Лабораторная работа по языкам программирования №6-7

Задача: Виртуальный класс-интерфейс для генераторов номеров студбилетов, от которого необходимо отнаследоваться два раза: МИЭМовским и ЭмГэТУУшным классом-генератором (рекомендуется изучить ключевые слова override и final). Протестировать на примерах.

Решение: Разработка виртуального класса с чисто виртуальной функцией, разработка двух классов наследников в которых будет определена функция. Разработка класса для генерации экземпляров классов наследников

Разработка виртуального базового класса Uni

```
class
uni
{
        public:
           int sex;
           std::string year;
           std::string month;
           std::string day;

           virtual std::string propusk(int, std::string, std::string, std::string) = 0;
        };
```

Разработка класса наследника Miem

```
class Miem :public uni {
   std::string propusk(int sex, std::string year, std::string month, std::string day) override{
        int sm = 0; // переменная для подсчета итоговой суммы
        std::string s = ""; //строчка, в которой будет находиться сгенерированный пропуск
        if (sex == 0) {
            s += std::to_string(8);//исходя из условия 8 – код для мужчины


        }
        else if(sex == 1) {
            s += std::to_string(4); // код для женщины, оба кода добавляем в строчку с помощью to_string()
        }
        else{
            std::cout<<"wrong args";// обработка ошибки
            return 0;
        }
        if (stoi(year) >= 1955 && stoi(year) <= 2022 && stoi(month) >= 1 && stoi(month) <= 12 && stoi(day) >= 1 && stoi(day) <= 30) {
            s = s + (year)+(month)+(day); // проверка даты на существование
        }
        else {
            std::cout<<"wrong args";//обработка ошибок
            return 0;
        }
        int id = rand() % (99999 - 10000) + 10000; // генерация псевдослучайного числа в диапазоне (10000,99999)
```

```
        std::string s1 = s+std::to_string(id);
        int sm1 = 0;
        for (int i = 0; i < s1.size(); ++i)
            sm1 += (s1[i] - '0') * (i + 1);
        if (sm1 % 11 == 4){
            id +=1;
        // так как для найденной суммы не всегда можно найти число, чтобы итоговая сумма делилась на 11, делаем проверку
        }
        s = s + std::to_string(id);
        for (int i = 0; i < s.size(); ++i)
            sm += (s[i] - '0') * (i + 1);
        for (int i = 0; i < 10; ++i) {
            if ((sm  % 11 + (i*15)%11)%11 == 0) {
                s += std::to_string(i); // находим нужную цифру и возвращаем строчку с генерированным пропуском
                return s;
                break;
            }
        }
        return 0;


    }
};
```

Разработка класса наследника MGTUU

```
class MGTUU : public uni {
    std::string propusk(int sex, std::string year, std::string month, std::string day) override {
        int sm = 0;
        std::string s = "";
        if (sex == 0) {
            s += std::to_string(2);


        }
        else if(sex == 1) {
            s += std::to_string(1);
        }
        else{
            std::cout<<"wrong args";
            return 0;
        }
        if (stoi(year) >= 1955 && stoi(year) <= 2022 && stoi(month) >= 1 && stoi(month) <= 12 && stoi(day) >= 1 && stoi(day) <= 30) {
            s = s + (year)+(month)+(day);
        }
        else {
            return 0;
        }
        int id = rand() % (9999 - 1000) + 1000;
        std::string s1 = s+std::to_string(id);
        int sm1 = 0;
        for (int i = 0; i < s1.size(); ++i)
            sm1 += (s1[i] - '0') * (i + 1);
```

```
        if (((sm1 % 10)%2)!=0){
            id +=1;


        }
        s = s + std::to_string(id);
        for (int i = 0; i < s.size(); ++i)
            sm += (s[i] - '0') * (i + 1);
        for (int i = 0; i < 10; ++i) {
            if ((sm  % 10 + (i*14)%10)%10 == 0) {
                s += std::to_string(i);
                return s;
                break;
            }
        }
        return 0;


    }
};
```

Классы разработаны полностью аналогично, за исключением проверки кратности суммы, которая выполнена аналогично, но для 10

Разработка класса генератора классов

```
class aboba{
public:
    uni* method(std::string name){
        if (name == "MIEM"){
            Miem *a = new Miem;
            return a;
        }
        else if( name == "MGTUU"){
            MGTUU *a = new MGTUU;
            return a;
        }
        exit(1);

    }

};
```

В зависимости от имени генератор возвращает экземпляр класса наследника

Функции для чтения флагов

```
int read_cin( int flag) { // чтение с консоли
    std::cout << "ENTER UNIVETRISTY: ";
    std::string name;

    std::cin >> name;
    std::cout << "ENTER SEX 0 FOR MEN AND 1 FOR WOMEN: ";
    int sex;
    std::cin >> sex;
    std::cout << "ENTER YEAR: ";
    std::string year;
    std::cin >> year;
    std::cout << "ENTER MONTH: ";
    std::string month;
    std::cin >> month;
    std::cout << "ENTER DAY: ";
    std::string day;
    std::cin >> day;
    if (flag == 0){ // флаг для вывода: 0 – в консоль, 1 – в файл
        if (name == "MIEM") {
            aboba uri;
            std::cout << uri.method("MIEM")->propusk(sex, year, month, day);
        }
        else if (name == "MGTUU") {
            aboba uri;
            std::cout << uri.method("MGTUU")->propusk(sex, year, month, day);
        }
```

```
        else{
            std::cout<<"wrong args";
        }

    }
    else if (flag == 1) {
        std::ofstream out;
        out.open("tofile.txt");
        if (name == "MIEM") {
            aboba uri;
            out << uri.method("MIEM")->propusk(sex, year, month, day);
            out.close();
        }
        else if(name == "MGTUU") {
            aboba uri;
            std::cout << uri.method("MGTUU")->propusk(sex, year, month, day);
            out.close();
        }
        else{
            std::cout<<"wrong args";
        }
    }
    return 0;
}
```

\

```
int read_file(int flag) { // функция для чтения из файла
    std::ifstream file("fromfile.txt");
    if (!file.is_open()) {
        std::cout << "MISTAKE";
        return 0;
    }
    std::string name;
    file >> name;
    int sex;
    file >> sex;
    std::string year;
    file >> year;
    std::string month;
    file >> month;
    std::string day;
    file >> day;
    if (flag == 0) { // флаг для записи ответа: 0 – в консоль, 1 – в файл
        if (name == "MIEM") {
            aboba uri;
            std::cout << uri.method("MIEM")->propusk(sex, year, month, day);



        }
        else if (name == "MGTUU") {
            aboba uri;
            std::cout << uri.method("MGTUU")->propusk(sex, year, month, day);
        }
```

```
        else{
            std::cout<<"wrong args";
        }
    }
    else if(flag == 1){
        std::ofstream out;
        out.open("tofile.txt");
        if (name == "MIEM") {
            aboba uri;
            out << uri.method("MIEM")->propusk(sex, year, month, day);
            out.close();
        }
        else if(name == "MGTUU") {
            aboba uri;
            out << uri.method("MGTUU")->propusk(sex, year, month, day);
            out.close();
        }
        else{
            std::cout<<"wrong args";
        }

    }
    return 0;


}
```

Main

```
int main(int argc, char** argv)
{
   if (argc == 1){
      read_cin(0);// флага нет, ввод и вывод в консоли
      exit(0);
   }
   else  if (argc == 2){
      if (strcmp(argv[1], "--tofile") == 0) {
         read_cin(1);// читает в консоли ответ записывает в файл
         exit(0);
      }
      else if (strcmp(argv[1], "--fromfile") == 0) {
         read_file(0); // читает из файла-ответ в консоль
         exit(0);


      }









   }
   else if (argc == 3){
      if (strcmp(argv[1], "--tofile") == 0 && strcmp(argv[2], "--fromfile") == 0 ||
         strcmp(argv[2], "--tofile") == 0 && strcmp(argv[1], "--fromfile") == 0) {
            read_file(1);// читает из файла – ответ в файл
            exit(0);
      }
```

```
    }



}
```

Makefile

```
out : main.o func.o
	g++ main.o func.o -o out

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

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

clean :
	rm *.o
distclean:
	rm out *.o
```