ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО
ОБРАЗОВАНИЯ «НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ «ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

Московский институт электроники и математики им. А. Н. Тихонова

Отчет по Лабороторной работе №7 по предмету «Языки Программирования»

Подготовил: студент группы СКБ221
Нугманов М. И.

Москва 2022

Оглавление

Table of contents

Иерархический список классов

Иерархия классов

Иерархия классов.

Алфавитный указатель классов

Классы

Классы с их кратким описанием.

Список файлов

Файлы

Полный список файлов.

Классы

Класс GenTickets

Класс, втором реализован шаблонный метод
`#include <classes.h>`

Открытые члены

- **GenTickets** ()
- **PassTicketGenerator** * **generate** (std::string, std::string, int, int, int)
  *Шаблонный метод класса **GenTickets** для общего доступа к генерации билетов*

Подробное описание

Класс, втором реализован шаблонный метод

См. определение в файле **classes.h** строка **47**

Конструктор(ы)

GenTickets::GenTickets () `[inline]`

См. определение в файле **classes.h** строка **49**

Методы

PassTicketGenerator * GenTickets::generate (std::string *name*, std::string *sex*, int *year*, int *month*, int *day*)

Шаблонный метод класса **GenTickets** для общего доступа к генерации билетов

См. определение в файле **classes.cpp** строка **65**

Объявления и описания членов классов находятся в файлах:

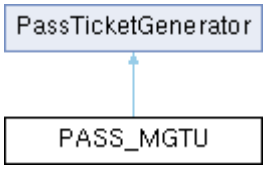
- **classes.h**
- **classes.cpp**

Класс PASS_MGTU

Класс, производный от абстрактного, в котором реализована генерация билета для МГТУ

`#include <classes.h>`

Граф наследования:PASS_MGTU:



Открытые члены

- **PASS_MGTU** (std::string **sex**, int **year**, int **month**, int **day**)
- std::string **Generator** () override final
  Метод класса МГТУ для генерации билета

Дополнительные унаследованные члены

Подробное описание

Класс, производный от абстрактного, в котором реализована генерация билета для МГТУ

См. определение в файле **classes.h** строка **34**

Конструктор(ы)

PASS_MGTU::PASS_MGTU (std::string *sex*, int *year*, int *month*, int *day*) `[inline]`

См. определение в файле **classes.h** строка **36**

Методы

std::string PASS_MGTU::Generator () `[final]`, `[override]`, `[virtual]`

Метод класса МГТУ для генерации билета

Замещает **PassTicketGenerator** (*стр.8*).

См. определение в файле **classes.cpp** строка **38**

Объявления и описания членов классов находятся в файлах:

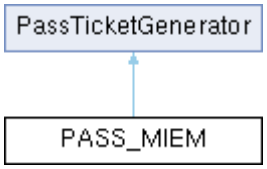
- **classes.h**
- **classes.cpp**

Класс PASS_MIEM

Класс, производный от абстрактного, в котором реализована генерация билета для МИЕМ

`#include <classes.h>`

Граф наследования:PASS_MIEM:



Открытые члены

- **PASS_MIEM** (std::string **sex**, int **year**, int **month**, int **day**)
- std::string **Generator** () override final
  Метод класса МИЕМ для генерации билета

Дополнительные унаследованные члены

Подробное описание

Класс, производный от абстрактного, в котором реализована генерация билета для МИЕМ

См. определение в файле **classes.h** строка **21**

Конструктор(ы)

PASS_MIEM::PASS_MIEM (std::string *sex*, int *year*, int *month*, int *day*) `[inline]`

См. определение в файле **classes.h** строка **23**

Методы

std::string PASS_MIEM::Generator () `[final]`, `[override]`, `[virtual]`

Метод класса МИЕМ для генерации билета

Замещает **PassTicketGenerator** (*стр.8*).

См. определение в файле **classes.cpp** строка **11**

Объявления и описания членов классов находятся в файлах:

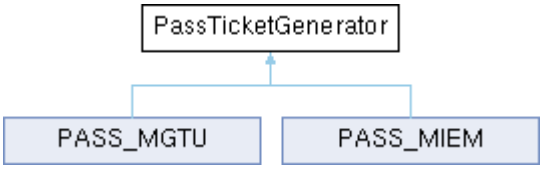
- **classes.h**
- **classes.cpp**

Класс PassTicketGenerator

Абстрактный класс, в котором есть чисто виртуальный метод генерации билета

`#include <classes.h>`

Граф наследования:PassTicketGenerator:



Открытые члены

- virtual std::string **Generator** ()=0

Защищенные данные

- int **sex**
- int **year**
- int **month**
- int **day**
- std::string **passticket** = ""

Подробное описание

Абстрактный класс, в котором есть чисто виртуальный метод генерации билета

См. определение в файле **classes.h** строка **8**

Методы

virtual std::string PassTicketGenerator::Generator () `[pure virtual]`

Замещается в **PASS_MIEM** (*cmp.7*) и **PASS_MGTU** (*cmp.6*).

Данные класса

int PassTicketGenerator::day `[protected]`

См. определение в файле **classes.h** строка **13**

int PassTicketGenerator::month `[protected]`

См. определение в файле **classes.h** строка **12**

std::string PassTicketGenerator::passticket = "" `[protected]`

См. определение в файле **classes.h** строка **14**

int PassTicketGenerator::sex [protected]

См. определение в файле **classes.h** строка **10**

int PassTicketGenerator::year [protected]

См. определение в файле **classes.h** строка **11**

Объявления и описания членов класса находятся в файле:

- **classes.h**

Файлы

Файл classes.cpp

Файл, в котором описаны все функции, реализующие генерацию билета

```
#include <random>
#include <string>
#include <chrono>
#include "classes.h"
```

Подробное описание

Файл, в котором описаны все функции, реализующие генерацию билета

См. определение в файле **classes.cpp**

classes.cpp

См. документацию.
```
00001
00004 #include <random>
00005 #include <string>
00006 #include <chrono>
00007 #include "classes.h"
00011     std::string PASS_MIEM::Generator() {
00012         int date = year * 10000 + month * 100 + day;
00013         std::string passticket = std::to_string(sex) + std::to_string(date);
00014         std::mt19937 randomizer(date +
std::chrono::steady_clock::now().time_since_epoch().count());
00015         std::uniform_int_distribution<int> NNNNN(10000, 99999);
00016         passticket += std::to_string(NNNNN(randomizer));
00017         int sum = 0;
00018         for(int i = 0; i < passticket.size(); i++) {
00019             sum += (passticket[i] - '0') * (i + 1);
00020         }
00021         while(sum % 11 == 4) {
00022             sum -= (passticket[9] - '0') * 10;
00023             int new_digit = (passticket[9] - '0' + 1) % 10;
00024             passticket[9] = new_digit + '0';
00025             sum += new_digit * 10;
00026         }
00027         for(int i = 0; i < 10; i++) {
00028             if((sum + i * 15) % 11 == 0) {
00029                 passticket += std::to_string(i);
00030                 break;
00031             }
00032         }
00033         return passticket;
00034     }
00038 std::string PASS_MGTU::Generator() {
00039         int date = year * 10000 + month * 100 + day;
00040         std::string passticket = std::to_string(sex) + std::to_string(date);
00041         std::mt19937 randomizer(date +
std::chrono::steady_clock::now().time_since_epoch().count());
00042         std::uniform_int_distribution<int> NNNN(1000, 9999);
00043         passticket += std::to_string(NNNN(randomizer));
00044         int sum = 0;
00045         for(int i = 0; i < passticket.size(); i++) {
00046             sum += (passticket[i] - '0') * (i + 1);
00047         }
00048         while((sum % 10) % 2 != 0) {
00049             sum -= (passticket[10] - '0') * 11;
00050             int new digit = (passticket[10] - '0' + 1) % 10;
00051             passticket[10] = new_digit + '0';
00052             sum += new_digit * 11;
00053         }
00054         for(int i = 0; i < 10; i++) {
00055             if((sum + i * 14) % 10 == 0) {
00056                 passticket += std::to string(i);
00057                 break;
00058             }
00059         }
00060         return passticket;
00061     }
00065 PassTicketGenerator* GenTickets::generate(std::string name, std::string sex, int
year, int month, int day) {
00066     if (name == "MIEM")
00067     {
00068         PASS_MIEM* univer = new PASS_MIEM(sex, year, month, day);
00069         return univer;
00070     }
00071     else
00072     {
00073         PASS_MGTU* univer = new PASS_MGTU(sex,year,month,day);
00074         return univer;
00075     }
00076
00077 }
```

Файл classes.h

Заголовочный файл, в котором описаны классы и функции для реализации программы
`#include <string>`

Классы

- class **PassTicketGenerator**
  Абстрактный класс, в котором есть чисто виртуальный метод генерации билета
- class **PASS_MIEM**
  Класс, производный от абстрактного, в котором реализована генерация билета для МИЕМ
- class **PASS_MGTU**
  Класс, производный от абстрактного, в котором реализована генерация билета для МГТУ
- class **GenTickets**
  Класс, втором реализован шаблонный метод

Подробное описание

Заголовочный файл, в котором описаны классы и функции для реализации программы

См. определение в файле **classes.h**

classes.h

```
См. документацию.00001
00004 #include <string>
00008 class PassTicketGenerator {
00009 protected:
00010     int sex;
00011     int year;
00012     int month;
00013     int day;
00014     std::string passticket = "";
00015 public:
00016     virtual std::string Generator() = 0;
00017 };
00021 class PASS_MIEM : public PassTicketGenerator {
00022 public:
00023     PASS_MIEM(std::string sex, int year, int month, int day) {
00024         this->sex = (sex == "man") ? 8 : 4;
00025         this->year = year;
00026         this->month = month;
00027         this->day = day;
00028     }
00029     std::string Generator() override final;
00030 };
00034 class PASS_MGTU : public PassTicketGenerator {
00035 public:
00036     PASS_MGTU(std::string sex, int year, int month, int day) {
00037         this->sex = (sex == "man") ? 1 : 0;
00038         this->year = year;
00039         this->month = month;
00040         this->day = day;
00041     }
00042     std::string Generator() override final;
00043 };
00047 class GenTickets {
00048 public:
00049     GenTickets(){}
00050     PassTicketGenerator* generate(std::string, std::string, int, int, int);
00051 };
```

Файл main.cpp

Главная функция, в который реализован ввод данных с консоли

```
#include <iostream>
#include <string>
#include "classes.h"
```

Функции

- int **main** ()

Подробное описание

Главная функция, в который реализован ввод данных с консоли

См. определение в файле **main.cpp**

Функции

int main ()

См. определение в файле **main.cpp** строка **8**

main.cpp

См. документацию.
```
00001
00004 #include <iostream>
00005 #include <string>
00006 #include "classes.h"
00007
00008 int main() {
00009     GenTickets ticket;
00010     int num;
00011     std::cout << "Enter the number of tickets: ";
00012     std::cin >> num;
00013     std::cout << "\n";
00014     for(int i = 0; i < num; i++) {
00015         std::cout << i+1 << ") ";
00016         std::string name, sex;
00017         int year, month, day;
00018         std::cin >> name >> sex >> year >> month >> day;
00019         std::cout << "   RESULT: " << ticket.generate(name, sex, year, month,
day)->Generator() << std::endl;
00020     }
00021     return 0;
00022 }
```

Алфавитный указатель

INDEX