**Московский авиационный институт**

**(Национальный исследовательский университет)**

Факультет: «Информационные технологии и прикладная математика»

Кафедра: 806 «Вычислительная математика и программирование»

Дисциплина: «Объектно-ориентированное программирование»

**Лабораторная работа № 2**

Тема:Операторы, литералы С++

Студент: Абдыкеримов Урмат

Группа: 80-206

Преподаватель: Чернышов Л.Н.

Дата:

Оценка:

Москва, 2020

1. Постановка задачи  
   *Создать класс Address для работы с адресами домов. Адрес должен состоять из строк с названиями города и улицы и чисел с номером дома и квартиры. Реализовать операции сравнения адресов, а также операции принадлежности адреса улице или городу. В операциях не должен учитываться регистр строки. Также необходимо сделать операцию, которая возвращает истину если два адреса находятся по соседству(один город, улица, дома стоят подряд).*  
   *Операцию сравнения равенства реализовать в виде перегрузке оператора. Операцию нахождения по соседству реализовать в виде перегрузки оператора &.  
   Реализовать пользовательский литерал для работы с константами Address.*
2. Описание программы

*В программе реализован класс Address с полями street, city, house, apartment, а также все методы описанные в постановке задачи.*

*Пользовательский литерал для Address принимает на вход строку в виде “ город , улица , дом , квартира” и возвращает адрес с этими параметрами.*

*В конструкторе Address все буквы street и city приводятся к нижнему регистру.*

1. Набор тестов

*test\_01:  
london baker 26 20*

*london*

*baker*

*test\_02:  
london baker 25 1*

*LONDON*

*BakEr*

*test\_03:  
london baker 26 13*

*lndn*

*bkr*

1. Результаты выполнения тестов

*test\_01:*First address is london baker 26 13

Enter second address

london baker 26 20

Second address is london baker 26 20

Equal : 0

Enter city to check

london

Is address in city "london" 1

Enter street to check

baker

Is address in street "baker" 1

Two addresses are near 0

*test\_02:*First address is london baker 26 13

Enter second address

london baker 25 1

Second address is london baker 25 1

Equal : 0

Enter city to check

LONDON

Is address in city "LONDON" 0

Enter street to check

BAkEr

Is address in street "BAkEr" 0

Two addresses are near 1

*test\_03:*First address is london baker 26 13

Enter second address

london baker 26 13

Second address is london baker 26 13

Equal : 1

Enter city to check

lndn

Is address in city "lndn" 0

Enter street to check

bkr

Is address in street "bkr" 0

Two addresses are near 0

1. Листинг программы

//Абдыкеримов Урмат Эмирбекович М8О-206Б-19

#include <iostream>

#include <cmath>

#include <algorithm>

using *namespace* std;

//строку нижнему регистру

*void* toLower(string& *s*) {

  for(*auto*& it: *s*) {

    it = tolower(it);

  }

}

*class* Address {

*public:*

  Address();

  Address(string, string, *int*, *int*);

  Address(const Address&);

*bool* inCity(string);

*bool* inStreet(string);

*bool* operator== (const Address&);

*bool* operator& (const Address&);

*void* print();

*private:*

  string city;

  string street;

*int* house;

*int* apartment;

};

//конструкторы

Address::Address() {

  city = "";

  street = "";

  house = 0;

  apartment = 0;

}

Address::Address(string *c*, string *s*, *int* *h*, *int* *a*) {

  toLower(*c*);

  toLower(*s*);

  city = *c*;

  street = *s*;

  house = *h*;

  apartment = *a*;

}

Address::Address(const Address& *rhs*) {

  city = *rhs*.city;

  street = *rhs*.street;

  house = *rhs*.house;

  apartment = *rhs*.apartment;

}

*bool* Address::inCity(string *c*) {

  return (*c* == city);

}

*bool* Address::inStreet(string *s*) {

  return (*s* == street);

}

//оператор сравнения

*bool* Address::operator== (const Address& *val*) {

  return (*val*.city == city && street == *val*.street && house == *val*.house && apartment == *val*.apartment);

}

//оператор нахождения рядом

*bool* Address::operator& (const Address& *val*) {

  return (*val*.city == city && street == *val*.street && abs(house - *val*.house) == 1);

}

*void* wht\_clr(string& *s*) {

  while(\**s*.begin() == ' ') {

*s*.erase(0, 1);

  }

  while(\**s*.rbegin() == ' ') {

*s*.erase(*s*.size() - 1, 1);

  }

}

*void* Address::print() {

  cout << city << " " << street << " " << house << " " << apartment << endl;

}

//пользовательский литерал

Address operator"" \_adrs(const *char*\* *val*, *unsigned*) {

  string fl = *val*;

  wht\_clr(fl);

  string city = fl.substr(0, fl.find(","));

  toLower(city);

  fl.erase(0, fl.find(",") + 1);

  wht\_clr(city);

  wht\_clr(fl);

  string street = fl.substr(0, fl.find(","));

  toLower(street);

  fl.erase(0, fl.find(",") + 1);

  wht\_clr(street);

  wht\_clr(fl);

*int* house = atoi(fl.substr(0, fl.find(",")).c\_str());

  fl.erase(0, fl.find(",") + 1);

  wht\_clr(city);

  wht\_clr(fl);

*int* apartment = atoi(fl.c\_str());

  return Address(city, street, house, apartment);

}

main() {

  Address a1 = "London,Baker,   26,     13"\_adrs;

  cout << "First address is ";

  a1.print();

  string city, street;

*int* house, apartment;

  cout << "Enter second address " << endl;

  cin >> city >> street >> house >> apartment;

  Address a2(city, street, house, apartment);

  cout << "Second address is ";

  a2.print();

  cout << "Equal : " << (a2 == a1) << endl;

  cout << "Enter city to check" << endl;

  cin >> city;

  cout << "Is address in city \"" << city << "\" " << a2.inCity(city) << endl;

  cout << "Enter street to check" << endl;

  cin >> street;

  cout << "Is address in street \"" << street << "\" " << a2.inStreet(street) << endl;

  cout << "Two addresses are near " << (a1 & a2) << endl;

}