

Pi

Generated by Doxygen 1.12.0

1 Class Index	1
1.1 Class List	1
2 File Index	3
2.1 File List	3
3 Class Documentation	5
3.1 Integration Class Reference	5
3.1.1 Detailed Description	5
3.1.2 Constructor & Destructor Documentation	5
3.1.2.1 Integration()	5
3.1.3 Member Function Documentation	6
3.1.3.1 calculatePi()	6
3.1.3.2 function()	6
3.2 ThreadData Struct Reference	7
3.2.1 Detailed Description	7
3.2.2 Member Data Documentation	8
3.2.2.1 end	8
3.2.2.2 result	8
3.2.2.3 start	8
3.2.2.4 step	8
4 File Documentation	9
4.1 Pi/Integration.cpp File Reference	9
4.1.1 Detailed Description	9
4.2 Pi/Integration.h File Reference	9
4.2.1 Macro Definition Documentation	10
4.2.1.1 INTEGRATION_H	10
4.3 Integration.h	10
4.4 Pi/Pi.cpp File Reference	10
4.4.1 Function Documentation	11
4.4.1.1 main()	11
Index	13

Chapter 1

Class Index

1.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Integration	
Klasa Integration do obliczania wartoci liczby PI metod ¹ numeryczn ¹	5
ThreadData	
Struktura przechowuj ¹ ca dane dla poszczególnych w ¹ tków	7

Chapter 2

File Index

2.1 File List

Here is a list of all files with brief descriptions:

Pi/ Integration.cpp	
Implementacja klasy Integration , służącej do obliczania wartości liczby metodą całkowania numerycznego przy użyciu wielowątkowości	9
Pi/ Integration.h	9
Pi/ Pi.cpp	10

Chapter 3

Class Documentation

3.1 Integration Class Reference

Klasa [Integration](#) do obliczania wartoci liczby PI metod¹ numeryczn¹.

```
#include <Integration.h>
```

Public Member Functions

- [Integration](#) (long long num_intervals, int num_threads)
Konstruktor klasy [Integration](#).
- double [calculatePi](#) ()
Główna metoda obliczająca wartość liczby PI.

Static Public Member Functions

- static double [function](#) (double x)
Funkcja reprezentująca wzór matematyczny do zintegrowania.

3.1.1 Detailed Description

Klasa [Integration](#) do obliczania wartoci liczby PI metod¹ numeryczn¹.

Klasa wykorzystuje podejście wielowątkowe do podziału pracy na kilka wątków w celu obliczenia przybliżonej wartoci liczby PI.

3.1.2 Constructor & Destructor Documentation

3.1.2.1 Integration()

```
Integration::Integration (  
    long long num_intervals,  
    int num_threads)
```

Konstruktor klasy [Integration](#).

Inicjalizuje liczbę przedziałów, liczbę wątków oraz inne wymagane zmienne.

Parameters

<i>num_intervals</i>	Liczba przedziałów do podziału obszaru.
<i>num_threads</i>	Liczba wątków do użycia w obliczeniach.
<i>num_intervals</i>	Liczba przedziałów do całkowania.
<i>num_threads</i>	Liczba wątków, które mają być wykorzystane do równoległego obliczania.

< Obliczenie długości kroku całkowania.

< Rezerwacja miejsca w wektorze dla wątków.

< Inicjalizacja wektora danych dla wątków.

3.1.3 Member Function Documentation

3.1.3.1 calculatePi()

```
double Integration::calculatePi ()
```

Główna metoda obliczająca wartość liczby PI.

Główna metoda klasy obliczająca przybliżoną wartość liczby π metodą numerycznego całkowania.

Metoda dzieli obszar na przedziały i rozdziela pracę między wątki.

Returns

Przybliżona wartość liczby PI.

Przybliżona wartość liczby π .

< Reszta do równomiernego rozdzielenia zakresu.

< Początkowy indeks zakresu dla wątków.

< Inicjalizacja danych wątku.

< Aktualizacja początku zakresu.

< Przekazanie danych wątku do funkcji.

< Synchronizacja wątków z wątkiem głównym.

< Całkowita suma.

< Korekta błędów zaokrąglenia.

< Skorygowanie wyniku w celu zmniejszenia błędów.

< Aktualizacja sumy po uwzględnieniu korekty.

< Uaktualnienie kompensacji błędów.

< Zapisanie nowej sumy.

< Wynik pomnożony przez 4 dla uzyskania liczby π .

3.1.3.2 function()

```
double Integration::function (
    double x) [static]
```

Funkcja reprezentująca wzór matematyczny do zintegrowania.

Funkcja matematyczna, której całka jest obliczana.

Parameters

<i>x</i>	Wartość wejściowa funkcji.
----------	----------------------------

Returns

Wynik obliczenia funkcji dla danego *x*.

Parameters

<i>x</i>	Argument funkcji.
----------	-------------------

Returns

Wartość funkcji w punkcie $x(1 - x^2)$.

The documentation for this class was generated from the following files:

- [Pi/Integration.h](#)
- [Pi/Integration.cpp](#)

3.2 ThreadData Struct Reference

Struktura przechowująca dane dla poszczególnych wątków.

```
#include <Integration.h>
```

Public Attributes

- long long [start](#)
Początkowy indeks przedziału dla wątku.
- long long [end](#)
Końcowy indeks przedziału dla wątku.
- double [step](#)
Długość kroku w podziale przedziału.
- double [result](#) = 0.0
Wynik obliczeń w danym wątku (domyślnie 0.0).

3.2.1 Detailed Description

Struktura przechowująca dane dla poszczególnych wątków.

Struktura zawiera informacje o zakresie iteracji, kroku oraz wynikach obliczeń.

3.2.2 Member Data Documentation

3.2.2.1 end

```
long long ThreadData::end
```

Końcowy indeks przedzia³u dla w¹tku.

3.2.2.2 result

```
double ThreadData::result = 0.0
```

Wynik obliczeń w danym w¹tku (domylnie 0.0).

3.2.2.3 start

```
long long ThreadData::start
```

Początkowy indeks przedzia³u dla w¹tku.

3.2.2.4 step

```
double ThreadData::step
```

D³ugość kroku w podziale przedzia³u.

The documentation for this struct was generated from the following file:

- Pi/[Integration.h](#)

Chapter 4

File Documentation

4.1 Pi/Integration.cpp File Reference

Implementacja klasy [Integration](#), służącej do obliczania wartości liczby π metodą całkowania numerycznego przy użyciu wielowątkowości.

```
#include "Integration.h"
#include <cmath>
```

4.1.1 Detailed Description

Implementacja klasy [Integration](#), służącej do obliczania wartości liczby π metodą całkowania numerycznego przy użyciu wielowątkowości.

4.2 Pi/Integration.h File Reference

```
#include <vector>
#include <thread>
#include <functional>
```

Classes

- struct [ThreadData](#)
Struktura przechowująca dane dla poszczególnych wątków.
- class [Integration](#)
Klasa [Integration](#) do obliczania wartości liczby π metodą numeryczną.

Macros

- `#define INTEGRATION_H`

4.2.1 Macro Definition Documentation

4.2.1.1 INTEGRATION_H

```
#define INTEGRATION_H
```

4.3 Integration.h

[Go to the documentation of this file.](#)

```
00001 #pragma once
00002
00003 #ifndef INTEGRATION_H
00004 #define INTEGRATION_H
00005
00006 #include <vector>
00007 #include <thread>
00008 #include <functional>
00009
00015 struct ThreadData {
00016     long long start;
00017     long long end;
00018     double step;
00019     double result = 0.0;
00020 };
00021
00028 class Integration {
00029 public:
00038     Integration(long long num_intervals, int num_threads);
00039
00047     double calculatePi();
00048
00055     static double function(double x);
00056
00057 private:
00058     long long num_intervals;
00059     int num_threads;
00060     double step;
00061     std::vector<std::thread> threads;
00062     std::vector<ThreadData> thread_data;
00063
00071     static void threadTask(ThreadData& data);
00072 };
00073
00074 #endif // INTEGRATION_H
```

4.4 Pi/Pi.cpp File Reference

```
#include <iostream>
#include <chrono>
#include "Integration.h"
```

Functions

- `int main ()`

Funkcja główna programu.

4.4.1 Function Documentation

4.4.1.1 main()

```
int main ()
```

Funkcja główna programu.

Program oblicza przybliżoną wartość liczby PI przy użyciu metody numerycznej. Użytkownik podaje liczbę przedziałów i liczbę wątków, a wynik oraz czas obliczeń są wypisywane na standardowym wyjściu.

Returns

Zwraca 0 w przypadku powodzenia, 1 w przypadku błędu.

< Liczba przedziałów do podziału obszaru pod krzywą.

< Liczba wątków do wykorzystania w obliczeniach.

< Wczytanie liczby przedziałów od użytkownika.

< Wczytanie liczby wątków od użytkownika.

Walidacja danych wejściowych.

Program sprawdza, czy liczba przedziałów i liczba wątków są dodatnie. W przypadku błędnych danych wypisuje komunikat i przerywa działanie.

< Zakończenie programu z kodem błędu.

Tworzenie obiektu klasy [Integration](#).

Obiekt tej klasy odpowiada za obliczenia przybliżonej wartości liczby PI z użyciem podanej liczby przedziałów i wątków.

Obliczanie wartości liczby PI.

Wywoływana metoda klasy [Integration](#) dokonuje obliczeń numerycznych.

Returns

Przybliżona wartość liczby PI.

Obliczenie czasu wykonania obliczeń.

Różnica czasu zakończenia i rozpoczęcia obliczeń podana w sekundach.

< Przybliżona wartość liczby PI.

< Czas wykonania obliczeń.

< Zakończenie programu.

Index

calculatePi
 Integration, [6](#)

end
 ThreadData, [8](#)

function
 Integration, [6](#)

Integration, [5](#)
 calculatePi, [6](#)
 function, [6](#)
 Integration, [5](#)

Integration.h
 INTEGRATION_H, [10](#)
INTEGRATION_H
 Integration.h, [10](#)

main
 Pi.cpp, [11](#)

Pi.cpp
 main, [11](#)
Pi/Integration.cpp, [9](#)
Pi/Integration.h, [9](#), [10](#)
Pi/Pi.cpp, [10](#)

result
 ThreadData, [8](#)

start
 ThreadData, [8](#)

step
 ThreadData, [8](#)

ThreadData, [7](#)
 end, [8](#)
 result, [8](#)
 start, [8](#)
 step, [8](#)