

My Project

Generated by Doxygen 1.15.0

1 Class Index	1
1.1 Class List	1
2 File Index	3
2.1 File List	3
3 Class Documentation	5
3.1 PiCalculator Class Reference	5
3.1.1 Detailed Description	5
3.1.2 Member Function Documentation	5
3.1.2.1 calculate()	5
4 File Documentation	7
4.1 PiCalculator.cpp File Reference	7
4.1.1 Detailed Description	7
4.2 PiCalculator.h File Reference	7
4.2.1 Detailed Description	7
4.3 PiCalculator.h	8
4.4 Projekt05.cpp File Reference	8
4.4.1 Detailed Description	8
4.4.2 Function Documentation	8
4.4.2.1 main()	8
Index	9

Chapter 1

Class Index

1.1 Class List

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

PiCalculator	Klasa odpowiedzialna za obliczenia matematyczne liczby Pi	5
--------------	---	---

Chapter 2

File Index

2.1 File List

Here is a list of all documented files with brief descriptions:

PiCalculator.cpp	
Implementacja metod klasy PiCalculator	7
PiCalculator.h	
Definicja klasy PiCalculator	7
Projekt05.cpp	
Glowny modul programu wykonawczego	8

Chapter 3

Class Documentation

3.1 PiCalculator Class Reference

Klasa odpowiedzialna za obliczenia matematyczne liczby PI.

```
#include <PiCalculator.h>
```

Public Member Functions

- double **calculate** (long long num_steps, int num_threads)
Metoda obliczajaca PI.

3.1.1 Detailed Description

Klasa odpowiedzialna za obliczenia matematyczne liczby PI.

- Klasa udostepnia metody do rownoleglego obliczania wartosci PI z wykorzystaniem biblioteki standardowej watkow C++11.

3.1.2 Member Function Documentation

3.1.2.1 calculate()

```
double PiCalculator::calculate (
    long long num_steps,
    int num_threads)
```

Metoda obliczajaca PI.

Parameters

<i>num_steps</i>	Liczba iteracji algorytmu.
<i>num_threads</i>	Liczba wątków roboczych.

Returns

double Wynik obliczen.

Wektor przechowujacy wyniki czastkowe z kazdego wątku Uzywamy wektora, aby uniknac wyscigu danych (data race) przy sumowaniu

Obliczamy ile krokow petli przypada na jeden watek

Uruchamiamy watek z wyrazeniem lambda Przekazujemy zmienne przez wartosc [=] oraz wektor wynikow przez referencje [&partial_sums]

Ostatni watek bierze na siebie ewentualna reszte z dzielenia

Petla całkowania numerycznego (metoda prostokatow z punktem srodkowym)

Zapisz wynik czastkowy w odpowiednim miejscu wektora

Czekamy na zakonczenie wszystkich wątkow (join)

Sumujemy wyniki czastkowe

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

- [PiCalculator.h](#)
- [PiCalculator.cpp](#)

Chapter 4

File Documentation

4.1 PiCalculator.cpp File Reference

Implementacja metod klasy [PiCalculator](#).

```
#include "PiCalculator.h"
#include <thread>
#include <vector>
```

4.1.1 Detailed Description

Implementacja metod klasy [PiCalculator](#).

4.2 PiCalculator.h File Reference

Definicja klasy [PiCalculator](#).

Classes

- class [PiCalculator](#)
Klasa odpowiedzialna za obliczenia matematyczne liczby Pi.

4.2.1 Detailed Description

Definicja klasy [PiCalculator](#).

4.3 PiCalculator.h

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

```
00001
00005
00006 #ifndef PICALCULATOR_H
00007 #define PICALCULATOR_H
00008
00015 class PiCalculator {
00016 public:
00023     double calculate(long long num_steps, int num_threads);
00024 };
00025
00026 #endif
```

4.4 Projekt05.cpp File Reference

Glowny modul programu wykonawczego.

```
#include <iostream>
#include <chrono>
#include <iomanip>
#include <string>
#include "PiCalculator.h"
```

Functions

- int [main](#) (int argc, char *argv[])

4.4.1 Detailed Description

Glowny modul programu wykonawczego.

- Odpowiada za:
- Parsowanie argumentow linii komend.
- Inicjalizacje procesu obliczeniowego.
- Precyzyjny pomiar czasu rzeczywistego wykonania zadania.

4.4.2 Function Documentation

4.4.2.1 main()

```
int main (
    int argc,
    char * argv[ ])
```

Sprawdzamy, czy podano odpowiednia liczbe argumentow Oczekujemy: ./Projekt05 <liczba_krokow> <liczba_watrow>

Pomiar czasu START

Obliczenia

Pomiar czasu STOP

Wypisujemy TYLKO czas na koncu, aby Python łatwo to odczytał. Python bedzie czytał ostatnia linie wyjścia.

Index

calculate
PiCalculator, [5](#)

main
Projekt05.cpp, [8](#)

PiCalculator, [5](#)
 calculate, [5](#)
PiCalculator.cpp, [7](#)
PiCalculator.h, [7](#)
Projekt05.cpp, [8](#)
 main, [8](#)