

## Отчет по лабораторной работ 6–7.

Программа состоит из 4 файлов:

- lab\_6\_7.cpp;
- Source.cpp;
- Header.h;
- Makefile.

### Файл lab\_6\_7. cpp

В файле подключены заголовочные файлы <iostream>, <string>, <cstring>, <fstream>, а также “Header.h”.

В файле находится функция main() с аргументами int argc, char\*\* argv. В зависимости от введенных флагов функция вызывает функции.

1. Считывание из консоли и вывод в консоль. Вызов был только функции.

```
if (argc == 1) Consol();
```

2. Введены флаги “--fromfile” и “--tofile”. Функция проверяет достаточно ли введено значений (файл, из которого будет запись, и файл, в которой будет запись (не обязательно)).

```
else if (argc >= 4 && argc <= 5 && strcmp(argv[1], "--fromfile") == 0 && strcmp(argv[2], "--tofile") == 0) Fromtofile();
```

3. Введен только флаг “--fromfile”. Проверка на кол-во значений, есть ли файл для считывания.

```
else if (argc == 3 && strcmp(argv[1], "--fromfile") == 0) Fromfile();
```

4. Введен флаг “--tofile”. Проверка на кол-во введенных значений.

```
else if (argc >= 2 && argc <= 3 && strcmp(argv[1], "--tofile") == 0) Tofile();
```

5. В любом другом случае программа выдаст ошибку.

```
else std::cout << "Error! You entered incorrectly" << std::endl;
```

В файле присутствует виртуальный класс интерфейс Styd.

```
class Styd {
```

```
public:
```

```
virtual std::string generator(char gender, int year, int month, int day) = 0;
```

```
};
```

Также шаблонный класс Generator.

```
class Generator : public Std{
public:
    bool error;
    int year, day, month, gender;
protected:
    virtual void gen() = 0;
    virtual std::string gener_styd(char gend, int y, int mon, int d) = 0;
    std::string generator(char gender, int year, int month, int day) final {
        gen();
        gener_styd(gender, year, month, day);
        return gener_styd(gender, year, month, day);
    }
    virtual void gener(char gend, int y, int mon, int d) final {}
};
```

От класса Generator наследуется класс Std\_MGTUU, который генерирует студенческие билеты вуза МГТУ. Дополнительно используется функция numT.

```
class Std_MGTUU : public Generator {
    int gender;
    long long styd;
protected:
    std::string Num;
    bool error = true;
    void gen() override{
        gender = 0;
        year = 0;
        month = 0;
        day = 0;
    }
    std::string gener_styd(char gend, int y, int mon, int d) override {
        if (gend == 'm') gender = 2;
        else if (gend == 'w') gender = 1;
        year = y;
        month = mon;
        day = d;
        Num = numT(gender, year, month, day);
        return Num;
    }
};
```

```
};
```

Также от класса Generator наследуется класс Styd\_MIEM для генерации номеров студенческий билетов МИЭМ. Дополнительно используется функция numM.

```
class Styd_MIEM : public Generator {
    int gender, year, month, day, Num;
    long long styd;
protected:
    bool error = true;
    void gen() override {
        gender = 0;
        year = 0;
        month = 0;
        day = 0;
    }
    std::string gener_styd(char gend, int y, int mon, int d) override {
        if (gend == 'm') gender = 8;
        else if (gend == 'w') gender = 4;
        year = y;
        month = mon;
        day = d;
        std::string Num = numM(gender, year, month, day);
        return Num;
    }
};
```

Для генерации билета определенного института введен класс vyz, в котором в зависимости от аргумента выполняется один из генераторов.

```
class vyz {
public:
    Styd* gen(std::string str) {
        if (str == "MIEM") {
            Styd* s1 = new Styd_MIEM;
            return s1;
        }
        else{
            Styd* s2 = new Styd_MGTUU;
            return s2;
        }
    }
};
```

Далее введены функции, которые вызываются при работе main().

1. Consol() предназначена запрашивает нужное кол-во билетов и, считывая с консоли данные, выводит билеты.

```
void Consol() {
    std::cout << "Enter the number of students." << std::endl;
    int n,i=0;
    std::cin >> n;
    while (i < n) {
        std::cout << "Enter the University (MIAMI or MGTU),gender (m,w), your
year of birth, month and day." << std::endl;
        std::string v;
        int year, month, day;
        char gender;
        std::cin >> v >> gender >> year >> month >> day;
        if (proverka(v, gender, year, month, day) == false) {
            std::cout << "Data entered incorrectly." << std::endl; return;
        }
        vyz s;
        std::string stydak = s.gen(v)->generator(gender, year, month, day);
        std::cout << stydak << std::endl;
        i += 1;
    }
    return;
}
```

2. Fromtofile() считывает данные из файла и выводит в файл в зависимости задан он или нет. Если файл не задан, то программа выведет в файл Tofile.txt.

```
void Fromtofile() {
    std::ifstream file;
    file.open(arg[3]);
    if (!file) {
        std::cout << "Error! The file did not open." << std::endl; return;
    }
    std::ofstream file2;
    if (arc == 5) file2.open(arg[4]);
    else file2.open("Tofile.txt");
    if (!file2) {
        std::cout << "Error! The file did not open." << std::endl; return;
    }
    std::string v;
    int year, month, day;
```

```

char gender;
while (!file.eof()) {
    file >> v >> gender >> year >> month >> day;
    if (proverka(v, gender, year, month, day) == false) {
        std::cout << "Data entered incorrectly." << std::endl; return;
    }
    vyz s;
    std::string stydak = s.gen(v)->generator(gender, year, month, day);
    file2 << stydak << std::endl;
}
std::cout << "Check your file or Tofile.txt" << std::endl;
file.close();
file2.close();
return;
}

```

3. Fromfile() считывает данные из файла и выводит номера билетов в КОНСОЛЬ.

```

void Fromfile() {
    std::ifstream file;
    file.open(arg[2]);
    if (!file) {
        std::cout << "Error! The file did not open." << std::endl; return;
    }
    std::string v;
    int year, month, day;
    char gender;
    while (!file.eof()) {
        file >> v >> gender >> year >> month >> day;
        if (proverka(v, gender, year, month, day) == false) {
            std::cout << "Data entered incorrectly." << std::endl; return;
        }
        vyz s;
        std::string stydak = s.gen(v)->generator(gender, year, month, day);
        std::cout << stydak << std::endl;
    }
    file.close();
    return;
}

```

4. Tofile() считывает данные из консоли и выводит их в заданный файл или в Tofile.txt.

```

void Tofile() {
    std::ofstream file2;

```

```

if (arc == 5) file2.open(arg[4]);
else file2.open("Tofile.txt");
if (!file2) {
    std::cout << "Error! The file did not open." << std::endl; return;
}
std::cout << "Enter the number of students." << std::endl;
int n, i = 0;
std::cin >> n;
while (i < n) {
    std::cout << "Enter the University (MIAMI or MGTU),gender (m,w), your
year of birth, month and day." << std::endl;
    std::string v;
    int year, month, day;
    char gender;
    std::cin >> v >> gender >> year >> month >> day;
    if (proverka(v, gender, year, month, day) == false) {
        std::cout << "Data entered incorrectly." << std::endl; return;
    }
    vyz s;
    std::string stydak = s.gen(v)->generator(gender, year, month, day);
    file2 << stydak << std::endl;
    i += 1;
}
file2.close();
std::cout << "Check your file or Tofile.txt" << std::endl;
return;
}

```

## Файл Source.cpp

В файле введены заголовочные файлы <iostream>, <string>, <ctime> и "Header.h".

Функция numT() введена для помощи генерации номер билетов в МГТУ.

```

std::string numT(double g, double y, double m, double d) {
    long long Num;
    int N4;
    bool p = false;
    Num = g * 100000000000000 + y * 10000000000 + m * 100000000 + d * 100000;
    srand(unsigned(time(0)));
    N4 = rand() % 1000;
    while (p != true) {
        srand(unsigned(time(0)));
        int N5 = rand() % 1000;
    }
}

```

```

        Num += (N4 * 10);
        if (Num % 10 == 0) p = true;
        else if (Num % 10 < 10) {
            p = true;
            Num += (Num % 10);
        }
    }
    std::string n = std::to_string(Num);
    return n;
}

```

Функция numM() введена для помощи генерации номера билетов в МИЭМ.

```

std::string numM(double g, double y, double m, double d) {
    long long int Num;
    bool p = false;
    Num = g * 1000000000000000 + y * 100000000000 + m * 100000000 + d * 1000000;
    srand(unsigned(time(0)));
    int N5 = rand() % 10000;
    while (p != true) {
        srand(unsigned(time(0)));
        int N5 = rand() % 10000;
        Num += (N5 * 10);
        if (Num % 11 == 0) p = true;
        else if (Num % 11 < 10) {
            p = true;
            Num += (Num % 11);
        }
    }
    std::string n = std::to_string(Num);
    return n;
}

```

Функция proverka() введена для проверки входных данных.

```

bool proverka(std::string v, char gender, int year, int month, int day) {
    if (v != "MIEM" && v != "MGTUU") return false;
    if (gender != 'm' && gender != 'w') return false;
    if (year <= 1900 || year > 2022) return false;
    if (month <= 0 || month > 12) return false;
    if (day <= 0 || day > 31) return false;
    if (month == 2 && ((year % 4 == 0 && year % 100 != 0) || year % 400 == 0) &&
day > 29) return false;
    if (month == 2 && (year % 400 != 0 || year % 4 != 0 || (year % 100 == 0 &&
year%400!=0)) && day > 28) return false;
}

```

```

        if ((month == 4 || month == 6 || month == 9 || month == 11) && day > 30)
return false;
        return true;

}

```

## Файл Header.h

В заголовочном файле собраны все глобальные переменные и функции.

```

extern int arc;
extern char** arg;
void Fromtofile();
void Fromfile();
void Tofile();
void Consol();
std::string numM(double gender, double year, double month, double day);
std::string numT(double gender, double year, double month, double day);
bool proverka(std::string v, char gender, int year, int month, int day);

```

## Файл Makefile

Файл создан для компиляции программы. Реализованы методы clean, который удаляет файлы с расширением .o, и distclean, который удаляет все файлы, в том числе сам Makefile.

```

main: Source.o lab_6_7.o
    g++ Source.o lab_6_7.o -o main

Source.o: Source.cpp
    g++ -c Source.cpp -o Source.o

lab_6_7.o: lab_6_7.cpp
    g++ -c lab_6_7.cpp -o lab_6_7.o

clean:
    rm Source.o lab_6_7.o

distclean : clean
    rm main *.txt Makefile

```