

Programowanie równoległe Lab 3

Wprowadzenie

Dzisiejsze laboratorium będzie bazowało na dodawaniu kolejnych linii kodu liczących przy użyciu bibliotek Thread, ThreadPool.

Thread

Tworzy i kontroluje wątek, ustawia jego priorytet i pobiera jego stan.

Link do dokumentacji:

<https://learn.microsoft.com/en-us/dotnet/api/system.threading.thread?view=net-9.0>

ThreadPool

Udostępnia pulę wątków, które mogą służyć do wykonywania zadań, publikowania elementów roboczych, przetwarzania asynchronicznego we/wy, oczekiwania w imieniu innych wątków i czasomierzy przetwarzania.

Link do dokumentacji:

<https://learn.microsoft.com/en-us/dotnet/api/system.threading.threadpool?view=net-9.0>

Zadanie 1.

Rozbuduj aplikację z poprzedniego laboratorium 2 o możliwość przetwarzania procesów w oddzielnym wątku za pomocą kolejnych metod: klasy Thread, klasy ThreadPool lub klasy Parallel.

Po uruchomieniu aplikacji użytkownik powinien mieć możliwość wyboru jaką metodą będzie prowadzone przetwarzanie (BackgroundWorker , TPL, Thread, ThreadPool).

Zadanie 2.

Dodaj do aplikacji w każdej opcji kod sprawdzający czas działania całej aplikacji oraz czas działania poszczególnych wątków.

Zadanie 3.

Dodaj do aplikacji opcję uruchomienia wszystkich procesów w jednym wywołaniu (jeden po drugim) jako podsumowanie powinniśmy otrzymać zestawienie czasów wykonania poszczególnych procesów (BackgroundWorker , TPL, Thread, ThreadPool).

Podsumowanie i przesłanie wyników ćwiczeń:

1. Utwórz plik ZIP nazwany **wyniki_Lab_3_Nazwisko_Imie_yyyy_mm_dd.zip**
2. W pliku umieść rezultat twojej pracy ze wszystkich zadań.
3. Gotowy plik ZIP umieść na platformie e-learningowej.

Ocenianie:

1. Prace oddane w terminie – ocen z zakresu 5 - 2.
2. Prace oddane do 7 dni po terminie - ocen z zakresu 4.5 - 2.
3. Prace oddane 7-14 dni po terminie - ocen z zakresu 4 - 2.
4. Prace oddane 14-30 dni po terminie - ocen z zakresu 3 - 2.

Skalowanie ocen:

3 (dst.) –	51-64%
3+ (dst. plus) –	65-74%
4 (db.) -	75-84%
4+ (db. plus) -	85-94%
5 (bdb.) -	95-100%