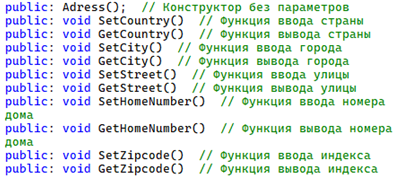
«Отчет по П. 3.2 Простейшие классы, одиночное наследование» Никита \*\*\* \*\*\*. Вариант 4.

Построить описание класса, содержащего информацию о почтовом адресе организации. Предусмотреть возможность раздельного изменения составных частей адреса и проверки допустимости вводимых значений. В случае недопустимых значений полей выбрасываются исключения. Написать программу, демонстрирующую все разработанные элементы класса.

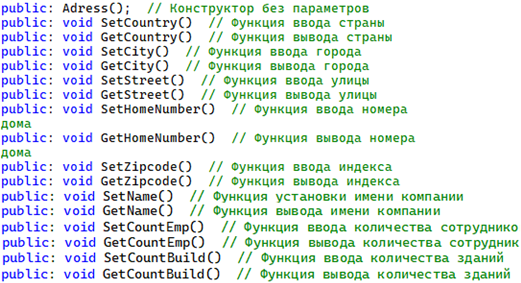
Создать дочерний класс, который содержит информацию о данной организации – наименование, количество сотрудников и т.п. Предусмотреть возможность раздельного изменения наименования организации и числа сотрудников и проверки допустимости вводимых значений.

1. Постановка задачи:

Функции класса Adress:



Функции класса Company:



Входные данные + ограничения:

А) Переменная Country. Тип string. Название страны.

Исключения: нет.

Б) Переменная City. Тип string. Название города.

Исключения: нет.

В) Переменная Street. Тип string. Название улицы.

Исключения: нет.

Г) Переменная Zipcode. Тип int. Индекс.

Исключения: буквы, строка, отрицательные числа, дробные числа

Д) Переменная HomeNumber. Тип int. Номер дома.

Исключения: буквы, строка, отрицательные числа, дробные числа

Е) Переменная Name. Тип string. Название компании.

Исключения: нет.

Ё) Переменная CountEmp. Тип int. Количество работников.

Исключения: буквы, строка, отрицательные числа, дробные числа

Ж) Переменная CountBuild. Тип int. Количество зданий.

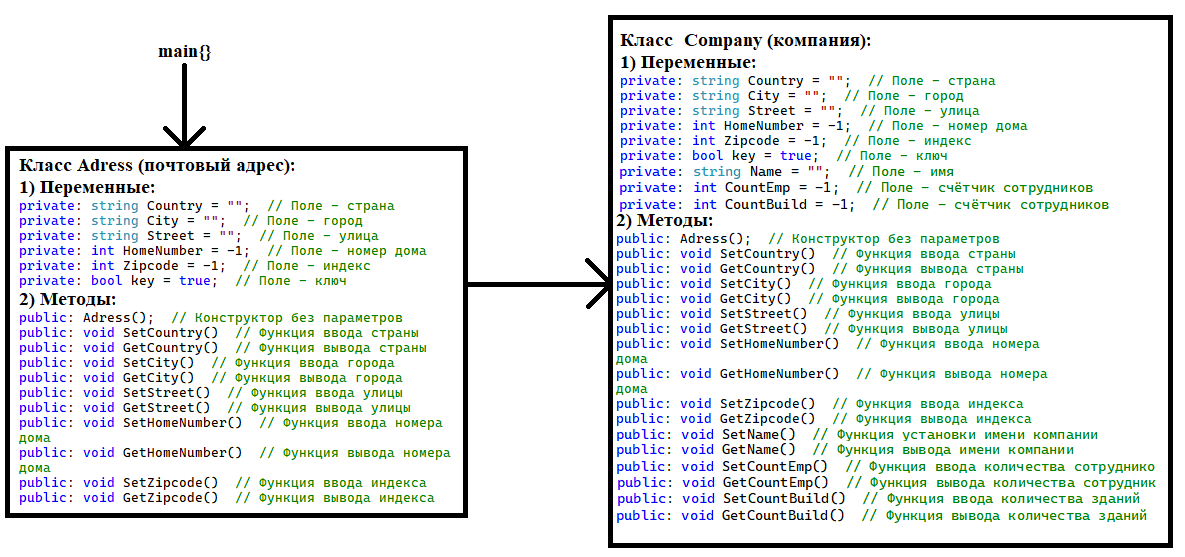
Исключения: буквы, строка, отрицательные числа, дробные числа

Промежуточные данные + ограничения:

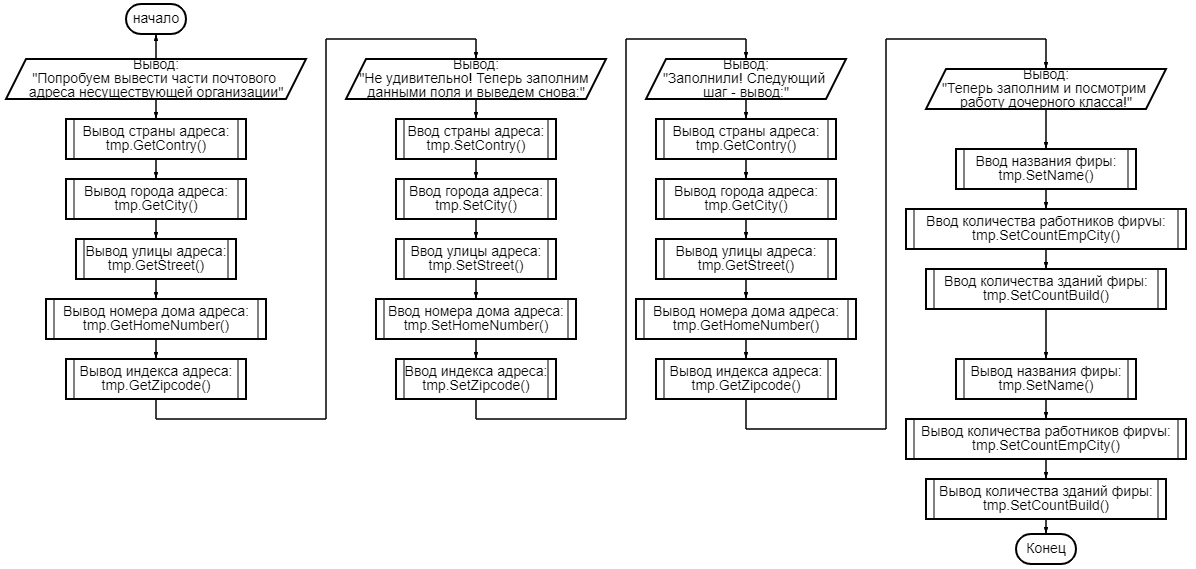
Среда разработки: Visual Studio 2022.

2. Разработка программы:

2.1 Описание классов, полей и методов классов.



2.2Разработка схемы алгоритма.



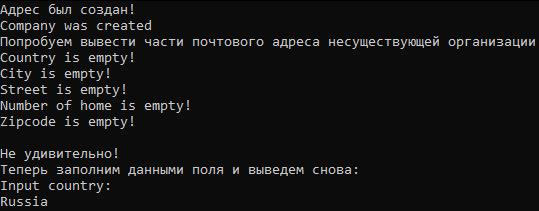
2.3 Описание пользовательского интерфейса.

«Взаимодействие пользователя с программой осуществляется с помощью диалога».

Примеры диалогов:

1. Ввод названия страны адреса:

Вводимое значение – “Russia”



1. Ввод название города адреса:

Вводимое значение – “Moscow”



1. Ввод названия улицы адреса:

Вводимое значение – “Lomonosova”



1. Ввод номера дома адреса:

Вводимое значение – “abc”



Вводимое значение – 5



Вводимое значение – 5.5



1. Ввод индекса адреса:

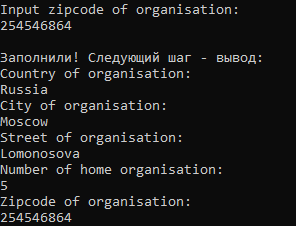
Вводимое значение – “abc”



Вводимое значение – 5.5

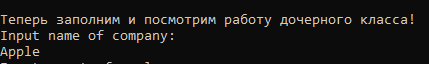


Вводимое значение - 254546864



1. Ввод названия фирмы:

Вводимо значение Apple



1. Ввод количества работников:

Вводимое значение – 50



Вводимое значение – “zxc”

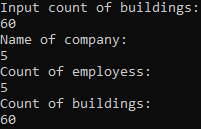


Вводимое значение – 2,25

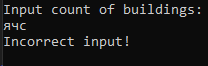


1. Ввод количества зданий у компании:

Вводимое значение – 60



Вводимое значение – “ячс”



Вводимое значение – 666,5



3. Реализация и тестирование программы

**3.1. Описание разработанной программы**

Для отцовского класса у нас определены методы для установления и получения основных необходимых полей класса (Имя страны адреса, город адреса, улица адреса, номер дома адреса и индекс адреса).

Для дочерного класса у нас определены всё те же методы, но добавлены поля с методы связанные с: названием компании, количеством работником компании, количеством зданий компании.

В функции main мы создаём представителя дочерного класса. Затем поочерёдно вызываем функции вывода основных полей представителя класса. Получаем ответы, что все поля пустые (что и должно быть в самом начале программы). Потом поочерёдно вызываем функции для записи информации в основные поля класса. После чего выводим значения всех полей в консоль.

Дальше мы работаем с методами дочерного класса. Вызываем методы для установки значений полей дочерного класса и выводим с помощью иных методов установленные нами значения.

**3.2. Тестирование программы**

|  |  |  |
| --- | --- | --- |
| Входные данные | Результат программы | Смысл |
| tmp.SetCountry()  (Russia) | Russia | Проверка на корректность работы программы |
| tmp.SetCity()  (Moscow) | Moscow | Проверка на корректность работы программы |
| tmp.SetStreet()  (Lomonosova) | Lomonosova | Проверка на корректность работы программы |
| tmp.SetHomeNumber()  (5) | 5 | Проверка на корректность работы программы |
| tmp.SetHomeNumber()  (Lomonosova) | Incorrect input! | Проверка на строку |
| tmp.SetHomeNumber()  (5,5) | Incorrect input! | Проверка на дробное число |
| tmp.SetZipcode()  (5,5) | Incorrect input! | Проверка на дробное число |
| tmp.SetZipcode()  (546641671) | 546641671 | Проверка на корректность работы программы |
| tmp.SetZipcode()  (dsadadas) | Incorrect input! | Проверка на строку |
| tmp.SetName()  (Apple) | Apple | Проверка на корректность работы программы |
| tmp.SetCountEmp()  (Apple) | Incorrect input! | Проверка на строку |
| tmp.SetCountEmp()  (228) | 228 | Проверка на корректность работы программы |
| tmp.SetCountEmp()  (1,1) | Incorrect input! | Проверка на дробное число |
| tmp.SetCountBuild()  (322) | 332 | Проверка на корректность работы программы |
| tmp.SetCountBuild()  (dogte) | Incorrect input! | Проверка на строку |
| tmp.SetCountBuild()  (1,1) | Incorrect input! | Проверка на дробное число |

Возможный диалог:

Ввод названия страны – ввод названия города – ввод названия улицы – ввод номера дома – ввод индекса – вывод – названия страны – вывод названия города – вывод названия улицы – вывод номера дома – вывод индекса – ввод названия компании – ввод количества работников – ввод количество сооружений – вывод названия компании – вывод количества работников – вывод количество сооружений.

Код:

#include <iostream> // Стандартная библиотека ввода-вывода

#include <Windows.h> // Библиотека консольных настроек

using namespace std; // Использование стандартного пространства имён

void InputINT(int\*); // Функция ввода целых чисел с проверкой

class Adress // Класс адреса

{

private: // Приватные поля

string Country = ""; // Поле-страна

string City = ""; // Поле-город

string Street = ""; // Поле-улица

int HomeNumber = -1; // Поле-номер дома

int Zipcode = -1; // Поле-индекс

public: // Публичные методы

Adress() // Описание конструктор без аргументов

{

cout << "Адрес был создан!" << endl;

}

void SetCountry() // Описание метода установки страны

{

cout << "Input country:" << endl;

cin >> Country;

}

void GetCountry() // Описание метода выводаа города

{

if (Country != "")

{

cout << "Country of organisation:" << endl;

cout << Country << endl;

}

else

{

cout << "Country is empty!" << endl;

}

}

void SetCity() // Описание метода установки города

{

cout << "Input city:" << endl;

cin >> City;

}

void GetCity() // Описание метода вывода города

{

if (City != "")

{

cout << "City of organisation:" << endl;

cout << City << endl;

}

else

{

cout << "City is empty!" << endl;

}

}

void SetStreet() // Описание метода установки улицы

{

cout << "Input Street:" << endl;

cin >> Street;

}

void GetStreet() // Описание метода вывода улицы

{

if (Street != "")

{

cout << "Street of organisation:" << endl;

cout << Street << endl;

}

else

{

cout << "Street is empty!" << endl;

}

}

void SetHomeNumber() // Описание метода установки номера дома

{

cout << "Input number of home:" << endl;

InputINT(&HomeNumber);

}

void GetHomeNumber() // Описание метода вывода номера дома

{

if (HomeNumber != -1)

{

cout << "Number of home organisation:" << endl;

cout << HomeNumber << endl;

}

else

{

cout << "Number of home is empty!" << endl;

}

}

void SetZipcode() // Описание метода установки индекса

{

cout << "Input zipcode of organisation:" << endl;

InputINT(&Zipcode);

}

void GetZipcode() // Описание метода вывода индекса

{

if (Zipcode != -1)

{

cout << "Zipcode of organisation:" << endl;

cout << Zipcode << endl;

}

else

{

cout << "Zipcode is empty!" << endl;

}

}

};

class Company : public Adress // Класс-наследник компания

{

private: // Приватные поля

string Name = ""; // Поле название компании

int CountEmp = -1; // Поле количество рабочих компании

int CountBuild = -1; // Поле количества зданий компании

public:

Company() // Конструктор без параметров

{

cout << "Company was created" << endl;

}

void SetName() // Описание метода установки названия компании

{

cout << "Input name of company:" << endl;

cin >> Name;

}

void GetName() // Описание метода вывода названия компании

{

if (Name != "")

{

cout << "Name of company:" << endl;

cout << Name << endl;

}

else

{

cout << "Name is empty!" << endl;

}

}

void SetCountEmp() // Описание метода установки количества работников компании

{

cout << "Input count of employees:" << endl;

InputINT(&CountEmp);

}

void GetCountEmp() // Описание метода вывода количества работников компании

{

if (CountEmp != -1)

{

cout << "Count of employess:" << endl;

cout << CountEmp << endl;

}

else

{

cout << "Count is empty!" << endl;

}

}

void SetCountBuild() // Описание метода установки количества зданий компании

{

cout << "Input count of buildings:" << endl;

InputINT(&CountBuild);

}

void GetCountBuild() // Описание метода вывода количества зданий компании

{

if (CountBuild != -1)

{

cout << "Count of buildings:" << endl;

cout << CountBuild << endl;

}

else

{

cout << "Count is empty!" << endl;

}

}

};

int main(int argc, char\* argv[])

{

SetConsoleCP(1251); // Установка настроек консоли вывода

SetConsoleOutputCP(1251); // Установка формы консоли вывода

setlocale(LC\_ALL, "Russian"); // Установка русского языка в консоле

Company tmp; // Создание представителя дочерного класса

cout << "Попробуем вывести части почтового адреса несуществующей организации" << endl;

tmp.GetCountry(); // Вызов метода вывода страны

tmp.GetCity(); // Вызов метода вывода города

tmp.GetStreet(); // Вызов метода вывода улицы

tmp.GetHomeNumber(); // Вызов метода вывода номера дома

tmp.GetZipcode(); // Вызов метода вывода индекса

cout << "\nНе удивительно! \nТеперь заполним данными поля и выведем снова:" << endl;

tmp.SetCountry(); // Вызов метода установки страны

tmp.SetCity(); // Вызов метода установки города

tmp.SetStreet(); // Вызов метода установки улицы

tmp.SetHomeNumber(); // Вызов метода установки номера дома

tmp.SetZipcode(); // Вызов метода установки индекса

cout << "\nЗаполнили! Следующий шаг - вывод:" << endl;

tmp.GetCountry(); // Вызов метода вывода страны

tmp.GetCity(); // Вызов метода вывода города

tmp.GetStreet(); // Вызов метода вывода улицы

tmp.GetHomeNumber(); // Вызов метода вывода номера дома

tmp.GetZipcode(); // Вызов метода вывода индекса

cout << "\nТеперь заполним и посмотрим работу дочерного класса!" << endl;

tmp.SetName(); // Вызов метода установки названия компании

tmp.SetCountEmp(); // Вызов метода установки количества сотрудников компании

tmp.SetCountBuild(); // Вызов метода установки количества зданий компании

tmp.GetName(); // Вызов метода вывода названия компании

tmp.GetCountEmp(); // Вызов метода вывода количества сотрудников компании

tmp.GetCountBuild(); // Вызов метода вывода количества зданий компании

return 0;

}

void InputINT(int\* tmp) // Описание функции ввода целых чисел с проверкой

{

while (!(cin >> \*tmp) || (cin.peek() != '\n') || (\*tmp <= 0))

{

cin.clear();

while (cin.get() != '\n');

cout << "Incorrect input!" << endl;

}

}