

Tablica Dynamiczna

Generated by Doxygen 1.9.3

1 Strona glowna	1
1.1 Wstep	1
2 Namespace Index	3
2.1 Namespace List	3
3 Class Index	5
3.1 Class List	5
4 File Index	7
4.1 File List	7
5 Namespace Documentation	9
5.1 AiSD Namespace Reference	9
5.1.1 Function Documentation	10
5.1.1.1 _setNow()	10
5.1.1.2 _timeTook()	10
5.1.1.3 ClearLogTxt()	10
5.1.1.4 DistortionsSimulation()	10
5.1.1.5 DoFunction()	11
5.1.1.6 FunctionByNO()	11
5.1.1.7 Log()	11
5.1.1.8 OverflowTable()	12
5.1.1.9 Presentation()	12
6 Class Documentation	13
6.1 AiSD::DynamicArray Class Reference	13
6.1.1 Constructor & Destructor Documentation	13
6.1.1.1 DynamicArray()	14
6.1.1.2 ~DynamicArray()	14
6.1.2 Member Function Documentation	14
6.1.2.1 Erase()	14
6.1.2.2 Insert()	14
6.1.2.3 IsEmpty()	14
6.1.2.4 IsFull()	14
6.1.2.5 PopBack()	14
6.1.2.6 PopFront()	15
6.1.2.7 Print()	15
6.1.2.8 PushBack()	15
6.1.2.9 PushFront()	15
6.1.2.10 Save()	15
6.1.2.11 Space()	15
6.1.3 Friends And Related Function Documentation	15
6.1.3.1 DoFunction	15

6.1.3.2 OverflowTable	16
6.2 AiSD::noneV Struct Reference	16
6.2.1 Detailed Description	16
7 File Documentation	17
7.1 include/DoxygenMainPage.h File Reference	17
7.2 DoxygenMainPage.h	17
7.3 include/DynamicArray.h File Reference	17
7.3.1 Macro Definition Documentation	17
7.3.1.1 T	18
7.4 DynamicArray.h	18
7.5 include/TestingFunction.hpp File Reference	18
7.6 TestingFunction.hpp	19
Index	21

Chapter 1

Strona glowna

1.1 Wstep

[Kliknij tutaj by przejsc do dokumentacji funkcji testujacych](#)
[Kliknij tutaj by przejsc do dokumentacji Dynamic Array](#)

Chapter 2

Namespace Index

2.1 Namespace List

Here is a list of all namespaces with brief descriptions:

AiSD	9
----------------------------	---

Chapter 3

Class Index

3.1 Class List

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

AiSD::DynamicArray	13
AiSD::noneV	
Pusta struktura. Varianty nie moga przyjmowac typ void. Alternatywnie moglem uzyc wbudowanego "monostate"	16

Chapter 4

File Index

4.1 File List

Here is a list of all files with brief descriptions:

include/ DoxygenMainPage.h	17
include/ DynamicArray.h	17
include/ TestingFunction.hpp	18

Chapter 5

Namespace Documentation

5.1 AiSD Namespace Reference

Classes

- class [DynamicArray](#)
- struct [noneV](#)

Pusta struktura. Varianty nie mogą przyjmować typ void. Alternatywnie mogłem użyć wbudowanego "monostate".

Functions

- void [Log](#) (std::string src, std::string in)
Ta funkcja zapisuje do pliku o ścieżce src zawartość in.
- auto [DoFunction](#) ([DynamicArray](#) &arr, int NO, [T](#) t, size_t i)
Ta funkcja wykonuje funkcje Dynamic Array, gdzie NO to numer funkcji od 0. Zapisze wszystkie operacje do pliku (LogFileName "Log.txt") na wypadek crashu. Mierzy także czas wykonywania operacji w mikrosekundach. Zwraca variant<bool,size_t,noneV (nothing)> (jako to co pierwotna funkcja Dynamic Array nam zwracała)
- auto [FunctionByNO](#) (int NO)
Zwraca makro funkcji z Dynamic Array. A dokładniej std::function<variant<bool, size_t,[noneV](#)> (AiSD::DynamicArray& a, T t1,size_t i1)>
- void [DistortionsSimulation](#) ([DynamicArray](#) &arr, int t)
Symulacja zakłócen funkcji. Wykonuje losowe funkcje dla losowych argumentów. Mierzy czas. Zapisuje dane do pliku log, na wypadek crashu.
- void [OverflowTable](#) ([DynamicArray](#) &arr)
Wykonuje te same funkcje Dynamic Array do pusta i pełna. Pesymistyczny scenariusz jest taki, że przyjmujemy zawsze te największe liczby jakie mogą wprowadzić. Mierzy czas dla funkcji powtarzanej tyle razy jaka jest wielkość tablicy. Przy okazji testuje takie sytuacje kiedy dodajemy kolejny element do pełnej tablicy oraz usuwamy element z pustej tablicy. Robi to poza mierzeniem czasu. Zmierzony czas jest w mikrosekundach i wyświetla się na ekranie.
- void [Presentation](#) ([DynamicArray](#) &arr)
NIESKONCZONA PETLA!!! Korzystanie z klasy Dynamic Array z poziomu wiersza poleceń. (Reczne testowanie)
- auto [_setNow](#) ()
Mala funkcja do otrzymania bieżącego czasu.
- std::string [_timeTook](#) (auto a, auto b)
Zwraca długość czasu pomiędzy dwoma chrono podany w string w mikrosekundach.
- void [ClearLogTxt](#) ()
Oproźnia plik Log.

5.1.1 Function Documentation

5.1.1.1 `_setNow()`

```
auto AiSD::_setNow ( )
```

Mala funkcja do otrzymania bierzącego czasu.

5.1.1.2 `_timeTook()`

```
std::string AiSD::_timeTook (
    auto a,
    auto b )
```

Zwraca dlugosc czasu pomiedzy dwoma chrono podany w string w mikrosekundach.

5.1.1.3 `ClearLogTxt()`

```
void AiSD::ClearLogTxt ( )
```

Oproznia plik Log.

5.1.1.4 `DistortionsSimulation()`

```
void AiSD::DistortionsSimulation (
    DynamicArray & arr,
    int t )
```

Symulacja zaklocen funkcji. Wykonuje losowe funkcje dla losowych argumentow. Mierzy czas. Zapisuje dane do pliku log, na wypadek crashu.

Parameters

<i>arr</i>	Dynamic Array
<i>t</i>	Ile losowych operacji ma sie wykonac

5.1.1.5 DoFunction()

```
auto AiSD::DoFunction (
    DynamicArray & arr,
    int NO,
    T t,
    size_t i )
```

Ta funkcja wykonuje funkcje Dynamic Array, gdzie NO to numer funkcji od 0. Zapisze wszystkie operacje do pliku (LogFileName "Log.txt") na wypadek crashu. Mierzy także czas wykonywania operacji w mikrosekundach. Zwraca `variant<bool,size_t,noneV (nothing)>` (jako to co pierwotna funkcja Dynamic Array nam zwracała)

Parameters

<i>arr</i>	Dynamic Array
<i>NO</i>	Numer funkcji Dynamic Array od 0
<i>t</i>	Pierwszy parametr funkcji Dynamic Array
<i>i</i>	Drugi parametr funkcji Dynamic Array

5.1.1.6 FunctionByNO()

```
auto AiSD::FunctionByNO (
    int NO )
```

Zwraca makro funkcji z Dynamic Array. A dokładniej `std::function<variant <bool, size_t,noneV> (AiSD::DynamicArray& a,T t1,size_t i1)>`

Parameters

<i>NO</i>	Numer funkcji Dynamic Array od 0
-----------	----------------------------------

5.1.1.7 Log()

```
void AiSD::Log (
    std::string src,
    std::string in )
```

Ta funkcja zapisuje do pliku o sciezce src zawartosc in.

Parameters

<i>src</i>	Sciezka pliku
<i>in</i>	Zawartosc do dodania do pliku

5.1.1.8 OverflowTable()

```
void AiSD::OverflowTable (
    DynamicArray & arr )
```

Wykonuje te same funkcje Dynamic Array do pusta i pelna. Pesymistyczny scenariusz jest taki, ze przyjmujemy zawsze te najwieksze liczby jakie moge wprowadzic. Mierzy czas dla funkcji powtarzanej tyle razy jaka jest wielkosc tablicy. Przy okazji testuje takie sytuacje kiedy dodajemy kolejny element do pelnej tablicy oraz usuwamy element z pustej tablicy. Robi to poza mierzaniem czasu. Zmierzony czas jest w mikrosekundach i wyswietla sie na ekranie.

Parameters

<i>arr</i>	Dynamic Array
------------	---------------

5.1.1.9 Presentation()

```
void AiSD::Presentation (
    DynamicArray & arr )
```

NIESKONCZONA PETLA!!! Korzystanie z klasy Dynamic Array z poziomu wiersza polecen. (Reczne testowanie)

Parameters

<i>arr</i>	Dynamic Array
------------	---------------

Chapter 6

Class Documentation

6.1 AiSD::DynamicArray Class Reference

```
#include <DynamicArray.h>
```

Public Member Functions

- [DynamicArray](#) (T rozmiar)
- virtual [~DynamicArray](#) ()
- void [Print](#) ()
- void [Save](#) ()
- bool [IsEmpty](#) ()
- bool [IsFull](#) ()
- size_t [Space](#) ()
- void [PushBack](#) (T t)
- void [PopBack](#) ()
- void [PushFront](#) (T t)
- void [PopFront](#) ()
- void [Insert](#) (T t, size_t i)
- void [Erase](#) (size_t i)

Friends

- auto [DoFunction](#) ([DynamicArray](#) &, int, T, size_t)

Ta funkcja wykonuje funkcje Dynamic Array, gdzie NO to numer funkcji od 0. Zapisze wszystkie operacje do pliku (LogFileName "Log.txt") na wypadek crashu. Mierzy także czas wykonywania operacji w mikrosekundach. Zwraca `variant<bool,size_t,noneV (nothing)>` (jako to co pierwotna funkcja Dynamic Array nam zwracała)

- void [OverflowTable](#) ([DynamicArray](#) &)

Wykonuje te same funkcje Dynamic Array do pusta i pelna. Pesymistyczny scenariusz jest taki, ze przyjmujemy zawsze te najwieksze liczby jakie moge wprowadzic. Mierzy czas dla funkcji powtarzanej tyle razy jaka jest wielkosc tablicy. Przy okazji testuje takie sytuacje kiedy dodajemy kolejny element do pelnej tablicy oraz usuwamy element z pustej tablicy. Robi to poza mierzaniem czasu. Zmierzony czas jest w mikrosekundach i wyswietla sie na ekranie.

6.1.1 Constructor & Destructor Documentation

6.1.1.1 DynamicArray()

```
AiSD::DynamicArray::DynamicArray (
    T rozmiar )
```

6.1.1.2 ~DynamicArray()

```
virtual AiSD::DynamicArray::~~DynamicArray ( ) [virtual]
```

6.1.2 Member Function Documentation

6.1.2.1 Erase()

```
void AiSD::DynamicArray::Erase (
    size_t i )
```

6.1.2.2 Insert()

```
void AiSD::DynamicArray::Insert (
    T t,
    size_t i )
```

6.1.2.3 IsEmpty()

```
bool AiSD::DynamicArray::IsEmpty ( )
```

6.1.2.4 IsFull()

```
bool AiSD::DynamicArray::IsFull ( )
```

6.1.2.5 PopBack()

```
void AiSD::DynamicArray::PopBack ( )
```

6.1.2.6 PopFront()

```
void AiSD::DynamicArray::PopFront ( )
```

6.1.2.7 Print()

```
void AiSD::DynamicArray::Print ( )
```

6.1.2.8 PushBack()

```
void AiSD::DynamicArray::PushBack (
    T t )
```

6.1.2.9 PushFront()

```
void AiSD::DynamicArray::PushFront (
    T t )
```

6.1.2.10 Save()

```
void AiSD::DynamicArray::Save ( )
```

6.1.2.11 Space()

```
size_t AiSD::DynamicArray::Space ( )
```

6.1.3 Friends And Related Function Documentation

6.1.3.1 DoFunction

```
auto DoFunction (
    DynamicArray & ,
    int ,
    T ,
    size_t ) [friend]
```

Ta funkcja wykonuje funkcje Dynamic Array, gdzie NO to numer funkcji od 0. Zapisze wszystkie operacje do pliku (LogFileName "Log.txt") na wypadek crashu. Mierzy także czas wykonywania operacji w mikrosekundach. Zwraca `variant<bool,size_t,noneV (nothing)>` (jako to co pierwotna funkcja Dynamic Array nam zwracała)

Parameters

<i>arr</i>	Dynamic Array
<i>NO</i>	Numer funkcji Dynamic Array od 0
<i>t</i>	Pierwszy parametr funkcji Dynamic Array
<i>i</i>	Drugi parametr funkcji Dynamic Array

6.1.3.2 OverflowTable

```
void OverflowTable (
    DynamicArray & ) [friend]
```

Wykonuje te same funkcje Dynamic Array do pusta i pelna. Pesymistyczny scenariusz jest taki, ze przyjmujemy zawsze te najwieksze liczby jakie moge wprowadzic. Mierzy czas dla funkcji powtarzanej tyle razy jaka jest wielkosc tablicy. Przy okazji testuje takie sytuacje kiedy dodajemy kolejny element do pelnej tablicy oraz usuwamy element z pustej tablicy. Robi to poza mierzaniem czasu. Zmierzony czas jest w mikrosekundach i wyswietla sie na ekranie.

Parameters

<i>arr</i>	Dynamic Array
------------	---------------

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

- [include/DynamicArray.h](#)

6.2 AiSD::noneV Struct Reference

Pusta struktura. Varianty nie moga przyjmowac typ void. Alternatywnie moglem uzyc wbudowanego "monostate".

```
#include <TestingFunction.hpp>
```

6.2.1 Detailed Description

Pusta struktura. Varianty nie moga przyjmowac typ void. Alternatywnie moglem uzyc wbudowanego "monostate".

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

- [include/TestingFunction.hpp](#)

Chapter 7

File Documentation

7.1 include/DoxygenMainPage.h File Reference

7.2 DoxygenMainPage.h

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

1

7.3 include/DynamicArray.h File Reference

```
#include <cstdint>
#include <iostream>
#include <array>
#include <fstream>
#include <sstream>
```

Classes

- class [AiSD::DynamicArray](#)

Namespaces

- namespace [AiSD](#)

Macros

- #define [T](#) long

7.3.1 Macro Definition Documentation

7.3.1.1 T

```
#define T long
```

7.4 DynamicArray.h

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

```
1 #ifndef DYNAMICARRAY_H
2 #define DYNAMICARRAY_H
3 #define T long
4
5 #include <cstdint>
6 #include <iostream>
7 #include <array>
8 #include <fstream>
9 #include <sstream>
10
11
12 using namespace std;
13
14 namespace AiSD{
15     class DynamicArray
16     {
17
18     public:
19         DynamicArray(T rozmiar);
20         virtual ~DynamicArray();
21         void Print();
22         void Save();
23         bool IsEmpty();
24         bool IsFull();
25         size_t Space();
26         void PushBack(T t);
27         void PopBack();
28         void PushFront(T t);
29         void PopFront();
30         void Insert(T t, size_t i);
31         void Erase(size_t i);
32
33     protected:
34
35     private:
36         friend auto DoFunction(DynamicArray&,int,T,size_t);
37         friend void OverflowTable(DynamicArray&);
38
39         T wczytajLiczbeCalkowita();
40         size_t capacity;
41         size_t size;
42         const string nazwaPliku="ZawartoscTablicy.txt";
43         T *tablica;
44
45
46     };
47 }
48 #endif // DYNAMICARRAY_H
```

7.5 include/TestingFunction.hpp File Reference

```
#include "DynamicArray.h"
#include <cstdint>
#include <sstream>
#include <functional>
#include <chrono>
#include <string>
#include <random>
#include <variant>
```

Classes

- struct `AiSD::noneV`

Pusta struktura. Varianty nie mogą przyjmować typ void. Alternatywnie mogłem użyć wbudowanego "monostate".

Namespaces

- namespace `AiSD`

Functions

- void `AiSD::Log` (std::string src, std::string in)
Ta funkcja zapisuje do pliku o ścieżce src zawartość in.
- auto `AiSD::DoFunction` (DynamicArray &arr, int NO, T t, size_t i)
Ta funkcja wykonuje funkcje Dynamic Array, gdzie NO to numer funkcji od 0. Zapisze wszystkie operacje do pliku (LogFileName "Log.txt") na wypadek crashu. Mierzy także czas wykonywania operacji w mikrosekundach. Zwraca variant<bool,size_t,noneV (nothing)> (jako to co pierwotna funkcja Dynamic Array nam zwracała)
- auto `AiSD::FunctionByNO` (int NO)
Zwraca makro funkcji z Dynamic Array. A dokładniej std::function<variant<bool, size_t,noneV> (AiSD::DynamicArray& a,T t1,size_t i1)>
- void `AiSD::DistortionsSimulation` (DynamicArray &arr, int t)
Symulacja zakłócen funkcji. Wykonuje losowe funkcje dla losowych argumentów. Mierzy czas. Zapisuje dane do pliku log, na wypadek crashu.
- void `AiSD::OverflowTable` (DynamicArray &arr)
Wykonuje te same funkcje Dynamic Array do pusta i pełna. Pesymistyczny scenariusz jest taki, że przyjmujemy zawsze te największe liczby jakie mogą wprowadzić. Mierzy czas dla funkcji powtarzanej tyle razy jaka jest wielkość tablicy. Przy okazji testuje takie sytuacje kiedy dodajemy kolejny element do pełnej tablicy oraz usuwamy element z pustej tablicy. Robi to poza mierzeniem czasu. Zmierzony czas jest w mikrosekundach i wyświetla się na ekranie.
- void `AiSD::Presentation` (DynamicArray &arr)
NIESKONCZONA PETLA!!! Korzystanie z klasy Dynamic Array z poziomu wiersza poleceń. (Reczne testowanie)
- auto `AiSD::_setNow` ()
Mala funkcja do otrzymania bieżącego czasu.
- std::string `AiSD::_timeTook` (auto a, auto b)
Zwraca długość czasu pomiędzy dwoma chrono podany w string w mikrosekundach.
- void `AiSD::ClearLogTxt` ()
Oproźnia plik Log.

7.6 TestingFunction.hpp

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

```
1  /*
2  *TESTINGFUNCTION_HPP_FILE
3  */
4  #ifndef TESTINGFUNCTION_HPP
5  #define TESTINGFUNCTION_HPP
6
7  #include "DynamicArray.h"
8
9  #include <cstdint>
10 #include <sstream>
11 #include <functional>
12 #include <chrono>
13 #include <string>
14 #include <random>
15 #include <variant>
16
```

```
17
18 namespace AiSD
19 {
20
29     void Log(std::string src, std::string in);
30
45     auto DoFunction(DynamicArray& arr, int NO, T t, size_t i);
46
53     auto FunctionByNO(int NO);
54
55
64     void DistortionsSimulation(DynamicArray& arr, int t);
65
75     void OverflowTable(DynamicArray& arr);
76
83     void Presentation(DynamicArray& arr);
84
88     auto _setNow();
89
93     std::string _timeTook(auto a, auto b);
94
98     void ClearLogTxt();
99
103     struct noneV{};
104 }
105
106
107
108 #endif // TESTINGFUNCTION_HPP
```


Index

- [_setNow](#)
 - [AiSD, 10](#)
 - [_timeTook](#)
 - [AiSD, 10](#)
 - [~DynamicArray](#)
 - [AiSD::DynamicArray, 14](#)
- [AiSD, 9](#)
 - [_setNow, 10](#)
 - [_timeTook, 10](#)
 - [ClearLogTxt, 10](#)
 - [DistortionsSimulation, 10](#)
 - [DoFunction, 10](#)
 - [FunctionByNO, 11](#)
 - [Log, 11](#)
 - [OverflowTable, 11](#)
 - [Presentation, 12](#)
- [AiSD::DynamicArray, 13](#)
 - [~DynamicArray, 14](#)
 - [DoFunction, 15](#)
 - [DynamicArray, 13](#)
 - [Erase, 14](#)
 - [Insert, 14](#)
 - [IsEmpty, 14](#)
 - [IsFull, 14](#)
 - [OverflowTable, 16](#)
 - [PopBack, 14](#)
 - [PopFront, 14](#)
 - [Print, 15](#)
 - [PushBack, 15](#)
 - [PushFront, 15](#)
 - [Save, 15](#)
 - [Space, 15](#)
- [AiSD::noneV, 16](#)
- [ClearLogTxt](#)
 - [AiSD, 10](#)
- [DistortionsSimulation](#)
 - [AiSD, 10](#)
- [DoFunction](#)
 - [AiSD, 10](#)
 - [AiSD::DynamicArray, 15](#)
- [DynamicArray](#)
 - [AiSD::DynamicArray, 13](#)
- [DynamicArray.h](#)
 - [T, 17](#)
- [Erase](#)
 - [AiSD::DynamicArray, 14](#)
- [FunctionByNO](#)
 - [AiSD, 11](#)
- [include/DoxygenMainPage.h, 17](#)
- [include/DynamicArray.h, 17, 18](#)
- [include/TestingFunction.hpp, 18, 19](#)
- [Insert](#)
 - [AiSD::DynamicArray, 14](#)
- [IsEmpty](#)
 - [AiSD::DynamicArray, 14](#)
- [IsFull](#)
 - [AiSD::DynamicArray, 14](#)
- [Log](#)
 - [AiSD, 11](#)
- [OverflowTable](#)
 - [AiSD, 11](#)
 - [AiSD::DynamicArray, 16](#)
- [PopBack](#)
 - [AiSD::DynamicArray, 14](#)
- [PopFront](#)
 - [AiSD::DynamicArray, 14](#)
- [Presentation](#)
 - [AiSD, 12](#)
- [Print](#)
 - [AiSD::DynamicArray, 15](#)
- [PushBack](#)
 - [AiSD::DynamicArray, 15](#)
- [PushFront](#)
 - [AiSD::DynamicArray, 15](#)
- [Save](#)
 - [AiSD::DynamicArray, 15](#)
- [Space](#)
 - [AiSD::DynamicArray, 15](#)
- [T](#)
 - [DynamicArray.h, 17](#)