

# Лабораторная работа 7

## Как работать с программой

Пользователь может ввести два флага: --fromfile(чтение из файла) и --tofile(запись в файл). После флагов нужно указывать имя соответствующих файлов. Например: --fromfile input.txt --tofile output.txt.

Пользователь вводит данные в следующем порядке: Название ВУЗа, пол, год рождения, месяц рождения, день рождения и получает сгенерированный номер студ. билета.

## Файлы

- main.cpp
- funcs.cpp
- student\_id.h
- student\_id.cpp
- miem.h
- miem.cpp
- mgtuu.h
- mgtuu.cpp
- template\_generator.h
- template\_generator.cpp

## main.cpp

```
#include <iostream>
#include <cstring>
#include <fstream>
#include "student_id.h"
#include "miem.h"
#include "mgtuu.h"
#include "template_generator.h"
#include "funcs.cpp"
```

В файле обрабатываются флаги, генерируется билет и выводится в файл\на консоль.

Вызов итоговой функции:

```
Template_Generator template_generator;
std::string sd = template_generator.generator(data[0], data[1], stoi(data[2]),
stoi(data[3]), stoi(data[4]))->get_id();
```

## funcs.cpp

В файле реализована функция для чтения из файла\из консоли

```
string* read(std::istream& in, bool flag_file, string fromfile_name)
```

Параметры функции:

std::istream& in	Ссылка на поток
bool flag_file	Флаг для файла. Если был введен флаг --fromfile, то в функцию передается флаг со значением True, иначе False.
string fromfile_name	Название файла для чтения. Если чтение с консоли, то передается пустая строка.

## student\_id.h, student\_id.cpp

В файлах описан класс Student\_ID и реализованы его методы.

```
class Student_ID
{
public:
    virtual std::string get_sex_for_id() = 0;

    void set_sex(std::string);
    void set_year(int);
    void set_month(int);
    void set_day(int);
    void set_id(std::string);

    std::string get_id();

    std::string get_year_for_id();
    std::string get_month_for_id();
    std::string get_day_for_id();

    virtual std::string get_name() = 0;
    std::string get_sex();
    int get_year();
    int get_month();
    int get_day();
protected:
    std::string id;
    std::string sex;
    int year;
    int month;
    int day;
};
```

Student\_ID – базовый класс, от которого наследуются классы MIEM и MGTUU

## miem.h, miem.cpp

В файлах описан класс MIEM и реализованы его методы.

```
class MIEM : public Student_ID
{
public:
    std::string get_name() override;
    std::string get_sex_for_id() override;
};
```

## mgtuu.h, mgtuu.cpp

В файлах описан класс MGTUU и реализованы его методы.

```
class MGTUU : public Student_ID
{
public:
    std::string get_name() override;
    std::string get_sex_for_id() override;
};
```

## template\_generator.h, template\_generator.cpp

```
class Template_Generator
{
public:
    Student_ID* generator(std::string, std::string, int, int, int);
protected:
    std::string generate_date(Student_ID*);
    std::string generate_random(Student_ID*);
};
```

Метод generator принимает на вход название университета, пол, дату рождения

```
Student_ID* Template_Generator :: generator(std::string university_name,
std::string sex, int year, int month, int day)
```

На основе этих данных создается объект соответствующего класса, куда записываются данные о студ.билете. Генерируется номер студ билета и также записывается в класс MIEM или MGTUU. Метод возвращает ссылку на объект класса MIEM или MGTUU.