

Отчёт к лабораторным работам 6–7

Лабораторная работа 6

Файл *main.cpp*

```
#include <iostream>
#include <fstream>
#include <cstdio>
#include <cstring>
#include <string>
#include <time.h>
#include "functions.h"
using std::cin;
using std::cout;
using std::endl;
using std::cerr;
using std::ifstream;
using std::ofstream;
using std::string;
using std::to_string;

const char* flag[2] = { "--tofile", "--fromfile" };

class stud_card {
public:
    virtual void gender(string sex1) = 0;
    virtual void c_in_numbers() = 0;
    virtual void date(int year1, int month1, int day1) = 0;
};

class MIEM : virtual public stud_card {
public:
    string sex, year, month, day, number, c, res = "";
    MIEM() {}; //default constructor
    void gender(string sex1) override {
        if (sex1 == "man") {
            res += "8";
        }
        else if (sex1 == "woman") {
            res += "4";
        }
    }
    void date(int year1, int month1, int day1) override {
        year = to_string(year1);
        res += year;
        if (month1 > 0 && month1 <= 12) {
            if (month1 >= 10) { month = to_string(month1); }
            else {
                month = "0" + to_string(month1);
            }
            res += month;
        }
        if (day1 >= 10 && day1 <= 31) { day = to_string(day1); }
        else if (day1 < 10 && day1 > 0) { day = "0" + to_string(day1); }
        res += day;
    }
    void c_in_numbers() override {
        int sum = 0;
        int position = 1;
        srand(time(0));
        string num = to_string(rand() % 90000 + 10000);
        res += num;
        for (int i = 0; i < res.size(); i++) {
```

```

        sum += ((res[i] - '0') * position);
        position++;
    }
    for (int cnumber = 0; cnumber <= 9; cnumber++) {
        if ((sum + cnumber * 15) % 11 == 0) {
            res += to_string(cnumber);
            break;
        }
    }
}
~MIEM() {}; //destructor
};

class MGTUUU : virtual public stud_card {
public:
    string sex, year, month, day, number, c, res = "";
    MGTUUU() {}; //default constructor
    void gender(string sex1) override {
        if (sex1 == "man") {
            res += "2";
        }
        else if (sex1 == "woman") {
            res += "1";
        }
    }
    void date(int year1, int month1, int day1) override {
        year = to_string(year1);
        res += year;
        if (month1 > 0 && month1 <= 12) {
            if (month1 >= 10) { month = to_string(month1); }
            else {
                month = "0" + to_string(month1);
            }
            res += month;
        }
        if (day1 >= 10 && day1 <= 31) { day = to_string(day1); }
        else if (day1 < 10 && day1 > 0) { day = "0" + to_string(day1); }
        res += day;
    }
    void c_in_numbers() override {
        int sum = 0;
        int position = 1;
        srand(time(0));
        string num = to_string(rand() % 9000 + 1000);
        res += num;
        for (int i = 0; i < res.size(); i++) {
            sum += ((res[i] - '0') * position);
            position++;
        }
        for (int cnumber = 0; cnumber <= 9; cnumber++) {
            if ((sum + cnumber * 14) % 10 == 0) {
                res += to_string(cnumber);
                break;
            }
        }
    }
}
~MGTUUU() {}; //destructor
};

int main(int argc, char** argv) {
    int yearstud, monthstud, daystud;
    string sexstud;
    if (argc == 1) {

```

```

    cerr << "Error: there are no flags" << endl;
}
else if (argc == 2) {
    cerr << "Error: not much segments" << endl;
}
else if (argc == 3) {
    char* name = argv[2];
    if (!(strcmp(argv[1], flag[0]))) {
        ofstream fout;
        fout.open(name);
        cin >> sexstud >> yearstud >> monthstud >> daystud;
        if (checkdata(yearstud, monthstud, daystud)) {
            MIEM stud1;
            stud1.gender(sexstud);
            stud1.date(yearstud, monthstud, daystud);
            stud1.c_in_numbers();
            MGTUU stud2;
            stud2.gender(sexstud);
            stud2.date(yearstud, monthstud, daystud);
            stud2.c_in_numbers();
            fout << stud1.res;
            fout << '\n';
            fout << stud2.res;
            fout << '\n';
            fout.close();
        }
        else { cerr << "Problems with date" << endl; }
    }
    else if (!(strcmp(argv[1], flag[1]))) {
        ifstream fin;
        fin.open(name);
        fin >> sexstud >> yearstud >> monthstud >> daystud;
        if (checkdata(yearstud, monthstud, daystud)) {
            MIEM stud1;
            stud1.gender(sexstud);
            stud1.date(yearstud, monthstud, daystud);
            stud1.c_in_numbers();
            MGTUU stud2;
            stud2.gender(sexstud);
            stud2.date(yearstud, monthstud, daystud);
            stud2.c_in_numbers();
            cout << stud1.res;
            cout << '\n';
            cout << stud2.res;
            cout << '\n';
        }
        else { cerr << "Problems with date" << endl; }
    }
}
else cerr << "Wrong flags" << endl;
}
else if (argc == 5) {
    if ((!(strcmp(argv[1], flag[1])))) {
        if ((!(strcmp(argv[3], flag[0])))) {
            char* name1 = argv[2];
            char* name2 = argv[4];
            ifstream fin;
            ofstream fout;
            fin.open(name1);
            fin >> sexstud >> yearstud >> monthstud >> daystud;
            if (checkdata(yearstud, monthstud, daystud)) {
                MIEM stud1;
                stud1.gender(sexstud);
                stud1.date(yearstud, monthstud, daystud);
                stud1.c_in_numbers();
            }
        }
    }
}

```

```

        MGTUU stud2;
        stud2.gender(sexstud);
        stud2.date(yearstud, monthstud, daystud);
        stud2.c_in_numbers();
        fout.open(name2);
        fout << stud1.res;
        fout << '\n';
        fout << stud2.res;
        fout << '\n';
        fin.close();
        fout.close();
    }
    else { cerr << "Problems with date" << endl; }
}
else {
    cerr << "Wrong flags" << endl;
}
}
else {
    cerr << "Wrong flags" << endl;
}
}
else {
    cerr << "Wrong flags" << endl;
}
}
}

```

В данном файле есть виртуальный класс-интерфейс `stud_card` с 3 чисто виртуальными функциями, реализованными в классах-наследниках `MIEM` и `MGTUU`, в которых реализованы три функции:

1. `void gender(string);`
2. `void date(int, int, int);`
3. `void c_in_numbers();`

Также в этом файле есть функция `int main(int argc, char ** argv)`, в которой учтены различные ситуации ввода флагов:

1. `--tofile <название файла>`
Запись результата (двух сформированных студенческих билетов) в заданный файл.
2. `--fromfile <название файла>`
Считывание данных студента из файла, заданного пользователем, для формирования двух студенческих билетов и вывода их на экран.
3. `--fromfile <название файла> --tofile <название файла>`
Считывание данных студента из файла, заданного пользователем, для формирования двух студенческих билетов и записи их в файл, заданный пользователем.

Файл `functions.cpp`

```

#include <iostream>
#include <string>
using std::to_string;

int checkdata(int yearr, int monthh, int dayy) {
    if (yearr <= 0 || (to_string(yearr).size()) < 4) { return 0; }
}

```

```

else {
    if (monthh < 0 || monthh > 12) { return 0; }
    else {
        if ((monthh == 1 || monthh == 3 || monthh == 5 || monthh == 7 ||
monthh == 8 || monthh == 10 || monthh == 12) && dayy > 0 && dayy <= 31) {
            return 1;
        }
        else if ((monthh == 4 || monthh == 6 || monthh == 9 || monthh ==
11) && dayy > 0 && dayy <= 30) {
            return 1;
        }
        else if (monthh == 2 && dayy > 0 && dayy <= 29) {
            int a = yearr % 10;
            int b = (yearr / 10) % 10;
            if ((a == b) && (a == 0) && (yearr % 400 == 0) && (dayy <=
29)) {
                return 1;
            }
            else if ((a != b) && (yearr % 4 == 0) && (dayy <= 29)) {
                return 1;
            }
            else if (((a != b) && (yearr % 4 != 0)) || ((a == b) &&
(a == 0) && (yearr % 400 != 0))) && (dayy <= 28)) {
                return 1;
            }
            else {
                return 0;
            }
        }
    }
}
return 0;
}

```

В данном файле реализована функция `int checkdata(int, int, int)`, которая осуществляет проверку даты на корректность (в том числе учтён вариант с високосным годом и количеством дней в феврале).

Файл functions.h

```

#ifndef FUNCTIONS
#define FUNCTIONS
int checkdata(int, int, int);
#endif

```

В данном файле присутствует объявление функции `int checkdata(int, int, int)`, которая реализована в файле `functions.h`

Файл Makefile

TR = prog

CC = g++

OB = functions.o main.o

\$(TR) : \$(OB)

\$(CC) \$(OB) -o \$(TR)

functions.o : functions.cpp

\$(CC) -c functions.cpp -o functions.o

main.o : main.cpp

\$(CC) -c main.cpp -o main.o

clean :

rm \$(OB)

distclean : clean

rm \$(TR) *.txt Makefile

В данном файле осуществлена сборка проекта и команды make, clean, distclean.

Также присутствуют файлы qwerty.txt и tr.txt, как один из вариантов теста программы (считывание из файла qwerty.txt данных студента) и запись их в tr.txt.

Например, содержимое qwerty.txt следующее:

man 1966 3 4

Тогда в качестве результата выполнения программы в файл tr.txt запишутся следующие два студенческих билета:

819660304815111

21966030495112

Лабораторная работа 7

Файл main.cpp

```
#include <iostream>
#include <fstream>
#include <cstdio>
#include <cstring>
#include <string>
#include <time.h>
#include "functions.h"
using std::cin;
using std::cout;
using std::endl;
using std::cerr;
using std::ifstream;
using std::ofstream;
```

```

using std::string;
using std::to_string;

const char* flag[2] = { "--tofile", "--fromfile" };

class template_pattern_student_number_generator {
public:
    string sex, year, month, day, number, c, res = "";
    virtual string generate(string sex1, int year1, int month1, int day1) final {
        gender(sex1);
        date(year1, month1, day1);
        c_in_numbers();
        en();
    }
    virtual void gender(string sex1) {};
    virtual void c_in_numbers() {};
    virtual void date(int year1, int month1, int day1) {
        year = to_string(year1);
        res += year;
        if (month1 > 0 && month1 <= 12) {
            if (month1 >= 10) { month = to_string(month1); }
            else {
                month = "0" + to_string(month1);
            }
            res += month;
        }
        if (day1 >= 10 && day1 <= 31) { day = to_string(day1); }
        else if (day1 < 10 && day1 > 0) { day = "0" + to_string(day1); }
        res += day;
    }
    virtual string en() final {
        return res;
    }
};

class MIEM : public template_pattern_student_number_generator {
public:
    MIEM() {}; //default constructor
    void gender(string sex1) override {
        if (sex1 == "man") {
            res += "8";
        }
        else if (sex1 == "woman") {
            res += "4";
        }
    }
    void c_in_numbers() override {
        int sum = 0;
        int position = 1;
        srand(time(0));
        string num = to_string(rand() % 90000 + 10000);
        res += num;
        for (int i = 0; i < res.size(); i++) {
            sum += ((res[i] - '0') * position);
            position++;
        }
        for (int cnumber = 0; cnumber <= 9; cnumber++) {
            if ((sum + cnumber * 15) % 11 == 0) {
                res += to_string(cnumber);
                break;
            }
        }
    }
    ~MIEM() {}; //destructor
};

```

```

class MGTUU : public template_pattern_student_number_generator {
public:
    MGTUU() {}; //default constructor
    void gender(string sex1) override {
        if (sex1 == "man") {
            res += "2";
        }
        else if (sex1 == "woman") {
            res += "1";
        }
    }
    void c_in_numbers() override {
        int sum = 0;
        int position = 1;
        srand(time(0));
        string num = to_string(rand() % 9000 + 1000);
        res += num;
        for (int i = 0; i < res.size(); i++) {
            sum += ((res[i] - '0') * position);
            position++;
        }
        for (int cnumber = 0; cnumber <= 9; cnumber++) {
            if ((sum + cnumber * 14) % 10 == 0) {
                res += to_string(cnumber);
                break;
            }
        }
    }
    ~MGTUU() {}; //destructor
};

class pointer {
public:
    template_pattern_student_number_generator* generator(string university1) {
        if (university1 == "MIEM") {
            MIEM* S1 = new MIEM;
            return S1;
        }
        else if (university1 == "MGTUU") {
            MGTUU* S2 = new MGTUU;
            return S2;
        }
        return 0;
    }
};

int main(int argc, char** argv) {
    int yearstud, monthstud, daystud;
    string universitystud, sexstud;
    if (argc == 1) {
        cerr << "Error: there are no flags" << endl;
    }
    else if (argc == 2) {
        cerr << "Error: not much segments" << endl;
    }
    else if (argc == 3) {
        char* name = argv[2];
        if (!(strcmp(argv[1], flag[0]))) {
            ofstream fout;
            fout.open(name);
            cin >> universitystud >> sexstud >> yearstud >> monthstud >> daystud;
            if (checkdata(yearstud, monthstud, daystud)) {

```



```

        pointer templateGenerator;
        string SB_1 = templateGenerator.generator(universitystud)-
>generate(sexstud, yearstud, monthstud, daystud);
        fout << universitystud;
        fout << ' ';
        fout << SB_1;
        fout.close();
    }
    else { cerr << "Problems with date" << endl; }
}
else if (!strcmp(argv[1], flag[1])) {
    ifstream fin;
    fin.open(name);
    fin >> universitystud >> sexstud >> yearstud >> monthstud >> daystud;
    if (checkdata(yearstud, monthstud, daystud)) {
        pointer templateGenerator;
        string SB_1 = templateGenerator.generator(universitystud)-
>generate(sexstud, yearstud, monthstud, daystud);
        cout << universitystud;
        cout << ' ';
        cout << SB_1;
        fin.close();
    }
    else { cerr << "Problems with date" << endl; }
}
else cerr << "Wrong flags" << endl;
}
else if (argc == 5) {
    if ((!strcmp(argv[1], flag[1]))) {
        if ((!strcmp(argv[3], flag[0]))) {
            char* name1 = argv[2];
            char* name2 = argv[4];
            ifstream fin;
            ofstream fout;
            fin.open(name1);
            fin >> universitystud >> sexstud >> yearstud >> monthstud >>
daystud;
            if (checkdata(yearstud, monthstud, daystud)) {
                pointer templateGenerator;
                string SB_1 = templateGenerator.generator(universitystud)-
>generate(sexstud, yearstud, monthstud, daystud);
                fout.open(name2);
                fout << universitystud;
                fout << ' ';
                fout << SB_1;
                fin.close();
                fout.close();
            }
            else { cerr << "Problems with date" << endl; }
        }
        else {
            cerr << "Wrong flags" << endl;
        }
    }
    else {
        cerr << "Wrong flags" << endl;
    }
}
else {
    cerr << "Wrong flags" << endl;
}
}
}

```

В данном файле есть класс-генератор общего вида с использованием паттерна Шаблонный Метод `template_pattern_student_number_generator` с 5 виртуальными функциями:

1. `void gender(string);`
2. `void date(int, int, int);`
3. `void c_in_numbers();`
4. `string en();`
5. `string generate(string, int, int, int);`

Также имеются два его класса-наследника: `MIEM` и `MGTUU`, в которых переопределены функции `void gender(string)` и `void c_in_numbers();`

Также в данном файле создан класс `pointer`, в котором при помощи указателя на объект типа `template_pattern_student_number_generator` определяется, указатель на объект какого класса `MIEM` или `MGTUU` возвращается в зависимости от введенного университета.

В этом файле есть функция `int main(int argc, char ** argv)`, в которой учтены различные ситуации ввода флагов:

4. `--tofile <название файла>`
Запись результата (соответствующего сформированного студенческого билета) в заданный файл.
5. `--fromfile <название файла>`
Считывание данных студента из файла, заданного пользователем, для формирования соответствующего студенческого билета и вывода его на экран.
6. `--fromfile <название файла> --tofile <название файла>`
Считывание данных студента из файла, заданного пользователем, для формирования соответствующего студенческого билета и записи их в файл, заданный пользователем.

Файл `functions.cpp`

```
#include <iostream>
#include <string>
using std::to_string;

int checkdata(int yearr, int monthh, int day) {
    if (yearr <= 0 || (to_string(yearr).size()) < 4) { return 0; }
    else {
        if (monthh < 0 || monthh > 12) { return 0; }
        else {
            if ((monthh == 1 || monthh == 3 || monthh == 5 || monthh == 7 ||
                monthh == 8 || monthh == 10 || monthh == 12) && day > 0 && day <= 31) {
                return 1;
            }
        }
    }
}
```

```

else if ((monthh == 4 || monthh == 6 || monthh == 9 || monthh ==
11) && dayy > 0 && dayy <= 30) {
    return 1;
}
else if (monthh == 2 && dayy > 0 && dayy <= 29) {
    int a = yearr % 10;
    int b = (yearr / 10) % 10;
    if ((a == b) && (a == 0) && (yearr % 400 == 0) && (dayy <=
29)) {
        return 1;
    }
    else if ((a != b) && (yearr % 4 == 0) && (dayy <= 29)) {
        return 1;
    }
    else if (((a != b) && (yearr % 4 != 0)) || ((a == b) &&
(a == 0) && (yearr % 400 != 0))) && (dayy <= 28)) {
        return 1;
    }
    else {
        return 0;
    }
}
}
}
return 0;
}

```

В данном файле реализована функция `int checkdata(int, int, int)`, которая осуществляет проверку даты на корректность (в том числе учтён вариант с високосным годом и количеством дней в феврале).

Файл functions.h

```

#ifndef FUNCTIONS
#define FUNCTIONS
int checkdata(int, int, int);
#endif

```

В данном файле присутствует объявление функции `int checkdata(int, int, int)`, которая реализована в файле `functions.h`

Файл Makefile

TR = prog

CC = g++

OB = functions.o main.o

\$(TR) : \$(OB)

\$(CC) \$(OB) -o \$(TR)

functions.o : functions.cpp

\$(CC) -c functions.cpp -o functions.o

```
main.o : main.cpp
```

```
$(CC) -c main.cpp -o main.o
```

```
clean :
```

```
rm $(OB)
```

```
distclean : clean
```

```
rm $(TR) *.txt Makefile
```

В данном файле осуществлена сборка проекта и команды make, clean, distclean.

Также присутствуют файлы qwerty.txt и tr.txt, как один из вариантов теста программы (считывание из файла qwerty.txt данных студента) и запись их в tr.txt.

Например, содержимое qwerty.txt следующее:

МІЕМ man 1966 3 4

Тогда в качестве результата выполнения программы в файл tr.txt запишется соответствующий студенческий билет:

МІЕМ 819660304317112