Tablica Dynamiczna

Generated by Doxygen 1.9.3

Chapter 1

Strona glowna

1.1 Wstep

< b>Algorytmy Dynamiczna Tablica< /b>
Damian Szopinski 185394
Maciej Pestka 170088
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:

Chapter 3

Class Index

3.1 Class List

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

Chapter 4

File Index

4.1 File List

Here is a list of all files with brief descriptions:

Chapter 5

Namespace Documentation

5.1 AiSD Namespace Reference

Classes

- class DynamicArray
- struct noneV

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

Functions

- void Log (std::string src, std::string in)

  Ta funkcja zapisuje do pliku o sciezce src zawartosc 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. Zapisje wszystkie operacje do pliku (LogFileName "Log.txt") na wypadek crashu. Mierzy takze czas wykonywania operacji w mikrosekundach. Zwraca variant<bool,size_t,noneV (nothing)> (jako to co pierwotna funkcja Dynamic Array nam zwracala)
- auto FunctionByNO (int NO, size_t cap)

  Zwraca makro funkcji z Dynamic Array. A dokladniej std::function<variant<bool, size_t,noneV> (AiSD::DynamicArray& a,T t1,size_t i1)>
- void DistortionsSimulation (DynamicArray &arr, int t)

  Symulacja zaklocen funkcji. Wykonuje losowe funkcje dla losowych arguumentow. Mierzy czas. Zapisuje dane do pliku log, na wypadek crashu.
- void 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 mierzeniem czasu. Zmierzony czas jest w mikrosekundach i wyswietla sie na ekranie.
- void Presentation (DynamicArray &arr)

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

  Mala funkcja do otrzymania bierzacego czasu.
- std::string _timeTook (auto a, auto b)

  Zwraca dlugosc czasu pomiedzy dwoma chrono podany w string w mikrosekundach.
- void ClearLogTxt ()

  Oproznia plik Log.

5.1.1 Function Documentation

5.1.1.1 _setNow()

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

Mala funkcja do otrzymania bierzacego 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 arguumentow. Mierzy czas. Zapisuje dane do pliku log, na wypadek crashu.

Parameters

arr	Dynamic Array
t	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. Zapisje wszystkie operacje do pliku (LogFileName "Log.txt") na wypadek crashu. Mierzy takze czas wykonywania operacji w mikrosekundach. Zwraca variant<bool,size_t,noneV (nothing)> (jako to co pierwotna funkcja Dynamic Array nam zwracala)

Parameters

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

5.1.1.6 FunctionByNO()

```
auto AiSD::FunctionByNO (
            int NO,
            size_t cap )
```

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

Parameters

NO	Numer funkcji Dynamic Array od 0
cap	Rozmiar tablicy

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

src	Sciezka pliku
in	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 mierzeniem czasu. Zmierzony czas jest w mikrosekundach i wyswietla sie na ekranie.

Parameters

arr	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

arr	Dynamic Array

Chapter 6

Class Documentation

6.1 AiSD::DynamicArray Class Reference

`#include <DynamicArray.h>`

Public Member Functions

- DynamicArray ()

  Ta metoda jest konstruktorem, ktra ustawia parametry poczatkowe.
- DynamicArray (size_t rozmiar)

  Te konstuktory dziasaja jak wedlug instrukcji.
- DynamicArray (size_t rozmiar, size_t N, const T &t)
- DynamicArray (const DynamicArray &dynamicArray1)

  Te konstuktory dodatkowe z ∗ dzialaja.
- DynamicArray (DynamicArray &&dynamicArray)
- DynamicArray & operator= (const DynamicArray other)
- DynamicArray & operator= (DynamicArray &dynamicArray)
- virtual ∼DynamicArray ()

  Te destruktor usuwa dynamiczna tablice;.
- void Print ()

  Ta metoda pokazuje zawartoIJ tablicy.
- void Save ()

  Ta metoda zapisuje do pliku.
- bool IsEmpty ()

  Ta metoda dziala zgodnie jak wedlug instrukcji w czasie O(1)
- bool IsFull ()

  Ta metoda dziala zgodnie jak wedlug instrukcji w czasie O(1)
- size_t Space ()

  Ta metoda dziala zgodnie jak wedlug instrukcji w czasie O(1)
- void PushBack (T t)

  Ta metoda dziala zgodnie jak wedlug instrukcji w czasie O(1) JeIJli zakres tablicy jest przepelniony to czas jest troche inny.
- void PopBack ()

  Ta metoda dziala zgodnie jak wedlug instrukcji w czasie O(1)
- void PushFront (T t)

Ta metoda dziala zgodnie jak wedlug instrukcji w czasie O(size) Jesli zakres tablicy jest przepelniony to czas jest troche inny.

- void PopFront ()

  Ta metoda dziala zgodnie jak wedlug instrukcji w czasie O(size)
- void Insert (T t, size_t i)

  Ta metoda dziala zgodnie jak wedlug instrukcji w czasie O(size)
- void Erase (size_t i)

  Ta metoda dziala zgodnie jak wedlug instrukcji w czasie O(size)
- T & operator[ ] (size_t i)

  Ten przeciazony operator dziala tak samo jak dla tablic.
- void Clear ()

  Ta metoda (dodatkowa) dziala zgodnie jak wed^{3} *ug instrukcji w czasie O(size)*
- size_t Search (const T &t)

  Ta metoda (dodatkowa) dziala zgodnie jak wedlug instrukcji w czasie O(size)
- bool EraseFirst (const T &t)

  Ta metoda (dodatkowa) dziala zgodnie jak wedlug instrukcji w czasie O(size)
- size_t EraseAll (const T &t)

  Ta metoda (dodatkowa) dziala zgodnie jak wedlug instrukcji NIE w czasie O(size)
- size_t Erase (size_t from, size_t to)

  Ta metoda (dodatkowa) dziala zgodnie jak wedlug instrukcji w NIE JESTEM PEWNY czasie O(size)
- void Read ()

  Ten operator czyta plik z wartoIJciami tablic;.
- void PowiekszanieTablicy ()

  Ta metoda (dodatkowa) dziala poprawnie.
- T & at (size_t i)

  Ta metoda (dodatkowa) dziala zgodnie jak w isntrukcji w czasie O(1)
- void Insert (T t, size_t iloscElementow, size_t i)

  Ta metoda (doatkowa z $*$) dodaje na pozycji 'i' tyle elementw t ile jest nadanie w operacji.

Friends

- auto DoFunction (DynamicArray &, int, T, size_t)

  Ta funkcja wykonuje funkcje Dynamic Array, gdzie NO to numer funkcji od 0. Zapisje wszystkie operacje do pliku (LogFileName "Log.txt") na wypadek crashu. Mierzy takze czas wykonywania operacji w mikrosekundach. Zwraca variant<bool,size_t,noneV (nothing)> (jako to co pierwotna funkcja Dynamic Array nam zwracala)
- 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 mierzeniem czasu. Zmierzony czas jest w mikrosekundach i wyswietla sie na ekranie.

6.1.1 Constructor & Destructor Documentation

6.1.1.1 DynamicArray() [1/5]

```
AiSD::DynamicArray::DynamicArray ( )
```

Ta metoda jest konstruktorem, ktra ustawia parametry poczatkowe.

6.1.1.2 DynamicArray() [2/5]

```
AiSD::DynamicArray::DynamicArray (
            size_t rozmiar )
```

Te konstuktory dziasaja jak wedlug instrukcji.

6.1.1.3 DynamicArray() [3/5]

```
AiSD::DynamicArray::DynamicArray (
            size_t rozmiar,
            size_t N,
            const T & t )
```

6.1.1.4 DynamicArray() [4/5]

```
AiSD::DynamicArray::DynamicArray (
            const DynamicArray & dynamicArray1 )
```

Te konstuktory dodatkowe z $*$ dzialaja.

6.1.1.5 DynamicArray() [5/5]

```
AiSD::DynamicArray::DynamicArray (
            DynamicArray && dynamicArray )
```

6.1.1.6 ~DynamicArray()

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

Te destruktor usuwa dynamiczna tablice;.

6.1.2 Member Function Documentation

6.1.2.1 at()

```
T & AiSD::DynamicArray::at (
            size_t i )
```

Ta metoda (dodatkowa) dziala zgodnie jak w isntrukcji w czasie O(1)

6.1.2.2 Clear()

```
void AiSD::DynamicArray::Clear ( )
```

Ta metoda (dodatkowa) dziala zgodnie jak wed^{3}ug instrukcji w czasie O(size)

6.1.2.3 Erase() [1/2]

```
size_t AiSD::DynamicArray::Erase (
            size_t from,
            size_t to )
```

Ta metoda (dodatkowa) dziala zgodnie jak wedlug instrukcji w NIE JESTEM PEWNY czasie O(size)

6.1.2.4 Erase() [2/2]

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

Ta metoda dziala zgodnie jak wedlug instrukcji w czasie O(size)

6.1.2.5 EraseAll()

```
size_t AiSD::DynamicArray::EraseAll (
            const T & t )
```

Ta metoda (dodatkowa) dziala zgodnie jak wedlug instrukcji NIE w czasie O(size)

6.1.2.6 EraseFirst()

```
bool AiSD::DynamicArray::EraseFirst (
            const T & t )
```

Ta metoda (dodatkowa) dziala zgodnie jak wedlug instrukcji w czasie O(size)

6.1.2.7 Insert() [1/2]

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

Ta metoda dziala zgodnie jak wedlug instrukcji w czasie O(size)

6.1.2.8 Insert() [2/2]

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

Ta metoda (doatkowa z $*$) dodaje na pozycji 'i' tyle elementw t ile jest nadanie w operacji.

6.1.2.9 IsEmpty()

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

Ta metoda dziala zgodnie jak wedlug instrukcji w czasie O(1)

6.1.2.10 IsFull()

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

Ta metoda dziala zgodnie jak wedlug instrukcji w czasie O(1)

6.1.2.11 operator=() [1/2]

```
DynamicArray & AiSD::DynamicArray::operator= (
            const DynamicArray other )
```

6.1.2.12 operator=() [2/2]

```
DynamicArray & AiSD::DynamicArray::operator= (
            DynamicArray & dynamicArray )
```

6.1.2.13 operator[]()

```
T & AiSD::DynamicArray::operator[] (
            size_t i )
```

Ten przeciazony operator dziala tak samo jak dla tablic.

6.1.2.14 PopBack()

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

Ta metoda dziala zgodnie jak wedlug instrukcji w czasie O(1)

6.1.2.15 PopFront()

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

Ta metoda dziala zgodnie jak wedlug instrukcji w czasie O(size)

6.1.2.16 PowiekszanieTablicy()

```
void AiSD::DynamicArray::PowiekszanieTablicy ( )
```

Ta metoda (dodatkowa) dziala poprawnie.

6.1.2.17 Print()

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

Ta metoda pokazuje zawartoIJ tablicy.

6.1.2.18 PushBack()

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

Ta metoda dziala zgodnie jak wedlug instrukcji w czasie O(1) JeIJli zakres tablicy jest przepelniony to czas jest troche inny.

6.1.2.19 PushFront()

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

Ta metoda dziala zgodnie jak wedlug instrukcji w czasie O(size) Jesli zakres tablicy jest przepelniony to czas jest troche inny.

6.1.2.20 Read()

```
void AiSD::DynamicArray::Read ( )
```

Ten operator czyta plik z wartoIJciami tablic;.

6.1.2.21 Save()

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

Ta metoda zapisuje do pliku.

6.1.2.22 Search()

```
size_t AiSD::DynamicArray::Search (
            const T & t )
```

Ta metoda (dodatkowa) dziala zgodnie jak wedlug instrukcji w czasie O(size)

6.1.2.23 Space()

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

Ta metoda dziala zgodnie jak wedlug instrukcji w czasie O(1)

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. Zapisje wszystkie operacje do pliku (LogFileName "Log.txt") na wypadek crashu. Mierzy takze czas wykonywania operacji w mikrosekundach. Zwraca variant<bool,size_t,noneV (nothing)> (jako to co pierwotna funkcja Dynamic Array nam zwracala)

Parameters

arr	Dynamic Array
NO	Numer funkcji Dynamic Array od 0
t	Pierwszy parametr funkcji Dynamic Array
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 mierzeniem czasu. Zmierzony czas jest w mikrosekundach i wyswietla sie na ekranie.

Parameters

arr	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

6.3 Record Struct Reference

struktora zosta^{3}a utworzona i prbowa^{3}em zamiast T jź zastźpic, ale coIJ si nie uda^{3}o.

`#include <DynamicArray.h>`

Public Attributes

- std::string name
- unsigned grade

6.3.1 Detailed Description

struktora zosta[3]a utworzona i prbowa[3]em zamiast T jź zastźpic, ale coIJ si nie uda[3]o.

6.3.2 Member Data Documentation

6.3.2.1 grade

```
unsigned Record::grade
```

6.3.2.2 name

```
std::string Record::name
```

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

- include/DynamicArray.h

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 <iostream>
#include <fstream>
#include <sstream>
#include <string>
```

Classes

- class AiSD::DynamicArray
- struct Record

  struktora zosta[3] *a utworzona i prbowa*[3] *em zamiast T jź zastźpic, ale colJ si nie uda*[3] *o.*

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 //Algorytmy Dynamiczna Tablica
2 //Maciej Pestka 170088
3 //Damian Szopinski 185394
4
5 #ifndef DYNAMICARRAY_H
6 #define DYNAMICARRAY_H
7 #define T long
8
9 #include <iostream>
10 #include <fstream>
11 #include <sstream>
12 #include <string>
13
14
15 namespace AiSD{
16     class DynamicArray
17     {
18
19         public:
23             DynamicArray();
27             DynamicArray(size_t rozmiar);
28             DynamicArray(size_t rozmiar, size_t N, const T& t);
32             DynamicArray(const DynamicArray &dynamicArray1);//copyconstruktor
33             DynamicArray(DynamicArray&& dynamicArray); //move-constructor
34             DynamicArray &operator =(const DynamicArray other); //copy assignment operator
35             DynamicArray& operator =(DynamicArray& dynamicArray); //move assignment operator
39             virtual ~DynamicArray();
43             void Print();
47             void Save();
51             bool IsEmpty();
55             bool IsFull();
59             size_t Space();
64             void PushBack(T t);
68             void PopBack();
73             void PushFront(T t);
77             void PopFront();
81             void Insert(T t, size_t i);
85             void Erase(size_t i);
89             T& operator [](size_t i);
90             //Funkcje na wymagania dodatkowe
94             void Clear();
98             size_t Search(const T& t);
102             bool EraseFirst(const T& t);
106             size_t EraseAll(const T& t);
110             size_t Erase(size_t from, size_t to);
114             void Read();
118             void PowiekszanieTablicy();
122             T& at(size_t i);
123             //funkcje z gwiazdka
127             void Insert(T t, size_t iloscElementow, size_t i);
128
129         protected:
130
131         private:
132             friend auto DoFunction(DynamicArray&,int,T,size_t);
133             friend void OverflowTable(DynamicArray&);
134
135             T wczytajLiczbeCalkowita();
136             size_t capacity;
137             size_t size;
138             const std::string nazwaPliku="ZawartoscTablicy.txt";
139             T *tablica;
140
141
```

```
142     };
143 }
147 struct Record{
148     std::string name;
149     unsigned grade;
150 };
151 #endif // DYNAMICARRAY_H
```

7.5 include/TestingFunction.hpp File Reference

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

Classes

- struct AiSD::noneV

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

Namespaces

- namespace AiSD

Functions

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

  Ta funkcja zapisuje do pliku o sciezce src zawartosc 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. Zapisje wszystkie operacje do pliku (LogFileName "Log.txt") na wypadek crashu. Mierzy takze czas wykonywania operacji w mikrosekundach. Zwraca variant<bool,size_t,noneV (nothing)> (jako to co pierwotna funkcja Dynamic Array nam zwracala)
- auto AiSD::FunctionByNO (int NO, size_t cap)

  Zwraca makro funkcji z Dynamic Array. A dokladniej std::function<variant<bool, size_t,noneV> (AiSD::DynamicArray& a,T t1,size_t i1)>
- void AiSD::DistortionsSimulation (DynamicArray &arr, int t)

  Symulacja zaklocen funkcji. Wykonuje losowe funkcje dla losowych arguumentow. Mierzy czas. Zapisuje dane do pliku log, na wypadek crashu.
- 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 mierzeniem czasu. Zmierzony czas jest w mikrosekundach i wyswietla sie na ekranie.
- void AiSD::Presentation (DynamicArray &arr)

  NIESKONCZONA PETLA!!! Korzystanie z klasy Dynamic Array z poziomu wiersza polecen. (Reczne testowanie)
- auto AiSD::_setNow ()

  Mala funkcja do otrzymania bierzacego czasu.
- std::string AiSD::_timeTook (auto a, auto b)

  Zwraca dlugosc czasu pomiedzy dwoma chrono podany w string w mikrosekundach.
- void AiSD::ClearLogTxt ()

  Oproznia plik Log.

7.6 TestingFunction.hpp

Go to the documentation of this file.

```
1 //Algorytmy Dynamiczna Tablica
2 //Damian Szopinski 185394
3 //Maciej Pestka 170088
4
5 /*
6 *TESTINGFUNCTION_HPP_FILE
7 */
8 #ifndef TESTINGFUNCTION_HPP
9 #define TESTINGFUNCTION_HPP
10
11 #include "DynamicArray.h"
12
13 #include <cstddef>
14 #include <sstream>
15 #include <functional>
16 #include <chrono>
17 #include <string>
18 #include <random>
19 #include <variant>
20
21
22 namespace AiSD
23 {
24
33     void Log(std::string src,std::string in);
34
49     auto DoFunction(DynamicArray& arr,int NO,T t,size_t i);
50
59     auto FunctionByNO(int NO,size_t cap);
60
61
70     void DistortionsSimulation(DynamicArray& arr,int t);
71
81     void OverflowTable(DynamicArray& arr);
82
89     void Presentation(DynamicArray& arr);
90
94     auto _setNow();
95
99     std::string _timeTook(auto a,auto b);
100
104     void ClearLogTxt();
105
109     struct noneV{};
110 }
111
112
113
114 #endif // TESTINGFUNCTION_HPP
```