«Отчет по П. 3.2 Простейшие классы, одиночное наследование» Никита Фролов \*\*\*. Вариант 22.

Создать класс животное, имеющий классификацию (указатель на строку), число конечностей, число потомков. Определить конструкторы, деструктор и функцию печати.

Создать public-производный класс - домашнее животное, имеющий кличку. Определить конструкторы по умолчанию и с разным числом параметров, деструкторы, функцию печати. Определить функции переназначения названия клички и числа потомков.

В случае недопустимых значений полей выбрасываются исключения. Написать программу, демонстрирующую все разработанные элементы класса.

1. Постановка задачи:

Функции класса Animal:

public: Animal (); // Конструктор без параметров

public: ~Animal() // Деструктор

public: void SetCountLimbs(); // Метод установки количества конечностей

public: void SetClass(); // Метод установка класса у животного

public: void GetClass(); // Метод получение класса животного

public: void GetCountLimbs(); // Метод получение количества конечностей животного

public: void SetCountDescendants()l // Метод установки предков животного

public: void GetCountDescendants(); // Метод получения количества предков животного

Функции класса HomeAnimal:

public: HomeAnimal(); // Конструктор без параметров

public: HomeAnimal(string Class); // Конструктор с параметром

public: void GetNickname(); // Метод получения клички животного

public: void SetNickname(); // Метод установки клички животного

public: ~HomeAnimal() // Деструктор

Входные данные + ограничения:

А) Переменная Class. Тип string. Название класса.

Исключения: нет.

Б) Переменная CountLimbs. Тип int. Количество конечностей у животного.

Исключения: буквы, строка, отрицательные числа, дробные числа

В) Переменная CountDescendants. Тип int. Количество предков у животного.

Исключения: буквы, строка, отрицательные числа, дробные числа

Промежуточные данные + ограничения:

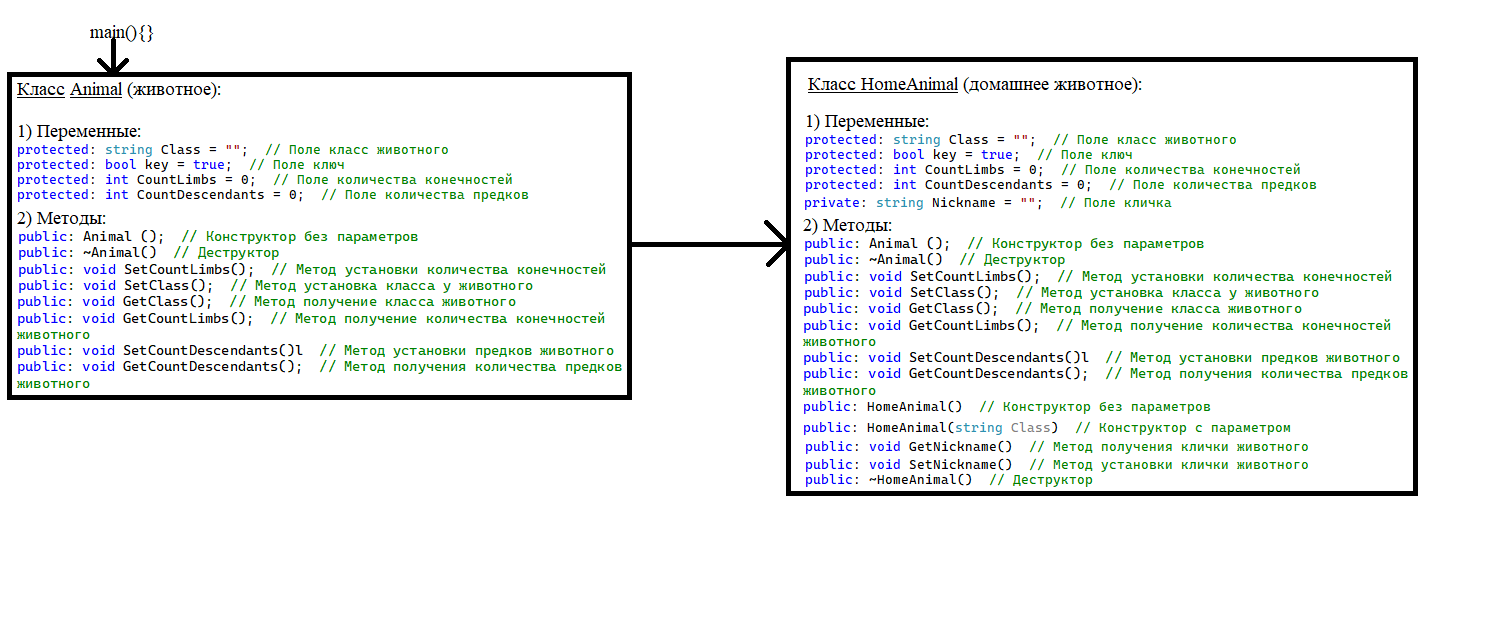
А) Переменная key. Тип bool. Ключ ошибок.

Исключения: буквы, числа, дробные, отрицательные числа, строки.

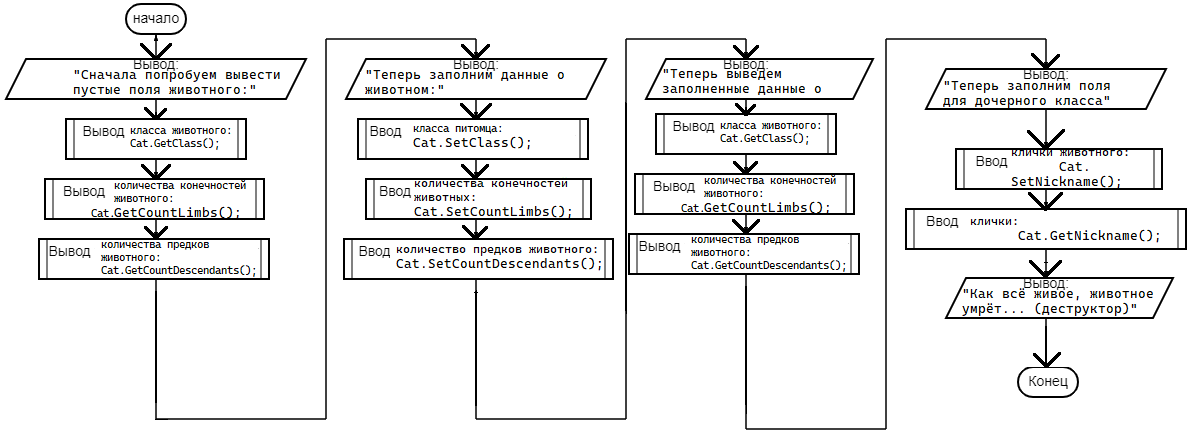
Среда разработки: Visual Studio 2022.

2. Разработка программы:

2.1 Описание классов, полей и методов классов.



2.2Разработка схемы алгоритма.



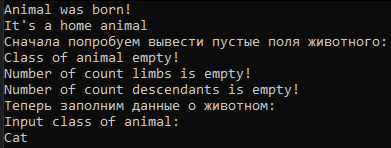
2.3 Описание пользовательского интерфейса.

«Взаимодействие пользователя с программой осуществляется с помощью диалога».

Примеры диалогов:

1. Ввод класса животного:

Вводимое значение – “Cat”



1. Ввод количества конечностей животного:

Вводимое значение – “abc”



Вводимое значение – 5

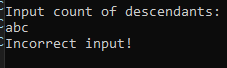


Вводимое значение – 5.5



1. Ввод количества предков животного:

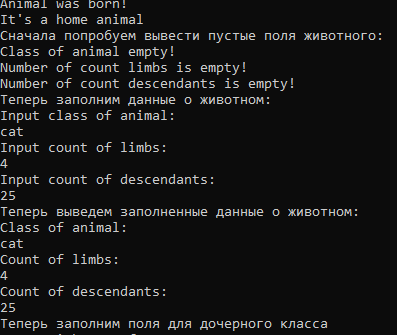
Вводимое значение – “abc”



Вводимое значение – 5.5

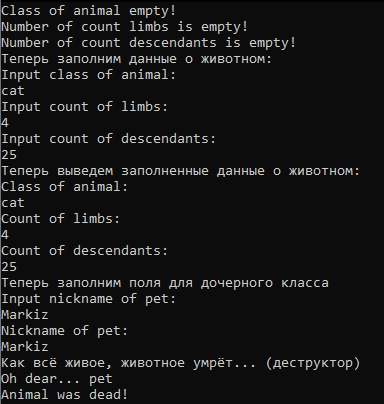


Вводимое значение - 25



1. Ввод названия фирмы:

Вводимое значение - Markiz



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 (Класс животного, количества конечностей, количества предков животного).

Для дочерного класса у нас определены всё те же методы, но добавлены поля с методы связанные с кличкой животного.

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 |
| Cat.SetClass();  (Cat) | Cat | Проверка на корректность работы программы |
| Cat.SetCountLimbs();  (5) | 5 | Проверка на корректность работы программы |
| Cat.SetCountLimbs();  (beb) | Incorrect input! | Проверка на строку |
| Cat.SetCountLimbs();  (5.5) | Incorrect input! | Проверка на дробное число |
| Cat.SetCountDescendants();  (2) | 2 | Проверка на корректность работы программы |
| Cat.SetCountDescendants();  (beb) | Incorrect input! | Проверка на строку |
| Cat.SetCountDescendants();  (3.5) | Incorrect input! | Проверка на дробное число |
| Cat.SetNickname();  (Markiz) | Markiz | Проверка на корректность работы программы |

Возможный диалог:

Ввод названия класса животного – ввод количества конечностей животного – ввод количества предков животного – ввод клички животного – вывод класса животного – вывод количества конечностей животного – вывод количества предков животного – вывод клички животного.

Код:

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <iostream>

using namespace std;

class Animal

{

public: Animal () // Конструктор без параметров

{

cout << "Animal was born!" << endl;

}

public: ~Animal() // Деструктор

{

cout << "Animal was dead!" << endl;

}

protected: string Class = "";

protected: bool key = true;

protected: int CountLimbs = 0;

protected: int CountDescendants = 0;

public: void SetCountLimbs() // Метод установки количества конечностей

{

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

int \_CountLimbs = 0;

char ch = '@';

if ((scanf("%d%c", &\_CountLimbs, &ch) != 2) || (ch != '\n') || (\_CountLimbs < 0))

{

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

key = false;

exit(0);

}

else

{

CountLimbs = \_CountLimbs;

}

}

public: void SetClass() // Установка класса у животного

{

cout << "Input class of animal:" << endl;

cin >> Class;

}

public: void GetClass() // Получение класса животного

{

if (Class != "")

{

cout << "Class of animal:" << endl;

cout << Class << endl;

}

else

{

cout << "Class of animal empty!" << endl;

}

}

public: void GetCountLimbs() // Метод получение количества конечностей животного

{

if ((CountLimbs != 0) && (key == true))

{

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

cout << CountLimbs << endl;

}

else

{

cout << "Number of count limbs is empty!" << endl;

}

}

public: void SetCountDescendants() // Метод установки предков животного

{

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

int \_CountDescendants = 0;

char ch = '@';

if ((scanf("%d%c", &\_CountDescendants, &ch) != 2) || (ch != '\n') || (\_CountDescendants < 0))

{

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

key = false;

exit(0);

}

else

{

CountDescendants = \_CountDescendants;

}

}

public: void GetCountDescendants() // Метод получения количества предков животного

{

if ((CountDescendants != 0) && (key == true))

{

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

cout << CountDescendants << endl;

}

else

{

cout << "Number of count descendants is empty!" << endl;

}

}

};

class HomeAnimal: public Animal

{

private: string Nickname = "";

public: HomeAnimal() // Конструктор без параметров

{

cout << "It's a home animal" << endl;

}

public: HomeAnimal(string Class) // Конструктор с параметром

{

this->Class = Class;

cout << "It's a home animal!" << endl;

}

public: void GetNickname() // Метод получения клички животного

{

if (Nickname != "")

{

cout << "Nickname of pet:" << endl;

cout << Nickname << endl;

}

else

{

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

}

}

public: void SetNickname() // Метод установки клички животного

{

cout << "Input nickname of pet:" << endl;

cin >> Nickname;

}

public: ~HomeAnimal() // Деструктор

{

cout << "Oh dear... pet" << endl;

}

};

int main(void)

{

setlocale(LC\_ALL, "Russian");

HomeAnimal Cat;

cout << "Сначала попробуем вывести пустые поля животного:" << endl;

Cat.GetClass();

Cat.GetCountLimbs();

Cat.GetCountDescendants();

cout << "Теперь заполним данные о животном:" << endl;

Cat.SetClass();

Cat.SetCountLimbs();

Cat.SetCountDescendants();

cout << "Теперь выведем заполненные данные о животном:" << endl;

Cat.GetClass();

Cat.GetCountLimbs();

Cat.GetCountDescendants();

cout << "Теперь заполним поля для дочерного класса" << endl;

Cat.SetNickname();

Cat.GetNickname();

cout << "Как всё живое, животное умрёт... (деструктор)" << endl;

return 0;

}