzech wątków w tym samym czasie, czyli wątku nr 1, wątku nr 2 oraz wątku nr 3, które wykonują się w następujących metodach, czyli:

- => 'static void WielowatkowoscObliczeniaNr1()':
  - > niniejszy wątek ma za zadanie wykonanie operacji matematycznej, która będzie polegała na tym, iż jest wykonane 100'000 iteracji pętli for, a następnie dodanie ich do zmiennej wynik\_obliczen
- => 'static void WielowatkowoscObliczeniaNr2()':
  - > niniejszy wątek ma za zadanie wykonanie operacji matematycznej, która będzie polegała na tym, iż jest obliczona silnia z liczby 100'000

=> 'static void WielowatkowoscObliczeniaNr3()':

---> niniejszy wątek ma za zadanie wykonanie operacji matematycznej,  
która będzie polegała na tym, iż są wykonane obliczenia sumy  
liczb parzystych w zakresie od 0 do 1'000'000 z zaznaczeniem,  
iż muszą wszystkie liczby być liczbami dodatnimi

Natomiast metoda 'public static void PrzedstawienieDefinicjiWielowatkowosci()' przedstawia definicję wielowątkowości, wyjaśnia jak wątki są definiowane, czym są wątki, jak i również jaka jest zaleta wielowątkowości [czyli wykonania kilku operacji w tym samym czasie, zamiast jednej operacji]. Do przedstawienia wspomnianych wyjaśnień zostały wykorzystane dwa źródła, które zostały wymienione na końcu przy zakończeniu wyjaśnień o wielowątkowości.

Przechodząc do drugiej aplikacji, czyli aplikacji okienkowej to w niej niestety nie zostało wyjaśnione czym jest wielowątkowość od strony teoretycznej, ale za to zaprezentowano działanie wielowątkowości z wykorzystaniem trzech wątków, które są niemal identyczne z tymi, które zostały zaprezentowane powyżej, jednak z pewną różnicą, gdzie:

=> pierwszy wątek [czyli iteracje pętli for, a następnie ich dodanie wykonują się 100'000'000 razy]

=> drugi wątek [czyli w tym przypadku zostaje wykonana operacji obliczenia silni z liczby 32]

=> trzeci wątek [obliczenie sumy liczb parzystych wykonuje się w zakresie od 0 do 100'000'000]

Podsumowując, niniejsze aplikacje, zarówno konsolowa, jak i okienkowa umożliwiają nie tylko zaobserwowanie działania wielowątkowości, ale i również jej zrozumienie od strony teoretycznej.

Opis działania programów, aplikacji [konsolowej i okienkowej] w C# opisał Łukasz W. Tworzydło.

Dane studenta:

=> Imię i nazwisko: Łukasz Tworzydło

=> Numer albumu: gd29623

=> Nr. kierunku: INIS4\_PR2.2

=> Przedmiot: Programowanie .NET