RМинистерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ   
СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)

КАФЕДРА АВТОМАТИЗИРОВАННЫХ СИСТЕМ УПРАВЛЕНИЯ

**ОТЧЁТ**

ПО ЛАБОРАТОРНОЙ РАБОТЕ №3

по дисциплине «Объектно-ориентированное программирование»

|  |  |
| --- | --- |
|  | Обучающийся гр. \_\_\_\_\_\_\_442-1\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Кадочников А.В.  (подпись) (И.О. Фамилия)  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (дата) |
| **\_\_\_\_\_\_\_\_\_**  (оценка) | Принял  Доцент кафедры АСУ, к.т.н.  (должность, ученая степень, звание)  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Алферов С. М.  (подпись) (И.О. Фамилия)  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (дата) |

Томск 2023

Оглавление

[1 Цель работы 3](#_Toc147752467)

[2 Задание 3](#_Toc147752468)

[3 Диаграмма классов 3](#_Toc147752469)

[4 Текст программы 4](#_Toc147752470)

[5 Результат работы 6](#_Toc147752471)

[6 Выводы 7](#_Toc147752472)

1 Цель работы

Научиться создавать классы с учетом правил предметной области. Уяснить необходимость скрытия некоторых свойств класса модификаторами private и protected.

2 Задание

Создать класс, имеющий заданные свойства, обеспечить доступ к свойствам (запись и чтение значений свойств) с учетом заданных ограничений. Все конструкторы класса так же должны обеспечивать создание объекта с учетом заданных ограничений. Написать программу, демонстрирующую сохранение ограничений при любых действиях внешней программы над объектом.

Вариант 2

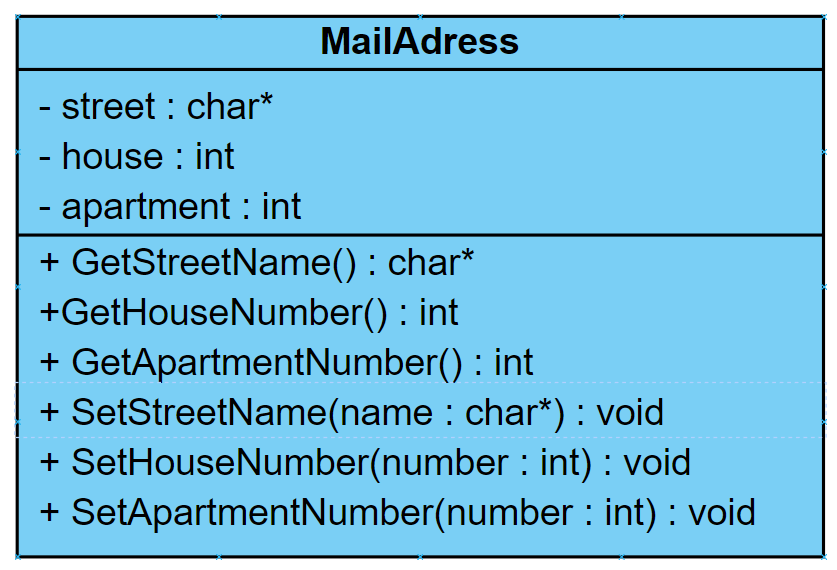
Класс: почтовый адрес

Свойства класса: Улица, дом, квартира

Правила ПО: Первые символы до пробела свойства «улица» указывают на тип. «ул» - улица, «пр» - проспект, «пер» - переулок. На переулке номера домов лежат в диапазоне от 1 до 30, на улице от 1 до 100, на проспекте от 1 до 1000. Если дом частный, то квартира не указывается.

3 Диаграмма классов

Диаграмма класса MailAdress представлена ниже(Рисунок 1).

Рисунок 1 Диаграмма класса

4 Текст программы

//Вариант 2

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <iostream>

#include <Windows.h>

#include <string>

class MailAdress

{

public:

char\* GetStreetName();

int GetHouseNumber();

int GetApartmentNumber();

void SetStreetName(char\* name);

void SetHouseNumber(int number);

void SetApartmentNumber(int number);

private:

char\* street = new char[100];

int house = 0;

int apartment = 0;

};

char\* MailAdress::GetStreetName() {

return this->street;

}

int MailAdress::GetHouseNumber() {

return this->house;

}

int MailAdress::GetApartmentNumber() {

return this->apartment;

}

void MailAdress::SetStreetName(char\* name) {

strcpy(street, name);

}

void MailAdress::SetHouseNumber(int number) {

house = number;

}

void MailAdress::SetApartmentNumber(int number) {

apartment = number;

}

void newMailAdress(MailAdress& Name) {

char\* newStreet = new char[100];

int newHouse = NULL;

int fHouse = 0;

int newApartment = NULL;

int fApartment = 0;

while (true) {

fseek(stdin, 0, SEEK\_END);

std::cout << "Введите имя улицы: ";

fgets(newStreet, 100, stdin);

newStreet[strcspn(newStreet, "\n")] = 0;

Name.SetStreetName(newStreet);

char\* temp = strtok(newStreet, " ");

int counter = 0;

while (temp != NULL) {

if (!strcmp(temp, "ул") && counter < 1) {

fHouse = 1;

}

if (!strcmp(temp, "пр") && counter < 1) {

fHouse = 2;

}

if (!strcmp(temp, "пер") && counter < 1) {

fHouse = 3;

}

counter++;

temp = strtok(NULL, " ");

};

if (fHouse != 0) break;

else std::cout << "Вы ввели неверное название улицы" << std::endl;

}

delete[] newStreet;

while (true) {

std::cout << "Введите номер дома: ";

std::cin >> newHouse;

if (fHouse == 1 && (newHouse > 100 || newHouse < 1)) {

std::cout << "номер дома на улице не может быть меньше 1 или больше 100" << std::endl;

}

else if (fHouse == 2 && (newHouse > 1000 || newHouse < 1)) {

std::cout << "номер дома на проспекте не может быть меньше 1 или больше 1000" << std::endl;

}

else if (fHouse == 3 && (newHouse > 30 || newHouse < 1)) {

std::cout << "номер дома в переулке не может быть меньше 1 или больше 30" << std::endl;

}

else break;

}

Name.SetHouseNumber(newHouse);

std::cout << "Этот дом частный? (д/Н): ";

char ch = 0;

std::cin >> ch;

switch (ch)

{

case 'д':

Name.SetApartmentNumber(-1);

break;

default:

std::cout << "Введите номер квартиры: ";

std::cin >> newApartment;

Name.SetApartmentNumber(newApartment);

break;

}

}

void outMailAdress(MailAdress& Name) {

std::cout << "Улица: " << Name.GetStreetName() << std::endl;

std::cout << "Дом: " << Name.GetHouseNumber() << std::endl;

if (Name.GetApartmentNumber() < 0) {

std::cout << "Частный дом" << std::endl;

}

else {

std::cout << "Квартира: " << Name.GetApartmentNumber() << std::endl;

}

}

int main()

{

SetConsoleOutputCP(1251);

SetConsoleCP(1251);

MailAdress Oleg;

MailAdress Anton;

newMailAdress(Oleg);

newMailAdress(Anton);

std::cout << std::endl;

std::cout << std::endl;

std::cout << "Введённые данные" << std::endl;

outMailAdress(Oleg);

std::cout << std::endl;

std::cout << std::endl;

outMailAdress(Anton);

return 0;

}

5 Результат работы

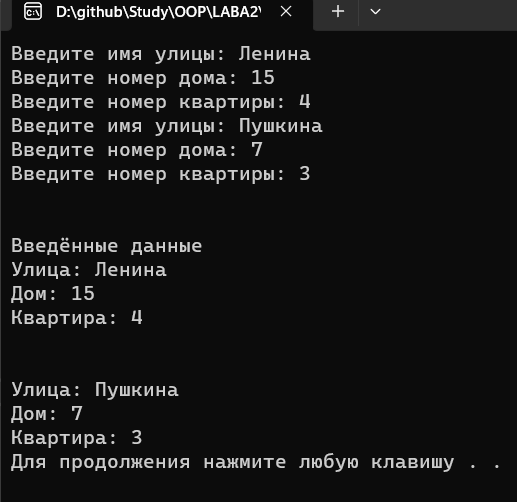
Результат работы программы приведён ниже(Рисунок 2)

Рисунок 2Результат работы программы

6 Выводы

В процессе работы над данной лабораторной работой я научился создавать классы с учетом правил предметной области. Уяснить необходимость скрытия некоторых свойств класса модификаторами private и protected.