

My Project

Main PageClassesFiles

Search

File List

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

Functions.cpp

файл с описанием классов и функций

Functions.h

Заголовочный файл с описанием классов и функций

main.cpp

Main file

My Project

Main PageClassesFiles

Search

Class List

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

C Generator

создаем класс, который по ключу выбирает студ. билет какого ВУЗа сгенерировать

C HSETicket

Создаем класс, который является наследником класса Ticket

C MGTUTicket

Создаем класс, который является аследником класса Ticket

C Ticket

создаем абстрактный класс, который соержжит одну чисто виртуальнуб функцию

Generator Class Reference

Public Member Functions | Static Public Attributes | List of all members

создаем класс, который по ключу выбирает студ. билет какого ВУЗа сгенерировать More...

#include <Functions.h>

Public Member Functions

Ticket * generator (std::string lol)

Static Public Attributes

static std::unordered_map< std::string, Ticket * > data_base_ = {{"HSE", new HSETicket()}, {"MGTU", new MGTUTicket()}}

Создаем неупорядоченную mapу

Detailed Description

создаем класс, который по ключу выбирает студ. билет какого ВУЗа сгенерировать

Данный класс генерирует студ. билет

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

• Functions.h

HSETicket Class Reference

Public Member Functions | List of all members

Создаем класс, который является наследником класса Ticket. More...

#include <Functions.h>

Inheritance diagram for HSETicket:

Ticket

HSETicket

Public Member Functions

std::string generate (std::string s, int y, int m, int d) override

virtual std::string generate (std::string s, int y, int m, int d)=0

Additional Inherited Members

Protected Attributes Inherited from Ticket

Detailed Description

Создаем класс, который является наследником класса Ticket.

Данный класс генерирует студ. билет, если введенный ВУЗ - HSE

Member Function Documentation

generate()

```
std::string HSETicket::generate ( std::string s,
                                int y,
                                int m,
                                int d
                                )
```

inline override virtual

Implements Ticket.

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

- Functions.h

MGTUTicket Class Reference

Public Member Functions | List of all members

Создаем класс, который является аследником класса Ticket. More...

```
#include <Functions.h>
```

Inheritance diagram for MGTUTicket:



Public Member Functions

```
std::string generate (std::string s, int y, int m, int d) override
virtual std::string generate (std::string s, int y, int m, int d)=0
```

Additional Inherited Members

Protected Attributes inherited from Ticket

Detailed Description

Создаем класс, который является аследником класса Ticket.

Данный класс генерирует студ. билет, если введенный ВУЗ - MGTU

Member Function Documentation

generate()

```
std::string MGTUTicket::generate ( std::string s,
                                int y,
                                int m,
                                int d
                                )
```

inline override virtual

Implements Ticket.

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

- Functions.h

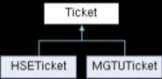
Ticket Class Reference

Public Member Functions | Protected Attributes | List of all members

создаем абстрактный класс, который соержжит одну чисто виртуальнуб функцию More...

```
#include <Functions.h>
```

Inheritance diagram for Ticket:



Public Member Functions

```
virtual std::string generate (std::string s, int y, int m, int d)=0
```

Protected Attributes

```
std::set<int> Myset
```

Detailed Description

создаем абстрактный класс, который соержжит одну чисто виртуальнуб функцию

Данный класс имеет множество, которое доступно только для наследников

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

- Functions.h

Functions.cpp File Reference

Functions

файл с описанием классов и функций More...

```
#include "Functions.h"
```

Functions

```
int summ (std::string s)
int last_num (std::string s, int n)
int Random_Number (int min, int max)
bool prov_den (int y, int m, int d)
```

Detailed Description

файл с описанием классов и функций

Данный файл содержит в себе определения основных классов, используемых в программе

Function Documentation

◆ last_num()

```
int last_num ( std::string s,
               int      n
               )
```

```
int last_num(std::string s, int n){
    long su = summ(s);
    for (int i = 0; i < n; i++){
        if ((su + (s.size() +1) * i) % n == 0){
            return i;
        }
    }
    return 0;
}
```

функция, которая вычисляет такую цифру, при которой особая сумма делится на n

◆ prov_den()

```
bool prov_den (int y,
               int m,
               int d
               )
```

```
bool prov_den(int y, int m, int d){
    if (y > 2022 || y < 1998 || m > 12 || m < 1 || d < 1 || d > 31){
        return false;
    }
    else if (y % 4 == 0 && y == 2 && d != 29){
        return false;
    }
    else if (y % 4 != 0 && y == 2 && d != 28){
        return false;
    }
    else if ((y == 1 || y == 3 || y == 5 || y == 7 || y == 8 || y == 10 || y == 12) && d != 31){
        return false;
    }
    else if (((y == 4 || y == 6 || y == 9 || y == 11) && d != 30)){
        return false;
    }
    else{
        return true;
    }
}
```

функция, которая проверяет правильность написания года, месяца и дня

♦ Random_Number()

```
int Random_Number ( int min,
                    int max
                    )
```

```
int Random_Number(int min, int max) {
    int num = min + rand() % (max - min + 1);
    return num;
}
```

функция, которая вычисляет псевдорандомное число

♦ summ()

```
int summ ( std::string s )
```

```
int summ(std::string s){
    long num = std::stol(s);
    long sum = 0;
    for (int i = 0; i < s.size(); i++){
        sum += (int(s[i]) - 48) * (i+1);
    }
    return num;
}
```

функция вычисляет особую сумму, состоящую из цифр студбилета, умноженных на их позицию в номере слева направо

Functions.h File Reference

Заголовочный файл с описанием классов и функций More...

```
#include <iostream>
#include <string>
#include <unordered_map>
#include <set>
#include <fstream>
```

Go to the source code of this file.

Classes

```
class Ticket
    создаем абстрактный класс, который соержит одну чисто виртуальную функцию More...

class HSETicket
    Создаем класс, который является наследником класса Ticket. More...

class MGTUTicket
    Создаем класс, который является аследником класса Ticket. More...

class Generator
    создаем класс, который по ключу выбирает студ. билет какого ВУЗа сгенерировать More...
```

Functions

```
int summ (std::string s)
int last_num (std::string s, int n)
bool prov_den (int y, int m, int d)
int Random_Number (int min, int max)
```

Detailed Description

Заголовочный файл с описанием классов и функций

Данный файл содержит в себе определения основных классов и функций, используемых в программе

Function Documentation

♦ last_num()

```
int last_num ( std::string s,
               int n
               )
```

создаем функцию, которая вычисляет такую цифру, при которой особая сумма делится на n

Parameters

s,n - строка, состоящая из каких-то цифр и число, на которое должна делиться особая сумма

Returns

цифру, при которой особая сумма делится на n

```
int last_num(std::string s, int n){
    long su = summ(s);
    for (int i = 0; i < n; i++){
        if ((su + (s.size() +1) * i) % n == 0){
            return i;
        }
    }
    return 0;
}
```

функция, которая вычисляет такую цифру, при которой особая сумма делится на n

♦ prov_den()

```
bool prov_den ( int y,
                int m,
                int d
                )
```

создаем функцию, которая проверяет правильность написания года, месяца и дня

Parameters

y,m,d - год, месяц, день

Returns

True, если удовлетворяет всем условиям, иначе False

```
bool prov_den(int y, int m, int d){
    if (y > 2022 || y < 1998 || m > 12 || m < 1 || d < 1 || d > 31){
        return false;
    }
    else if (y % 4 == 0 && y == 2 && d != 29){
        return false;
    }
    else if (y % 4 != 0 && y == 2 && d != 28){
        return false;
    }
    else if ((y == 1 || y == 3 || y == 5 || y == 7 || y == 8 || y == 10 || y == 12) && d != 31){
        return false;
    }
    else if (((y == 4 || y == 6 || y == 9 || y == 11) && d != 30)){
        return false;
    }
    else{
        return true;
    }
}
```

функция, которая проверяет правильность написания года, месяца и дня

♦ Random_Number()

```
int Random_Number ( int min,
                    int max
                    )
```

создаем функцию, которая вычисляет псевдорандомное число

Parameters

min,max - границы выбора числа

Returns

псевдорандомное число

```
int Random_Number(int min, int max) {
    int num = min + rand() % (max - min + 1);
    return num;
}
```

функция, которая вычисляет псевдорандомное число

♦ summ()

```
int summ ( std::string s )
```

создаем функция, которая вычисляет особую сумму, состоящую из цифр студбилета, умноженных на их позицию в номере слева направо

Parameters

s - строка, состоящая из каких-то цифр

Returns

сумму всех цифр в строке

```
int summ(std::string s){
    long num = std::stol(s);
    long sum = 0;
    for (int i = 0; i < s.size(); i++){
        sum += (int(s[i]) - 48) * (i+1);
    }
    return num;
}
```

функция вычисляет особую сумму, состоящую из цифр студбилета, умноженных на их позицию в номере слева направо

main.cpp File Reference

main file More...

```
#include "Functions.h"
```

Functions

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

Detailed Description

main file

Function Documentation

Если нет флага

```
if (argc == 1){
    std::cerr << "Enter the '--tofile' or '--fromfile'";
    return 0;
}
```

Если флагов больше двух или первый флаг не равен и "tofile"

```
if ((strcmp(argv[1], "--tofile") == 1 && strcmp(argv[1], "--fromfile") == 1) || argc > 3) {
    std::cerr << "Wrong command, try again";
    return 0;
}
```

Если флаг один и равен "--tofile"

```
if ((strcmp(argv[1], "--tofile") == 0) && argc == 2){
    std::ofstream output;
    output.open("laba6-7_output.txt");
    if (output.is_open()){
        int n;
        std::cout << "Введите количество студенческих, которые хотите сгенерировать: ";
        std::cin >> n;
        for(int i = 0; i < n; i++){
            output << "Введите ВУЗ: ";
            std::string s;
            std::cin >> s;
            output << "Введите пол, дату рождения через пробел: ";
            std::string a;
            int y, m, d;
            std::cin >> a >> y >> m >> d;
            Generator templategenerator;
            std::string B_1 = templategenerator.generator(s)->generate(a, y, m, d);
            output << "Сгенерированный студенческий: "<< B_1 << std::endl;
        }
    }
    output.close();
}
```

```
int main(int argc, char **argv) {
    if (argc == 1){
        std::cerr << "Enter the '--tofile' or '--fromfile'";
        return 0;
    }

    if ((strcmp(argv[1], "--tofile") == 1 && strcmp(argv[1], "--fromfile") ==
1) || argc > 3) {
        std::cerr << "Wrong command, try again";
        return 0;
    }

    if ((strcmp(argv[1], "--tofile") == 0) && argc == 2){
        std::ofstream output;///создаем объект класса ofstream
        output.open("laba6-7_output.txt");///связываем объект с файлом
        if (output.is_open()){
            int n;///количество студенческих билетов
            std::cout << "Введите количество студенческих, которые хотите
сгенерировать: ";
            std::cin >> n;
            for(int i = 0; i < n; i++){
                output << "Введите ВУЗ: ";
                std::string s;///пол студента
                std::cin >> s;
                output << "Введите пол, дату рождения через пробел: ";
                std::string a;
                int y, m, d;///дата рождения
                std::cin >> a >> y >> m >> d;
                Generator templategenerator;
                std::string B_1 = templategenerator.generator(s)->generate(a,
y, m, d);///генерируем студенческих билет
                output << "Сгенерированный студенческий: "<< B_1 <<
std::endl;
            }
        }
        output.close();
    }
}
```

Если флаг один и равен "--fromfile"

```
if ((strcmp(argv[1], "--fromfile") == 0) && argc == 2){
    std::ifstream input;
    input.open("laba6-7_input.txt");
    if (input.is_open()){
        int n;
        std::cout << "Введите количество студенческих, которые хотите сгенерировать: ";
        input >> n;
        for(int i = 0; i < n; i++){
            std::cout << "Введите ВУЗ: ";
            std::string s;
            input >> s;
            std::cout << "Введите пол, дату рождения через пробел: ";
            std::string a;
            int y, m, d;
            input >> a >> y >> m >> d;
            Generator templategenerator;
            std::string B_1 = templategenerator.generator(s)->generate(a, y, m, d);
            std::cout << "Сгенерированный студенческий: " << B_1 << std::endl;
        }
    }
    input.close();
}
```

```
if ((strcmp(argv[1], "--fromfile") == 0) && argc == 2){
    std::ifstream input;///создаем объект класса ifstream
    input.open("laba6-7_input.txt");///открываем файл
    if (input.is_open()){
        int n;///количество студенческих билетов
        std::cout << "Введите количество студенческих, которые хотите
сгенерировать: ";
        input >> n;
        for(int i = 0; i < n; i++){
            std::cout << "Введите ВУЗ: ";
            std::string s;
            input >> s;
            std::cout << "Введите пол, дату рождения через пробел: ";
            std::string a;///пол студента
            int y, m, d;///дата рождения
            input >> a >> y >> m >> d;
            Generator templategenerator;
            std::string B_1 = templategenerator.generator(s)->generate(a, y,
m, d);
            std::cout << "Сгенерированный студенческий: " << B_1 << std::endl;
        }
    }
    input.close();
}
```

Если флагов два : "tofile" и "fromfile"

```
if ((strcmp(argv[1], "--fromfile") == 0) && (strcmp(argv[2], "--tofile") == 0)){
    std::ifstream input;
    std::ofstream output;
    input.open("laba6-7_input.txt");
    output.open("laba6-7_output.txt");
    if (input.is_open() && output.is_open()){
        int n;
        output << "Введите количество студенческих, которые хотите сгенерировать: ";
        input >> n;
        for(int i = 0; i < n; i++){
            output << "Введите ВУЗ: ";
            std::string s;
            input >> s;
            output << "Введите пол, дату рождения через пробел: ";
            std::string a;
            int y, m, d;
            input >> a >> y >> m >> d;
            Generator templategenerator;
            std::string B_1 = templategenerator.generator(s)->generate(a, y, m, d);
            output << "Сгенерированный студенческий: " << B_1 << std::endl;
        }
    }
    input.close();
    output.close();
}
```

```
if ((strcmp(argv[1], "--fromfile") == 0) && (strcmp(argv[2], "--tofile") ==
0)){
    std::ifstream input;
    std::ofstream output;
    input.open("laba6-7_input.txt");
    output.open("laba6-7_output.txt");
    if (input.is_open() && output.is_open()){
        int n;
        output << "Введите количество студенческих, которые хотите
сгенерировать: ";
        input >> n;
        for(int i = 0; i < n; i++){
            output << "Введите ВУЗ: ";
            std::string s;
            input >> s;
```

```
        output << "Введите пол, дату рождения через пробел: ";
        std::string a;
        int y, m, d;
        input >> a >> y >> m >> d;
        Generator templategenerator;
        std::string B_1 = templategenerator.generator(s)->generate(a, y,
m, d);
        output << "Сгенерированный студенческий: " << B_1 << std::endl;
    }
}
input.close();
output.close();
}
```