

# ОТЧЕТ по лабораторной работе№67

## Что делает программа?

Программа в зависимости от значения флага записывает студенческие билеты в файл `out.txt` или в нужный вам файл при использовании соответствующего флага .

## Форматы ввода

flag	correct/incorrect	Error
--tofile	cottect	-
--fromfile "filename.txt"	cottect	-
--tofile --fromfile "filename.txt"	cottect	-
else flags	incorrect	Error
nothing	incorrect	Error

## О реализации

Если введен флаг `--tofile` программа будет ожидать ввода от пользователя до тех пор, пока не будет введен `'break'` и затем запишет результаты в файл. Если введен флаг `--fromfile` и название файла, который надо обработать. Программа приступит к действию, пока не достигнет конца файла.

Сама реализация основана на посимвольном вводе из файла или из консоли и считывании из файла:

```
cout << "Enter the sex of student: ";
cin >>sex;
cout << "Enter the university of student: ";
cin >> university;
cout <<"Enter the year of student: ";
cin >> year;
cout <<"Enter the month of stufent: ";
cin >> month;
cout << "Enter the day of student: ";
cin >> day;
```

```
ofstream fout(filename, std::ios_base::app);
```

Программа проверяет введенные данные при помощи отдельной функции и выводит ошибки:

```
int CheckDate(string sex, string university, int year, int month, int day)
{
    if (sex != "woman" && sex!="man"){
        return 0;
    }
    if (university != "MIEM" && university != "MGITU"){
        return 0;
    }
    if (year < 1000 || year > 2022)
        return 0;

    switch(month)
    {
        case 1:
        case 3:
        case 5:
        case 7:
        case 8:
        case 10:
        case 12:
            return IsRange(day, 1, 31);
        case 2:
            if (year % 4 == 0 && (year % 100 != 0 || year % 400 == 0)) return IsRange(day, 1, 29);
            else return IsRange(day, 1, 28);
        case 4:
        case 6:
        case 9:
        case 11:
            return IsRange(day, 1, 30);
        default:
            return 0;
    }
}
```

Основа алгоритма лежит в классовом устройстве файла:

```
class Interface{
public:
    virtual string generate(string sex, int year, int month, int day) = 0;
};

class Template: public Interface{
protected:
    string generate(string s, int year, int month, int day) final{
        set_sex(s);
        Data(year,month,day);
        set_N();
        RandomNumber();
        set_C();
    }
};
```

```

return result;
}

string result, sex, year, month,day, C;

int N, num, k;

virtual void set_sex(string s) = 0;

void Data(int y, int m, int d){
    y = y * 10000 + m * 100 + d;
    year = to_string(y);
    result+=year;
}
virtual void set_N() = 0;
void RandomNumber(){
    std::mt19937 gen(rand());
    std::uniform_int_distribution<> dis(1, 100000);
    string x = to_string(dis(gen));
    if (x.size() < N){
        string ss(N - x.size(), '0');
        x = ss + x;
    }
    else if(x.size() > N){
        x = string(x,1,N);
    }
    result+=x;
};
void set_C(){
    int sum = 0, flag = 1;
    for(int i =0; i<result.size();i++)
    {
        sum+=(result[i]- '0') * (i + 1);
    }
    for (int i = 0; i<10; i++)
    {
        if ((sum + i*k) % num ==0){
            result+=to_string(i);
            flag = 1;
            break;
        }
        else flag = 0;
    }
    if (flag == 0){
        result = string(result,0,9);
        RandomNumber();
        set_C();}
}

};

class MIEM: public Template{

```

```

public:
    void set_sex(string s) override{
        if(s == "woman"){
            result+='4';
        }
        else result+='8';
    }
    void set_N() override{
        N=5;
        num = 11;
        k = 15;
    }
};

class MGTUU: public Template{
public:
    void set_sex(string s) override{
        if(s == "woman"){
            result+='1';
        }
        else result+='2';
    }
    void set_N() override{
        N=4;
        num = 10;
        k = 14;
    }
};

class template_pattern_student_number_generator {
public:
    Interface* generator(string str){
        if (str == "MIEM"){
            Interface* student = new MIEM;
            return student;
        }
        else if (str == "MGTUU"){
            Interface* student = new MGTUU;
            return student;
        }
        return 0;
    }
};

```

Как видите, я использовал интерфейсный класс и реализовал шаблонный метод в производном классе-наследнике.

## Преимущества и ошибки

- Программа общается с пользователем.
- Если вы допустили ошибку, программа об этом сообщит. То есть она приближена к универсальной.

- Флаги записываются в строгом порядке
  - Строгий порядок ввода пол->вуз->год->месяц->день
-