

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ «НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

Кафедра «Компьютерная безопасность»

ОТЧЕТ
К ЛАБОРАТОРНОЙ РАБОТЕ №5

по дисциплине

«Языки программирования»

Работу выполнила
студентка группы
СКБ-232

подпись, дата

Д.В. Иванова

Работу проверил

подпись, дата

С.А. Булгаков

Содержание

Постановка задачи	3
1 Описание функции main	4
2 Тестирование	4
1 Проверка вывода в XML формате	4
2 Проверка вывода информации о городе	4
3 Смена мэра	4
4 Изменение числа жителей	5
5 Конструктор с 3 параметрами	5
6 Конструктор копирования	5
Приложение А	6
Приложение Б	7

Постановка задачи

Нужно было разработать класс ‘Object’ описывающий какой-либо объект окружающего мира. Предусмотреть в классе наличие характеристик либо состояния которые бы хранились в его полях с учетом принципа инкапсуляции. Для указанного класса реализовать конструкторы умолчания и копирования, а так же деструктор. А также реализовать в классе ‘Object’ метод ‘WriteXML(std::ostream)’ записывающий представление объекта класса на языке XML в поток вывода языка Си++.

1 Описание функции main

Описание класса

Класс Town содержит 4 приватных поля: количество жителей, имя мэра, название города и название страны. У класса три конструктора: без параметров (который по умолчанию присваивает страну Russia, город Moscow, мэра Sobyenin и количество жителей 1000000), с 3 параметрами: название города, имя мэра и количество жителей и конструктор копирования. А также деструктор. Кроме того, класс содержит методы: изменение майора, изменение количества жителей, возвращение количества жителей, имя мэра, название города, название страны, вывод информации о городе в формате текста и в формате XML.

2 Тестирование

1 Проверка вывода в XML формате

```
Town town(5000, "kitten", "Norway", "Prehewill");
town.WriteXML(std::cout);
output:
<?xml version="1.0" encoding="UTF-8" ?>
  <NumberOfCitizens =" 5000" />
  <MayorName =" kitten " />
  <CountryName =" Norway " />
  <TownName =" Prehewill " />
```

2 Проверка вывода информации о городе

```
Town Moscow;
Moscow.Town_info();
output:
-----
Country: Russia
Town name: Moscow
Mayor: Sobyenin
Number of citizens: 1000000
-----
```

Город Moscow создан конструктором без параметров, поэтому подобный вывод верен.

3 Смена мэра

```
Town Moscow;
Moscow.change_mayor("C0olH4ck3r");
Moscow.Town_info();
output:
```

```
-----  
Country: Russia  
Town name: Moscow  
Mayor: C0olH4ck3r  
Number of citizens: 1000000  
-----
```

4 Изменение числа жителей

```
Moscow.change_num_of_cit(2000000);  
Moscow.Town_info();  
output:
```

```
-----  
Country: Russia  
Town name: Moscow  
Mayor: C0olH4ck3r  
Number of citizens: 2000000  
-----
```

5 Конструктор с 3 параметрами

```
Town town(5000, "kitten", "Norway", "Prehewill");  
output:
```

```
-----  
Country: Norway  
Town name: Prehewill  
Mayor: kitten  
Number of citizens: 5000  
-----
```

6 Конструктор копирования

```
Town Moscow2(Moscow);  
Moscow2.Town_info();  
output:
```

```
-----  
Country: Russia  
Town name: Moscow  
Mayor: Sobyenin  
Number of citizens: 1000000  
-----
```

Приложение А

UML 2.0-схема класса *Town*

Town
<ul style="list-style-type: none">- number_of_citizens: int- mayor_name: std::string- country_name: std::string- town_name: std::string
<ul style="list-style-type: none">+ Town() <<constructor>>+ Town(int num_cit, std::string mayor, std::string country, std::string town) <<constructor>>+ Town(const Town &to_copy) <<copy constructor>>+ change_mayor(std::string new_mayor_name)+ change_num_of_cit(int new_num)+ howmanycit()+ whoismayor()+ whattown()+ Town_info()+ WriteXML(std::ostream& o)+ ~Town() <<destructor>>

Приложение Б

Код Town.h

```
#ifndef Town_h
#define Town_h

#include <string>

class Town{
private:
    int number_of_citizens;
    std::string mayor_name;
    std::string country_name;
    std::string town_name;

public:
    Town();

    Town(int num_cit, std::string mayor, std::string country, std::string town);

    Town(const Town &to_copy);

    void change_mayor(std::string new_mayor_name);

    void change_num_of_cit(int new_num);

    int howmanycit();

    std::string whoismayor();

    std::string whatcountry();

    std::string whattown();

    void Town_info();

    void WriteXML(std::ostream& o);

    ~Town();
};

#endif //Town_h
```

Код Town.cpp

```
#include "Town.h"
#include <iostream>
#include <string>
Town::Town(){ //constructor
    number_of_citizens = 1000000;
    mayor_name = "Sobyanin";
    country_name = "Russia";
    town_name = "Moscow";
}
Town::Town(int num_cit, std::string mayor, std::string country, std::string town) //second constructor
: number_of_citizens(num_cit), mayor_name(mayor), country_name(country), town_name(town) {}

Town::Town(const Town &to_copy){ //copy constructor
    if (to_copy.town_name != "") town_name = to_copy.town_name;
    if (to_copy.mayor_name != "") mayor_name = to_copy.mayor_name;
    if (to_copy.country_name != "") country_name = to_copy.country_name;
    number_of_citizens = to_copy.number_of_citizens;
}
void Town::change_mayor(std::string new_mayor_name){
    this -> mayor_name = new_mayor_name;
}
void Town::change_num_of_cit(int new_num){
    this -> number_of_citizens = new_num;
}

int Town::howmanycit() { return number_of_citizens; }

std::string Town::whoismayor() { return mayor_name; }

std::string Town::whatcountry() { return country_name; }

std::string Town::whattown() { return town_name; }
```

```

void Town::Town_info(){
    std::cout << "-----\n";
    std::cout << "Country: " << country_name << "\n";
    std::cout << "Town name: " << town_name << "\n";
    std::cout << "Mayor: " << mayor_name << "\n";
    std::cout << "Number of citizens: " << number_of_citizens << "\n";
    std::cout << "-----\n";
}

void Town::WriteXML(std::ostream& o){
    o << "<?xml version=\"1.0\" encoding=\"UTF-8\" ?>" << std::endl;
    o << "  <NumberOfCitizens =\" " << howmanycit() << "\" />" << std::endl;
    o << "  <MayorName =\" " << whoismayor() << "\" />" << std::endl;
    o << "  <CountryName =\" " << whatcountry() << "\" />" << std::endl;
    o << "  <TownName =\" " << whattown() << "\" />" << std::endl;
}

Town::~~Town(){} //destructor

```

Код Classes.cpp

```

#include <iostream>
#include <string>
#include "Town.h"

int main(){
    Town Moscow, town(5000, "kitten", "Norway", "Prehewill");
    Moscow.Town_info();
    Town Moscow2(Moscow);
    Moscow.change_mayor("C0olH4ck3r");
    Moscow.Town_info();
    Moscow2.Town_info();
    town.WriteXML(std::cout);
    Moscow.change_num_of_cit(2000000);
    Moscow.Town_info();
    town.Town_info();
    return 0;
}

```