

Отчёт. Лабораторная работа №6-7.

Программа, поделена на три файла:

- файл c.cpp
- файл func.cpp
- файл lab67.h

Файл c.cpp

`#include <...>` - директива препроцессора, которая указывает препроцессору включить содержимое указанного файла в точку, где отображается директива

Подключены заголовочные файлы для организации ввода-вывода (`<iostream>`, `<fstream>`), заголовочный файл `"lab67.h"`, содержащий объявления функций из `func.cpp`. `<cstring>` для того, чтобы сравнить введенный пользователем флаги с необходимыми. `<set>` для работы с множеством.

Using – объявление, позволяющее сделать видимыми члены пространства имён в том месте, где это объявление написано. В данном случае я использовала `using`, чтобы сделать видимыми `cout`, `cin`, `endl`, `string` и `cerr`.

В `c.cpp` содержится функция `int main()` с параметрами `argc` и `argv[]`. При запуске пользователь может ввести флаги `--tofile`, `--fromfile`, а также указать файл, с которого будут считываться данные, и файл, куда будет записан полученный результат. Предусмотрен некорректный ввод флагов, а именно:

- чрезмерное количество флагов

```
else if (argc > 5) {  
    cerr << "ERROR! Too many flags" << endl;  
    return -1;  
}
```

- некорректное имя флага

```
else {  
    cerr << "ERROR! Flags are not correct" << endl;  
    return -1;  
}
```

В случае ошибки программа завершается и на экран выводится фраза с объяснением ошибки.

Если все флаги корректны или флаги вовсе не были введены, по введенным данным создадутся два студбилета (для МИЭМА и МГТУУ).

Чтение из файла и с консоли, а также вывод аналогичны

```
cin >> n;  
for (int i = 1; i <= n; i++)  
{  
    cin >> university >> sex >> year >> month >> day;  
    if (correct_data(university, sex, year, month, day) == 0)  
    {  
        unite tp;  
        string answer = tp.generator(university)->generate(sex, year, month, day);  
        usedNum.insert(answer);  
        cout << answer << endl;  
    }  
}
```

```

        else {
            cerr << "You did a mistake. I can't print anything" << endl;
        }
    }
}

```

Также в с.сpp пять классов, а именно:

- виртуальный класс-интерфейс `interface`, который содержит в себе одну чисто виртуальную функцию, которая после будет определена классе-генераторе общего вида

```

class interface {
public:
    virtual string generate(char s, int y, int m, int d) = 0;
};

```

- класс-генератор общего вида, написанный с использованием паттерна Шаблонный метод.

```

class tempate_pattern_generator : public interface {
public:
protected:
    string generate(char s, int y, int m, int d) final {
        base_generation();
        gen_sex(s);
        gen_date(y, m, d);
        gen_NC();
        return take_();
    }

protected:
    string result;
    int ferr = 0;
    string sex, year, month, day, N;
    int num, C;
    ~tempate_pattern_generator() {};
    virtual void base_generation() final { //метод базовой генерации
        sex = "";
        year = "";
        month = "";
        day = "";
        N = "";
        C = -1;
    }
    virtual void gen_sex(char s) {}; //метод, генерирующий первой цифры студбилета
    virtual void gen_date(int y, int m, int d) final { //метод, считывающий дату
                                                        //и преобразовывающий дату
        if (y > 1000 && y < 2023) {
            year = to_str(y);
            result += year;
        }
        if (m > 0 && m < 13) {
            if (m < 10) { month = "0" + to_str(m); }
            else
            {
                month = to_str(m);
            }
            result += month;
        }
        if (m == 1 || m == 3 || m == 5 || m == 7 || m == 8 || m == 10 || m == 12) {
            if (d > 0 && d < 32) {
                if (d < 10) { day = "0" + to_str(d); }
                else
                {
                    day = to_str(d);
                }
                result += day;
            }
        }
    }
}

```

```

    }
    else if (m == 2)
    {
        if (d > 0 && d < 29) {
            if (d < 10) { day = "0" + to_str(d); }
            else
            {
                day = to_str(d);
            }
            result += day;
        }
    }
    else if (m == 4 || m == 6 || m == 9 || m == 11) {
        if (d > 0 && d < 31) {
            if (d < 10) { day = "0" + to_str(d); }
            else
            {
                day = to_str(d);
            }
            result += day;
        }
    }
};
virtual void gen_NC() {}; //метод генерации N и C

virtual string take_() final { //метод вывода результата
    return result;
}
};

```

- класс-генератор студбилета МИЭМА. В нем переопределены некоторые методы из базового класса.

```

class generator_MEMA : public tempate_pattern_generator {
protected:
    ~generator_MEMA() {};
    void gen_sex(char s) override {
        if (s == 'M' || s == 'm') { sex = "8"; result += sex; }
        else if (s == 'F' || s == 'f') { sex = "4"; result += sex; }
    }

    void gen_NC() override {
        num = 11;
        while (C == -1) {
            N = to_str(randomnum(0, 99999));
            if (N.size() < 5) {
                string sn(5 - N.size(), '0');
                N = sn + N;
            }
            C = ost(result, N, num);
            if (C > -1) {
                if (usedNum.find(result + N + to_str(C)) == usedNum.end())
                {
                    result += N + to_str(C);
                    usedNum.insert(result); //usedNum - set
                }
                else { C = -1; }
            }
        }
    }
};

```

- класс-генератор номера студбилета МГТУУ аналогичен предыдущему классу, есть лишь незначительные различия

```

if (s == 'M') { sex = 2; } //другое значение первой цифры билета
else if (s == 'W') { sex = 1; }

```

```
N = to_str(randomnum(0, 9999)); //y N 4 знака, а не 5
```

- класс, который создает указатель на класс generator_MEM, если введен “MIEM”. Если “MGTUU” - указатель на класс generator_MGTUU.

```
class unite {
public:
    interface* generator(string str) {
        if (str == "MIEM") {
            interface* st1 = new generator_MEMA;
            return st1;
        }
        else if (str == "MGTUU") {
            interface* st2 = new generator_MGTUU;
            return st2;
        }
    }
};
```

Файл func.cpp

Некоторые функции, которые используются для проверки значений, перевода из одного типа в другой и генерации ПСЧ.

В этом файле дополнительно подключен файл `<random>`, который дает возможность более удачного сгенерировать псевдослучайное число, чем `rand()`.

int randomnum(int min, int max)

Данная функция выводит псевдослучайное число

```
int randomnum(int min, int max) {
    std::random_device rd;
    std::mt19937 num(rd());
    std::uniform_int_distribution<int> distrib(min, max);
    return distrib(num);
}
```

`std::random_device` генерирует сид для генератора чисел на основе вихря Мерсенна

`std::mt19937 num(rd());` вихрь Мерсенна, который генерирует ПСЧ

`std::uniform_int_distribution<>` задает диапазон

int correct_data(string un, char s, int y, int m, int d)

Проверяет корректность введенных данных

```
if (un != "MIEM" && un != "MGTUU") {
    cerr << "Incorrect univeristy entry." << endl;
    return -1;
}
if (s != 'M' && s != 'm' && s != 'F' && s != 'f') {
    cerr << "..." << endl;
    return -1;
}
if (y < 1000 || y > 2023) {
    cerr << "..." << endl;
    return -1;
}
if (m < 1 || m > 12) {
    cerr << "..." << endl;
    return -1;
}
if (m == 1 || m == 3 || m == 5 || m == 7 || m == 8 || m == 10 || m == 12) {
    if (d < 1 || d > 31) {
        cerr << "..." << endl;
        return -1;
    }
}
```

```

    }
else if (m == 2)
{
    if (d < 1 || d > 28) {
        cerr << "..." << endl;
        return -1;
    }
}
else if (m == 4 || m == 6 || m == 9 || m == 11) {
    if (d < 1 || d > 30) {
        cerr << "..." << endl;
        return -1;
    }
}
return 0;
}

```

int ost(string result, string N, int num)

Эта функция вначале подсчитывает сумму цифр номер студбилета (за исключением последней цифры), умноженных на свой порядковый номер, а после находит такое число $C \in [0,10)$, что сумма цифр была кратна введенному num.

```

int ost(string result, string N, int num) {
    int ans, sum_of_num_mult_pos;
    ans = -1;
    int j = 0;
    sum_of_num_mult_pos = 0;
    for (int i = 0; i < result.size(); i++)
    {
        sum_of_num_mult_pos += to_int(result[i]) * (i + 1);
        j = i + 1;
    }
    j += 1;
    for (int i = 0; i < N.size(); i++){
        sum_of_num_mult_pos += to_int(N[i]) * j;
        j += 1;
    }
    for (int i = 0; i < 10; i++) {
        if (num == 11) {
            ans = (sum_of_num_mult_pos + i * 15) % 11 == 0 ? i : -1;
        }
        else if (num == 10) {
            ans = (sum_of_num_mult_pos + i * 14) % 10 == 0 ? i : -1;
        }
    }
    return ans;
}

```

Возвращает -1, если нужное значение не было найдено.

string to_str(int x)

Преобразует число в строку с помощью потока для чтения ostream

```

string to_str(int x) {
    std::ostringstream str;
    str << x;
    return str.str();
}

```

int to_int(char str)

переводит символ в цифру с помощью оператора switch

```

int to_int(char str) {
    switch (str) {
        case '0':
            return 0;
        case '1':
            return 1;
    }
}

```

.....

Файл lab67.h

```

#ifndef LAB67
#define LAB67
int ost(std::string result, std::string N, int num);
std::string to_str(int x);
int to_int(char str);
int randomnum(int min, int max);
int correct_data(std::string un, char s, int y, int m, int d);
#endif

```

В заголовочном файле перечислены функции, используемые в func.cpp и c.cpp. Также реализована защита от переопределения.

Makefile

Мейк-файл создан для компиляции программы. Реализованы методы clean и distclean.

```

TR = main
CC = g++
OB = func.o c.o
$(TR) : $(OB)
    $(CC) $(OB) -o $(TR)

func.o : func.cpp
    $(CC) -c func.cpp -o func.o

c.o : c.cpp
    $(CC) -c c.cpp -o c.o

clean :
    rm $(OB)

distclean : clean
    rm $(TR) *.txt Makefile

```

clean удаляет файлы с расширением .o, а distclean удаляет все файлы, в том числе и мейкфайл.

Тесты

- Входные данные
./main
1
MGTUU F 2000 10 10
Выходные данные
12000101018675
- Входные данные
./main
2
MGTUU M 2009 10 10
MIEM f 1997 11 23
Выходные данные
22009101098079
419971123122329
- Входные данные
./main --fromfile r.txt //считалось с файла
3

MIEM m 2000 10 10
MGTUU f 2003 12 24
MIEM f 1999 5 6

Выходные данные
820001010198259
12003122473199
419990506853389

4. Входные данные

./main --tofile out.txt //записалось в файл с названием out.txt

1

MIEM F 1999 10 10

Выходные данные
419991010006159

5. Входные данные

./main --tofrom ppp.txt

Выходные данные
ERROR! Flags are not correct

6. Входные данные

./main

1

MGTUU F 1999 19 19

Выходные данные
Incorrect month entry. Who are you, stranger? Why do you have more than 12 month
or less than 0
You did a mistake. I can't print anything

7. Входные данные

./main --tofile 12.txt 67.txt -fromfile puf.txt

Выходные данные
ERROR! Too many flags

Тесты мейкфайла

1) Входные данные

make

Выходные данные

g++ -c func.cpp -o func.o

g++ -c c.cpp -o c.o

g++ func.o c.o -o main

2) Входные данные

make clean

Выходные данные

rm func.o c.o

3) Входные данные

```
make distclean
```

Выходные данные

```
rm func.o c.o
```

```
rm main *.txt Makefile
```

4) Входные данные сразу после make distclean

```
make
```

Выходные данные

```
make: *** No targets specified and no makefile found. Stop.
```