

**Московский авиационный институт
(национальный исследовательский университет)**

Институт №8 «Информационные технологии и прикладная математика»
Кафедра 806 «Вычислительная математика и программирование»
Дисциплина «Объектно-ориентированное программирование»

Лабораторная работа №1
Тема: Простые классы на языке C++

Студент: Инютин М. А.
Группа: М8О-207Б-19
Преподаватель: Чернышев Л. Н.
Дата:
Оценка:

Постановка задачи

Изучить систему сборки на языке C++, систему контроля версий и основы работы с классами в C++.

Вариант 19. Создать класс **Adress** для работы с адресами домов. Адрес должен состоять из строк с названием города и улицы и чисел с номером дома и квартиры. Реализовать операции сравнения адресов, а также операции проверки принадлежности адреса к улице и городу. В операциях не должен учитываться регистр строки. Так же необходимо сделать операцию, которая возвращает истину, если два адреса находятся по соседству (на одной улице в одном городе и дома стоят подряд).

Описание программы

У класса **Adress** создадим четыре приватные переменные: 2 строки для названия города и улицы и 2 беззнаковых целых числа для номеров дома и квартиры. Чтобы не учитывать регистр строк, реализуем функцию сравнения строк с использованием `std::tolower`. Реализуем конструктор с параметрами и без, деструктор для объектов класса, методы сравнения адресов и принадлежности городу и улице. Чтобы было проще писать и читать код, вынесем реализацию методов класса в отдельный файл. В основном файле реализуем ввод и вывод данных, взаимодействие с методами класса.

Набор тестов

На ввод программе поступает две записи. В каждой по две строки и два беззнаковых целых числа, название города, улицы, номер дома и квартиры соответственно.

Тест №1

Moscow LeningradskyProspect 14 12

Moscow LeningradskyProspect 12 12

Тест №2

Moscow LeningradskyProspect 5 108

Moscow LeningradskyProspect 5 191

Тест №3

Moscow Lenina 12 10

Stavropol Lenina 12 10

Тест №4

Stavropol Dzerzhinskogo 86 12

Stavropol Dzerzhinskogo 88 9

Тест №5

Stavropol Dzerzhinskogo 86 12

Stavropol Dzerzhinskogo 90 12

Тест №6

Moscow TsArEvA 12 812

Moscow tSaReVa 12 812

Результаты выполнения тестов

Программа отвечает на три запросы: равны ли адреса, одинаковы ли город и улица у двух адресов, находятся ли адреса по соседству.

Тест №1

```
cat test1.txt | ./oop_exercise_01
```

Equal? NO

Same city and street? YES

Neighbors? YES

Тест №2

```
cat test2.txt | ./oop_exercise_01
```

Equal? NO

Same city and street? YES

Neighbors? YES

Тест №3

```
cat test3.txt | ./oop_exercise_01
```

Equal? NO

Same city and street? NO

Neighbors? NO

Тест №4

```
cat test4.txt | ./oop_exercise_01
```

Equal? NO

Same city and street? YES

Neighbors? YES

Тест №5

```
cat test5.txt | ./oop_exercise_01
```

Equal? NO

Same city and street? YES

Neighbors? NO

Тест №6

```
cat test6.txt | ./oop_exercise_01
```

Equal? YES

Same city and street? YES

Neighbors? YES

Листинг программы

Программа представлена в трёх файлах: объявление класса в `adress.hpp`, реализация методов в `adress.cpp` и взаимодействие с объектами класса в `main.cpp`.

adress.hpp

```
#ifndef ADDRESS_HPP
#define ADDRESS_HPP

#include <string>

class Address {
private:
    std::string City, Street;
    unsigned int House, Flat;
public:
    std::string GetCity();
    std::string GetStreet();
    Address() : City(), Street(), House(), Flat() {};
    Address(const std::string &city, const std::string
&street, const unsigned int &house, const unsigned int &flat) :
City(city), Street(street), House(house), Flat(flat) {};
    ~Address() {};

    static bool CompareString(const std::string &lhs,
                             const std::string &rhs);
    static bool IsEqual(const Address &lhs,
                        const Address &rhs);
    static bool IsInCity(const Address &address,
                         const std::string &city,
                         const std::string &street);
    static bool Neighbors(const Address &lhs,
                          const Address &rhs);
};

#endif // ADDRESS_HPP
```

adress.cpp

```
#include "address.hpp"

std::string Address::GetCity() {
    return this->City;
}

std::string Address::GetStreet() {
    return this->Street;
}

// Returns true if string are equal (ignores lowercase)
bool Address::CompareString(const std::string &lhs,
                           const std::string &rhs) {
    if (lhs.size() == rhs.size()) {
        for (int i = 0; i < lhs.size(); ++i) {
            if (std::tolower(lhs[i]) != std::tolower(rhs[i])) {
                return false;
            }
        }
        return true;
    } else {
        return false;
    }
}

bool Address::IsEqual(const Address &lhs, const Address &rhs) {
    return CompareString(lhs.City, rhs.City) &&
        CompareString(lhs.Street, rhs.Street) && (lhs.House == rhs.House) &&
        (lhs.Flat == rhs.Flat);
}

bool Address::IsInCity(const Address &address,
                      const std::string &city, const std::string &street) {
    return CompareString(address.City, city) &&
        CompareString(address.Street, street);
}

bool Address::Neighbors(const Address &lhs, const Address &rhs) {
    return CompareString(lhs.City, rhs.City) &&
        CompareString(lhs.Street, rhs.Street) && (std::abs((int)lhs.House -
        (int)rhs.House) == 2 or rhs.House == lhs.House);
}
```

main.cpp

```
#include <bits/stdc++.h>
#include "adress.hpp"
/*
 * Инютин М А М80-207Б-19
 * Создать класс Address для работы с адресами домов. Адрес должен
 * состоять из строк с названием города и улицы и чисел с номером
 * дома и квартиры. Реализовать операции сравнения адресов, а
 * также операции проверки принадлежности адреса к улице и городу.
 * В операциях не должен учитываться регистр строки. Так же
 * необходимо сделать операцию, которая возвращает истину если два
 * адреса находятся по соседству (на одной улице в одном городе
 * и дома стоят подряд).
 */

signed main () {
    std::string city, street;
    unsigned int house = 0, flat = 0;
    std::cout << "Input first adress city: ";
    std::cin >> city;
    std::cout << "Input first adress street: ";
    std::cin >> street;
    std::cout << "Input first adress house number: ";
    std::cin >> house;
    std::cout << "Input first adress flat numer: ";
    std::cin >> flat;
    Adress a = Adress(city, street, house, flat);
    house = 0, flat = 0;
    std::cout << "Input second adress city: ";
    std::cin >> city;
    std::cout << "Input second adress street: ";
    std::cin >> street;
    std::cout << "Input second adress house number: ";
    std::cin >> house;
    std::cout << "Input second adress flat numer: ";
    std::cin >> flat;
    Adress b = Adress(city, street, house, flat);
    std::cout << std::endl;
    if (house * flat == 0) {
        std::cout << "Invalid input! Programm terminates with
exit code -1\n";
        return -1;
    }
    std::cout << "Equal? " << (Adress::IsEqual(a, b) ? "YES" :
"NO") << "\n";
    std::cout << "Same city and street? " << (Adress::IsInCity(a,
b.GetCity(), b.GetStreet()) ? "YES" : "NO") << "\n";
    std::cout << "Neighbors? " << (Adress::Neighbors(a, b) ?
"YES" : "NO") << "\n";
    return 0;
}
```

Выводы

Я научился создавать классы, описывать их члены и методы на языке C++, познакомился с автоматической системой сборки CMake, вводом и выводом из библиотеки `iostream` и строками из библиотеки `string`.

Список литературы

1. ООП в картинках / Хабр — Habr
URL: <https://habr.com/ru/post/463125/> (дата обращения: 27.09.2020)