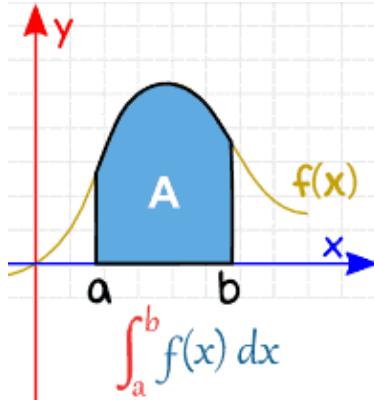


Programowanie równoległe Lab 2

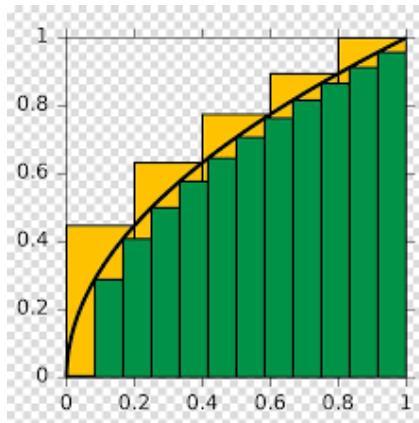
Wprowadzenie / Przypomnienie

Numeryczne liczenie całki.

O całce możemy myśleć jak o polu powierzchni pomiędzy wykresem funkcji a osią x.



Numeryczne liczenie całki możemy wykonać poprzez podział przedziału na bardzo małe odcinki, następnie liczeniu dla każdego odcinka wartości funkcji, po czym aproksymacji wartość całki dla odcinka za pomocą pola prostokąta lub trapezu. Suma wszystkich policzonych pól stanowi numerycznie wyliczoną wartość całki.



Zadanie 1.

Zadanie polega na numerycznym wyznaczeniu wartości całki dla jednej z funkcji:

$$y=2x + 2x^2, y=2x^2, y=2x-3$$

Właściwą funkcję użytkownik powinien wybrać podczas uruchomienia przetwarzania, a obliczenia powinny być wykonane na trzech przedziałach x:

- 1) od -10 do 10,
- 2) od -5 do 20,
- 3) od -5 do 0

z wykorzystaniem trapezów jako elementów aproksymujących. Obliczenia mają się wykonywać jednowątkowo, synchronicznie czyli obliczenia dla kolejnego przedziału powinny wystartować po zakończeniu poprzedniego.

Jako rezultat działania naszej aplikacji powinniśmy otrzymać wyniki dla poszczególnych przedziałów oraz całkowity czas w jakim zostały wykonane obliczenia np. w milisekundach.

Zadanie 2.

Bazując na zadaniu 1 dodaj nową funkcjonalność w taki sposób, aby obliczenia dla każdego z przedziałów zostały wykonane równocześnie, ale w oddzielnych wątkach za pomocą klasy BackgroundWorker.

Klasa BackgroundWorker umożliwia uruchamianie operacji na osobnym, dedykowanym wątku. Czasochłonne operacje, takie jak pobieranie i transakcje bazy danych, mogą spowodować, że interfejs użytkownika wydaje się wyglądać tak, jakby przestał odpowiadać podczas ich działania. Celem wizualizacji postępu obliczeń można użyć np. klasę BackgroundWorker.

<https://learn.microsoft.com/pl-pl/dotnet/api/system.componentmodel.backgroundworker?view=net-8.0>

Aplikacja powinna informować użytkownika o postępie obliczeń w krokach co 10%, oraz powinna umożliwiać przerwanie działania (zatrzymanie obliczeń). Na koniec aplikacja powinna wyświetlić na ekranie podsumowanie dla każdego z przedziałów oddzielnie.

W aplikacji wykorzystaj przykładowy interfejs i jego implementację.

```
interface IFunction
{
    0 references
    decimal GetY(decimal x);
}
```

Parametry do metody DoWork można przekazać za pomocą metody klasy BackgroundWorker:

```
public void RunWorkerAsync(object? argument)
```

Przekazane parametry przychodzą jako właściwość Argument instancji klasy DoWorkEventArgs.

Metoda ReportProgress również posiada swoją wersję z parametrem userState przenoszoną do „wątku głównego”

```
worker.ReportProgress(i * 10, );
▲ 2 of 2 ▼ void BackgroundWorker.ReportProgress(int percentProgress, object userState)
    Raises the BackgroundWorker.ProgressChanged event.
    userState: The state object passed to BackgroundWorker.RunWorkerAsync(object).
```

Rzutowanie typu object na inny typ referencyjny można wykonać za pomocą operatora 'as' (przykład rzutowania na typ WorkArguments):

```
private void DoWork(object sender, DoWorkEventArgs e)
{
    var arg = e.Argument as WorkArguments;
```

Jako rezultat działania naszej aplikacji powinniśmy otrzymać wyniki dla poszczególnych przedziałów oraz całkowity czas w jakim zostały wykonane obliczenia np. w milisekundach.

Zadanie 3.

Rozbuduj aplikację z zadania 2 o możliwość przetwarzania procesów w oddzielnym wątku za pomocą biblioteki TPL.

Funkcjonalności przerywania procesu oraz informowania o postępie i zakończeniu powinny zostać zachowane.

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

<https://learn.microsoft.com/pl-pl/dotnet/standard/parallel-programming/task-parallel-library-tpl>

Utwórz stosowny interfejs do rozdzielenia metody przetwarzania, oraz utwórz i wykorzystaj jego implementacje.

Pisz aplikację tak aby jak najmniej kodu związanego z logiką aplikacji było zdublowane (Zasada DRY)

Zadanie 4 (opcja dla chętnych).

Rozbuduj aplikację o testy wyników poprawności obliczeń oraz dokumentację w której dodasz podsumowania i przemyślenia na temat użytych klas (które implementacje są najskuteczniejsze zarówno jeśli chodzi o jakość obliczeń oraz szybkość działania).

Podsumowanie i przesłanie wyników ćwiczeń:

1. Utwórz plik ZIP nazwany **wyniki_Lab_2_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 – ocean z zakresu 5 - 2.
2. Prace oddane do 7 dni po terminie - ocean z zakresu 4.5 - 2.
3. Prace oddane 7-14 dni po terminie - ocean z zakresu 4 - 2.
4. Prace oddane 14-30 dni po terminie - ocean 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%