Лабораторная работа №6

Файл flags.cpp

В данном файле используются заголовочные файлы `<cstring>` для сравнения двух строк, `<fstream>` для чтения и записи данных из/в файл, `"Header.h"`, в котором содержатся классы, и `using` – объявления, которые сообщают компилятору, что будет использованы объекты из пространства имён `std`.

Используются аргументы командной строки argc и argv для ввода и считывания флагов, которые указывают вывод результата в файл и ввод из файла, --tofile и --fromfile соответственно. Задаём две переменные логического типа данных. Затем с помощью операторов if-else и else if задаём условие, которое проверяет количество аргументов в командной строке и осуществляет лексикографическую проверку строк. Если условие выполняется, то флаги приравниваются к true и выполняется вывод номеров студенческих билетов МИЭМа и МГТУУ. Пример для флага “—tofile”:

```
if (argc==3 && !strcmp(argv[1],"--tofile"))
{
    tofile=true;
    ofstream fout;
    fout.open(argv[2]);
        cout<<"Write your gender, year, month and day of
birth."<<endl<<"Example: female 2002 10 29"<<endl;
    cin>> sex >> year >> month >> day;
if (data_verification(sex,year, month,day)==true)
    {
        MIEM st1;
        st1.gender(sex);
        st1.date(year,month,day);
        st1.complicated_numbers();
        MGTUU st2;
        st2.gender(sex);
        st2.date(year,month,day);
        st2.complicated_numbers();
        st1.Print(fout);
        st2.Print(fout);
        fout.close();
    }
}
```

Файл func.cpp

В данном файле представлена функция, переводящая переменную с типом данных `char` в переменную целочисленного типа. Она используется в заголовочной файле `"Header.h"`. Для функции применена конструкция switch case:

```
int to_integer(char a) {
    switch (a) {
    case '0':
        return 0;
```

```
    case '1':
        return 1;
    case '2':
        return 2;
    case '3':
        return 3;
    case '4':
        return 4;
    case '5':
        return 5;
    case '6':
        return 6;
    case '7':
        return 7;
    case '8':
        return 8;
    case '9':
        return 9;
    default:
        return -1;
    }
}
```

Также представлена функция, которая проверяет корректность входных данных, она используется в файле «flags.cpp»:

```
bool data_verification(string s,int y,int m,int d)
{
    if ((s!="Female" && s!="female") && (s!="Male" &&
s!="male"))
    {
        cerr<<"Incorrect gender: write female/male;
Female/Male"<<endl;
        return false;
    }
    if (y<1000 || y>2023)
    {
        cerr<<"Error of year";
        return false;
    }
    if (m<1 || m>12)
    {
        cerr<<"Error of month";
        return false;
    }
    if (d<1 || d>31)
    {
        cerr<<"Error of day";
        return false;
    }
    return true;
}
```

Файл Header.h

В файле используются заголовочные файлы <fstream>, <time.h> (для использования функции time() в генерировании случайных чисел), <string> (для использования строк).

Class stud_tick – виртуальный класс-интерфейс, который содержит чисто-виртуальные методы и один обычный метод, который одинаков для классов-наследников:

```
class stud_tick
{
protected:
    string printer;
    string str;
    string sex, y, m, d,n,c;
public:
    virtual void gender(string sex) = 0;
    virtual void complicated_numbers() = 0;
    virtual void Print(ostream&) = 0;
    virtual void date (int year,int month,int day)
    {
        if (year>1950 && year<2023)
        {
            y=to_string(year);
            printer+=y;
        }
        if (month>0 && month<13)
        {
            if (month>9 && month<13)
                m=to_string(month);
            else if (month>0 && month<9)
                m="0" + to_string(month);
            printer+=m;
        }
        if (day>0 && day<32)
        {
            if (day>10 && day<32)
                d=to_string(day);
            else if (day<10 && day>0)
                d="0"+to_string(day);
            printer+=d;
        }
    }
};
```

Class MIEM – наследник класса stud_tick. Он переопределяет методы базового класса, таким образом реализовывается полиморфизм в классе. После каждого имени метода пишется модификатор **override** , который позволяет компилятору следить за тем, чтобы метод, помеченный этим модификатором действительно переопределял метод базового класса.
В методе gender(string sex) каждому полу задаётся определённая цифра:

```
void gender(string sex) override
    {
        if (sex == "Female" || sex == "female" )
        {
            string sex= "4";
            printer+=sex;
        }
        else if (sex == "Male" || sex == "male")
        {
            string sex = "8";
            printer+=sex;
        }
    }
```

В методе complicated_numbers() задаётся псевдослучайное число и ищется последняя цифра номера студенческого билета:

```
void complicated_numbers() override
    {
        int tempor = 0;
        int cn=0;
        int poz=0;
        string num;
        bool flag=false;
        while (flag!=true)
        {
            srand(time(0));
            num=to_string(rand()%89999+10000);
            for(int i=0;i<printer.size();i++)
            {
                tempor+=(to_integer(printer[i]))*(i+1);
                poz+=1;
            }
            poz+=1;
            for(int i=0;i<num.size();i++)
            {
                tempor+=to_integer(num[i])*poz;
                poz+=1;
            }
            for (cn=0;cn<=9;cn++)
            {
                if ((tempor+cn*15)%11==0)
                {
                    flag=true;
                    break;
                }
            }
        }
        if (flag==true)
        {
            printer+=num;
            c=to_string(cn);
            printer+=c;
```

```
        }
    }
```

В методе Print(osteram& out) выводится номер студенческого билета:

```
void Print(ostream& out) override
    {
        out<<"MIEM: "<<printer<<endl;
    }
```

Class MGTUU – наследник класса stud_tick. Он так же, как и Class MIEM, переопределяет методы базового класса.

Makefile

Makefile собирает программный проект и очищает от временных файлов. Distclean - удалит не только файлы с расширением .o , которые удаляются при исполнении цели 'clean', но также exe.

```
make : flags.o func.o
    g++ -o make flags.o func.o
clean :
     rm *.o
distclean : clean
          rm -f make
```

Лабораторная работа №7

Изменения в файле «Header.h»

В classe stud_tick только одна чисто-виртуальная функция:

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

};
```

Создан класс-генератор общего вида pattern_generator, который наследуется от класса stud_tick. Данный класс определяет скелет алгоритма. Классы MIEM и MGTUU наследуются от класса-генератора и реализовывают шаги алгоритма шаблонного метода:

```
class pattern_generator : public stud_tick
{
public:
```

```
    string printer;
    string sex, y, m, d, n, c;
    string generate(string sex,int year, int month,int day)
final //final используется для того, чтобы метод не
переопределялся
    {
        gender(sex);
        date(year,month,day);
        complicated_numbers();
        return otv();
    }
    virtual void gender(string sex){};
    virtual void complicated_numbers(){};
    virtual void date (int year,int month,int day)
    {
        if (year>1950 && year<2023)
        {
            y=to_string(year);
            printer+=y;
        }
        if (month>0 && month<13)
        {
            if (month>9 && month<13)
                m=to_string(month);
            else if (month>0 && month<9)
                m="0" + to_string(month);
            printer+=m;
        }
        if (day>0 && day<32)
        {
            if (day>10 && day<32)
                d=to_string(day);
            else if (day<10 && day>0)
                d="0"+to_string(day);
            printer+=d;
        }
    }
    virtual string otv() final
    {
        return printer;
    }
};
```

Создан класс, в котором с помощью указателя на объект типа stud_tick определяется, что если введён университет «MIEM»/ «MGTUU» , то возвращается указатель на объект класса «MIEM»/ «MGTUU»:

```
class temp
{
public:
    stud_tick* generator(string university)
    {
        if (university=="MIEM")
```

```
        {
            MIEM* S_B1 = new MIEM;
            return S_B1;
        }
        else if (university=="MGTUU")
        {
            MGTUU *S_B2 = new MGTUU;
            return S_B2;
        }
        return 0;
    }
};
```

Изменение в файле «flags.cpp»
Теперь выводится студенческий билет только одного из университета. Пользователь должен указать университет и свои данные. Пример для флага “—tofile”:

```
if (argc==3 && !strcmp(argv[1],"--tofile"))
    {
        tofile=true;
        ofstream fout;
        fout.open(argv[2]);
        cout<<"Write university, your gender, year, month and
day of birth."<<endl<<"Example: MIEM female 2002 10 29"<<endl;
        cin>> university>> sex >> year >> month >> day;
        temp gen;
        string studb = gen.generator(university)-
>generate(sex,year,month,day);
        fout<<university<<"\n"<<studb<<endl;
        fout.close();
    }
```