Отчёт. Лабораторная работа №3.

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. И `<cstring>` для того, чтобы сравнить введенный пользователем флаги с необходимыми.

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 полученные результат. Предусмотрен некорректный ввод флагов, а именно:

- чрезмерное количество флагов

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

- некорректное имя флага

```
else {
	std::cout << "ERROR! Flags are not correct" << std::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 введеным данным создадутся два студбилета (для МИЭМА и МГТУУ).

Также в c.cpp разработаны четыре класса, а именно:

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

```
class interface { //Виртуальный класс-интерфейс
public:
    void virtual generator(char sex, int year, int month, int day) = 0;
};
```

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

```
class template_pattern_generator : public interface { //наследуется от интерфейса
public:
    void generator(char sex,int year,int month,int day) override {
        gener_base(); //базовый конструктор
        generate(sex, year, month, day); //конструктор
```

```
        print(); //функция, которая будет выводить номер студбилета
    }
protected:
    void virtual gener_base() = 0;
    void virtual generate(char s, int y, int m, int d) = 0;
    void virtual print() { cout << "template_pattern_generator is working" << endl; }
};
```

- класс-генератор студбилета МИЭМА

```
class generator_MEM : public template_pattern_generator { //наследуется от генератора общего
                                                          // вида
                                                          // является также потомком интерфейса
    int sex;
    double result; // в этой переменной будет хранится номер студбилета
    int year, month, day, N, C;
protected:
    void gener_base() override { //переопределение функции (базовый конструктор)
        sex = 0;
        year = 0;
        month = 0;
        day = 0;
        N = -1;
        C = -1;
    }
    void print() override { //переопределение функции вывода данных
        if (flag_error == 1) //проверяется наличие ошибок
                             // если они есть, выводится ошибка
        {
            if (flag_write == 0) //тут проверка того, куда выводить полученное значение
            {
                cout.precision(0); //установка кол-ва знаков после запятой
                cout << std::fixed << result << endl; //вывод результата
            }
            else { fout.precision(0); fout << std::fixed << result << endl; }
        }
        else { cout << "ERROR!" << endl; }
    }
    void generate(char s, int y, int m, int d) override { //переопределение функции
        if (s == 'M') { sex = 8; } //Множество проверок на корректность введенных данных
        else if (s == 'W') { sex = 4; }
        else { flag_error = 0; }

        if (y > 0 && y < 2023) { year = y; }
        else { flag_error = 0; }

        if (m > 0 && m < 13) { month = m; }
        else { flag_error = 0; }

        if (m == 1 || m == 3 || m == 5 || m == 7 || m == 8 || m == 10 || m == 12) {
            if (d > 0 && d < 32) { day = d; }
            else { flag_error = 0; }
        }
        else if (m == 2)
        {
            if (d > 0 && d < 29) { day = d; }
            else { flag_error = 0; }
        }
        else if (m == 4 || m == 6 || m == 9 || m == 11) {
            if (d > 0 && d < 31) { day = d; }
            else { flag_error = 0; }
        }

        while (C == -1) //цикл выполняется до тех пор, пока не найдутся такие N и C, что
                        // сумма цифр, умноженных на их порядковый номер, будет кратна
                        // 11
        {
```

```
            result = 0;
            N = Random(0, 100000);
            if (N <= 9) { day *= 10000; result += day * 100; }
            else if (N >= 10 && N <= 99) { day *= 1000; result += day * 1000; }
            else if (N >= 100 && N <= 999) { day *= 100; result += day * 10000; }
            else if (N >= 1000 && N <= 9999) { day *= 10; result += day * 100000; }
            else { result += day * 1000000; }
            result += (double)sex * 100000000000000 + (double)year * 10000000000 + month *
100000000 + N * 10;
            C = ost(sex, year, month, day, N, 11); //данная функция будет описана позже
            if (C > -1)
            {
                result += C;
            }
        }
    }
};
```

- 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 = Random(0, 10000); //у N 4 знака, а не 5
```

Файл func.cpp

В этом файле дополнительно подключен файл <time.h >, который содержит функции для работы со временем

Функции, реализующие необходимые действия для решения задачи вынесены в отдельный файл.

int Random(int min, int max)

Как понятно из названия, данная функция выводит псевдослучайное число

```
int Random(int min, int max) {
      if (min > max) {
            int b = max;
            max = min;
            min = b;
      }
      srand(time(NULL));
      return min + rand() % (max - min);
     }
```

srand(time(NULL)) – функция, которая задает начальное значение для функции `rand()`

`rand()` генерирует псевдослучайные числа. А написанная формула позволяет получить псевдослучайное число на отрезке.

int ost(int s, int y, int m, int d, int 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 ∈ [0,10), что сумма цифр была кратна введенному num.

```
int ost(int s, int y, int m, int d, int n, int num) {
      int i;
      if (num == 11) { i = 14; }
      else if (num == 10) { i = 13; }
      int k = 0;
      while (n > 0) {
```

```
            int b = n % 10;
            k += b * (i--);
            n = (int)(n / 10);
        }
        while (d > 0) {
            int b = d % 10;
            k += b * (i--);
            d = (int)(d / 10);
        }
        while (m > 0) {
            int b = m % 10;
            k += b * (i--);
            m = (int)(m / 10);
        }
        while (y > 0) {
            int b = y % 10;
            k += b * (i--);
            y = (int)(y / 10);
        }
        k += s;
        int ans = -1;
        for (int C = 0; C < 10; C++) {
            if (num == 11) {
                if ((k + 15 * C) % 11 == 0) { ans = C; }
            }
            else if (num == 10) {
                if ((k + 14 * C) % 10 == 0) { ans = C; }
            }
        }
        return ans;
    }
```

Возвращает -1, если нужное значение не было найдено.

Файл lab67.h

```
#ifndef LAB67
#define LAB67
extern int flag_write;
extern int flag_read;
int Random(int min, int max);
int ost(int s, int y, int m, int d, int n, int num);
#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

1. Входные данные
   ./main
   W 2000 10 10
   Выходные данные
   12000101018675
   420001010618675

2. Входные данные
   ./main
   M 2009 10 10
   Выходные данные
   22009101091726
   820091100091725

3. Входные данные
   ./main --fromfile r.txt //считалось с файла
   M 2000 10 21
   Выходные данные
   12000101018675
   420001010618675

4. Входные данные
   ./main --tofile out.txt //записалось в файл с названием out.txt
   W 1999 10 10
   Выходные данные
   11999101049685
   419991010849680

5. Входные данные
   ./main --tofrom ppp.txt
   Выходные данные
   ERROR! Flags are not correct

6. Входные данные
   ./main
   W 2929 10 10
   Выходные данные
   ERROR!
   ERROR!

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.