МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное бюджетное образовательное учреждение

высшего образования

«УЛЬЯНОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет информационных систем и технологий

Кафедра «Измерительно-вычислительные комплексы»

Курсовая работа

по дисциплине «Основы программирования»

Тема: «Односвязный список семейства волчьих»

Выполнил:

студент группы ИСТбд-11

Ульянин Артём Альбертович

Ульяновск

2020

УЛЬЯНОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Кафедра «Измерительно-вычислительные комплексы»

Отчет

о проверке курсовой работы по дисциплине «Основы программирования»

Студент: Ульянин Артём Альбертович группы ИСТбд-11

Тема: Связный список семейства волчьих

|  |  |  |
| --- | --- | --- |
| Проверяемый материал | Отметка о выполнении | Примечание |
| В пояснительной записке присутствуют (проверка на комплектность):   1. Титульный лист 2. Задание на курсовую работу 3. Руководство программисту 4. Руководство пользователю 5. Физический носитель информации (например, CD диск), с которого файлы могут быть скопированы (то есть, нет дефектов на диске), носитель информации не содержит вирусов |  | Обязательный минимум |
| Реализация и работоспособность программы для пользователя:   1. Загрузка данных из файла в список 2. Отображение полей данных на экране 3. Добавление элемента в список 4. Поиск элемента в списке 5. Удаление элемента из списка 6. Сохранение списка в файл |  | Обязательный минимум |
| Реализация и работоспособность программы для программиста:   1. Реализация дерева классов (7 классов) 2. Присутствует и дано объяснение использования «инкапсуляции» 3. Присутствует и дано объяснение использования «наследования» 4. Присутствует и дано объяснение использования «полиморфизма» 5. Присутствует и дано объяснение «работы с библиотечным контейнером типа список» 6. Присутствует и дано объяснение использования «перегрузки оператора» 7. Присутствует и дано объяснение использования «шаблонов» |  | Обязательный минимум |
| Реализация класса «нижнего уровня» или  реализация класса «среднего уровня» |  | Выполнение на оценку 4 или 5 |

Работа защищена с оценкой Дата

Кафедра «Измерительно-вычислительные комплексы»

Задание на курсовую работу по дисциплине «Основы программирования»

студент\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ группы \_\_\_\_\_\_\_

Тема: Связный список \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Технические условия:

1. Реализация дерева объектов выбранного типа (не менее 7 классов):

а) один базовый,

б) два подкласса – типа «сын» базового класса,

в) у каждого «сына» два своих «сына», то есть четыре «внука» базового класса,

г) базовый класс и классы «сыновья» – абстрактные,

д) создаются объекты, относящиеся только к классам «внукам»,

е) каждый из семи классов имеет не менее одной уникальной характеристики (личного поля данных).

2. Обязательные методы объекта: отображение полей данных, заполнение полей данных, загрузка данных из файла, сохранение данных в файл.

3. Реализация каждого класса в отдельном модуле, состоящем из заголовочного файла (.h) и файла реализации функций класса (.cpp).

4. Реализация списка объектов (библиотечный класс list), методы работы со списком: загрузка списка из файла, отображение полей данных элементов списка на экране, добавление элемента в список, поиск элемента в списке, удаление элемента из списка, сохранение списка в файл. Отдельный модуль.

5. Реализация выбора перечисленных методов работы со списком (меню).

6. Обязательно использовать: основные свойства объектов (инкапсуляция, наследование, полиморфизм), перегрузку оператора (например, для вывода информации об объекте в стандартный поток вывода).

7. Язык программирования – С++.

Содержание пояснительной записки:

*1. Титульный лист*

*2. Задание на курсовую работу*

*3. Отчет о проверке*

*4. Руководство пользователю*

*5. Руководство программисту*

Дата выдачи задания 15.03.2020 г. Срок выполнения 15.05.2020 г.

Зав. кафедрой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ С.К. Киселев

(подпись) (Ф.И.О.)

Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А.И. Армер

(подпись) (Ф.И.О.)

Работа защищена с оценкой Дата

Оглавление

1. Введение . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .. . . . . . . . 5

2. Руководство программиста . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 6

3. Руководство пользователя . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 10

4. Заключение . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .. . . . . . . . . . . . . . . . . . . . . . . 13

**Введение**

**Цель курсовой работы:**

Разработать список, удовлетворяющий техническим требованиям. Спроектировать дерево (иерархию) простых классов, реализовать необходимый функционал, включающий в себя функции сохранения и чтения из файла, вывод списка на экран, добавление, поиск и удаление элемента, использовать понятия «Инкапсуляция», «Наследование», «Полиморфизм», «Список» «Перегрузка операторов» и «Шаблоны».

Инкапсуляция – механизм, связывающий код и данные, которыми манипулирует код и защищает от некорректного использования;

Наследование – процесс, с помощью которого один объект приобретает свойства другого, т.е. поддерживается иерархическая, классификация;

Полиморфизм – свойство, позволяющее использовать один и тот же интерфейс для общего класса действий;

Список – это динамическая структура данных, элементы которой содержат ссылку на следующий элемент;

Перегрузка операторов – позволяет определить действия, которые будет выполнять оператор;

Шаблоны – средство языка C++ , предназначенное для кодирования обобщённых алгоритмов, без привязки к некоторым параметрам;

**Руководство программисту**

Исходный код программы состоит из 20 файлов, из которых 10 заголовочных файлов с расширением «.h» и 10 файлов с расширением «.cpp»(реализация методов классов и функционала программы). В таблице 1 приведен список файлов с комментариями о назначении того или иного файла.

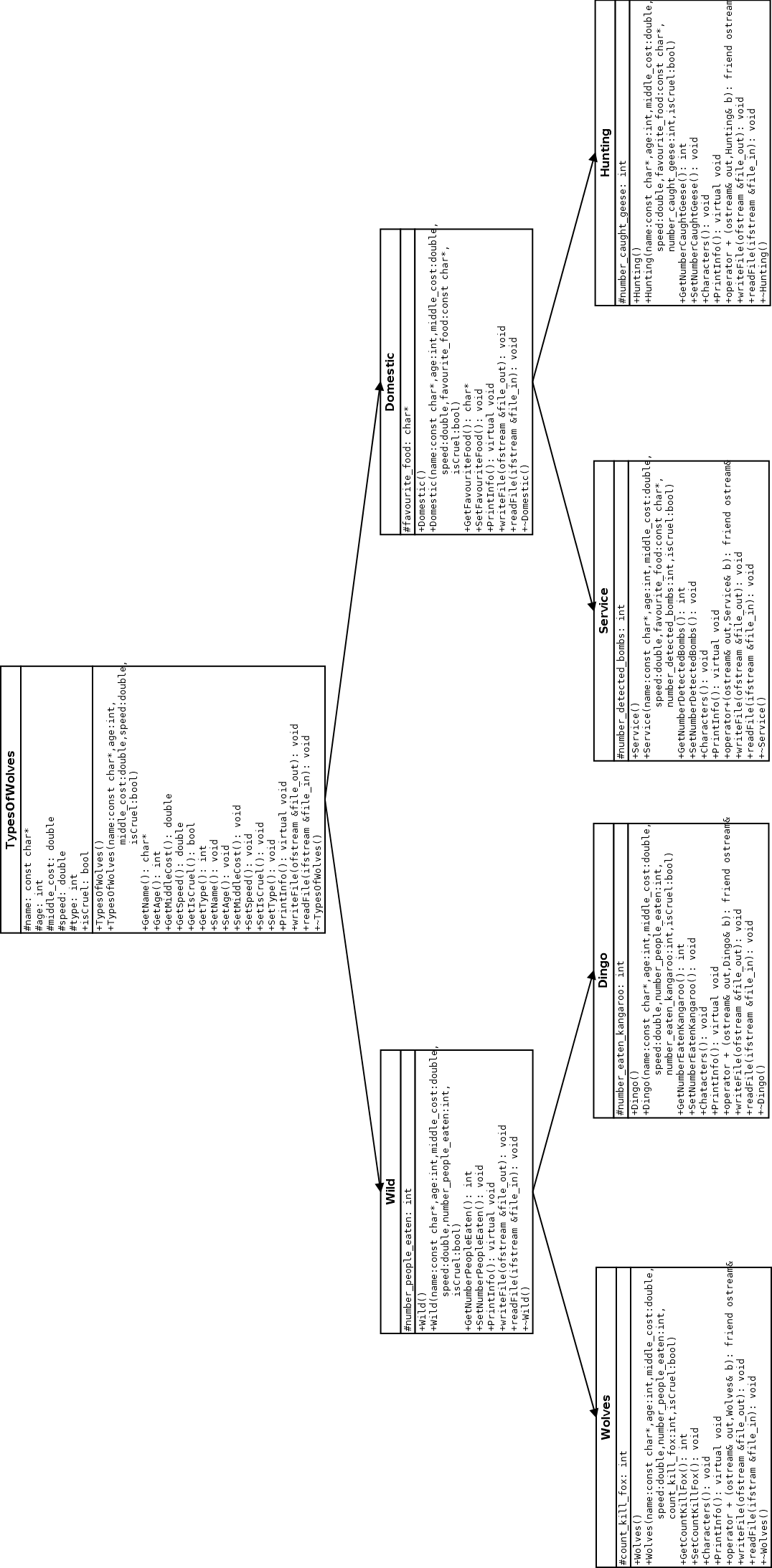
**Таблица 1.**

|  |  |  |
| --- | --- | --- |
| **«.h» файл** | **«.cpp» файл** | **Комментарий** |
| TypesOfWolves.h | TypesOfWolves.cpp | Заголовочный файл и файл реализации класса TypesOfWolves |
| Wild.h | Wild.cpp | Заголовочный файл и файл реализации класса Wild |
| Domestic.h | Domestic.cpp | Заголовочный файл и файл реализации класса Domestic |
| Wolves.h | Wolves.cpp | Заголовочный файл и файл реализации класса Wolves |
| Dingo.h | Dingo.cpp | Заголовочный файл и файл реализации класса Dingo |
| Service.h | Service.cpp | Заголовочный файл и файл реализации класса Service |
| Hunting.h | Hunting.cpp | Заголовочный файл и файл реализации класса Hunting |
| List.h |  | Заголовочный файл и он же файл реализации класса List |
| Interface.h | Interface.cpp | Заголовочный файл и файл реализации меню программы |
| Node.h | Node.cpp | Заголовочный файл и файл реализации класса Node |
|  | Main.cpp | Файл содержащий функцию main() |

## Иерархия классов

При выполнении курсовой работы, согласно техническому заданию, была разработана иерархия классов в виде «дерева», реализующая единственное наследование.

Наследование в классах осуществляется с помощью спецификатора :**public**.



Класс TypesOfWolves является базовым классом. Данный класс имеет 2 подкласса – Wild и Domestic («сыновья» базового класса). Объекты этих классов не создаются и не используются, т.к. данные классы являются абстрактными. Классы Wild и Domestic имеют, в свою очередь, по 2 «сына» Wolves, Dingo и Service, Hunting соответственно, которые являются «внуками» базового класса TypesOfWolves. Пользователю в программе доступно создание и операции только с объектами типа: Wolves, Dingo, Service, Hunting

Поля данных, всех вышеперечисленных классов, размещены в секции protected: описания классов для обеспечения инкапсуляции данных. Т.е. доступ к полям данных классов возможен только из самого класса и классов-наследников. Доступ из вне возможен только с помощью функций-членов (интерфейса) класса, чем и обеспечивается принцип инкапсуляции.

Для обеспечения функциональности работы с объектами в каждом классе были созданы методы, обеспечивающие:

* заполнение полей данных

Например, в классе TypesOfWolves:

*void SetName();*

*void SetAge();*

*void SetMiddleCost();*

*void SetSpeed();*

*void SetIsCruel();*

*void SetType();*

* получение значений полей данных (все методы, имеющие приставку get\_).

Например, в классе TypesOfWolves:

*char\* GetName();  
int GetAge();*

*double GetMiddleCost();*

*double GetSpeed();*

*bool GetIsCruel();*

*int GetType();*

* отображение полей данных.

Например, в классе TypesOfWolves:

*virtual void PrintInfo() const;*

Функция ​ virtual void PrintInfo()​ ​ объявлена при помощи ключевого слова virtual, что означает, что она может быть определена где-то в других классах. В этом заключается принцип полиморфизма (в зависимости от вызываемого объекта функция будет иметь разную реализацию). Принципы построения других классов аналогичны.

* загрузка данных из файла.

Например, в классе TypesOfWolves:

*void readFile (std::ifstream &file\_in);*

* сохранение данных в файл.

Например, в классе TypesOfWolves:

*void writeFile (std::ofstream &file\_out);*

Также в данной работе также была реализована такая возможность языка как перегрузка оператора, например перегрузка оператора вывода на стандартное устройство.

*friend std::ostream& operator+(std::ostream& out, Dingo& b)*

# Список

Согласно техническому заданию, для реализации списка объектов было предложено использовать класс list (список). В данной курсовой работе реализован контейнер – односвязный список. Структура односвязного списка представлена на Рис.2.

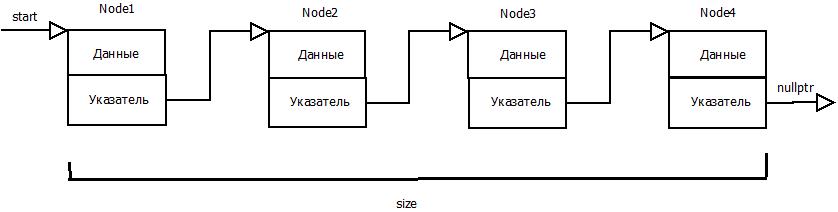


Рис.2 Структура односвязного списка.

Структура списка тривиальна и не требует особых пояснений. Необходимо отметить, что в данной реализации Список List (и узлы списка Node) оформлены в виде шаблонных классов, что делает код более универсальным (позволяет использовать любые типы данных).

Так же необходимо отметить, что в качестве данных, в узлах списка, мы будем хранить не объекты, а указатели на них: каждый элемент списка будет иметь указатель на следующий элемент. Данное решение позволит хранить в одном списке указатели на объекты производных классов (Wolves, Dingo, Service, Hunting), т.к. они имеют общий базовый тип – TypesOfWolves.

В классе List реализованы следующие методы работы со списком:

* загрузка списка из файла: *void download()*
* сохранение списка в файл: *void Output()*
* добавление элемента в список:  *void Add(T \*element)*
* удаление элемента из списка: *void deleteElement(int index)*
* поиск элемента в списке: *T findById(int index)*
* отображение полей данных элементов списка на экране: *void printAll()*
* отображение количества элементов в списке *void printSize()*
* удаление элемента из начала списка: *void popFront()*

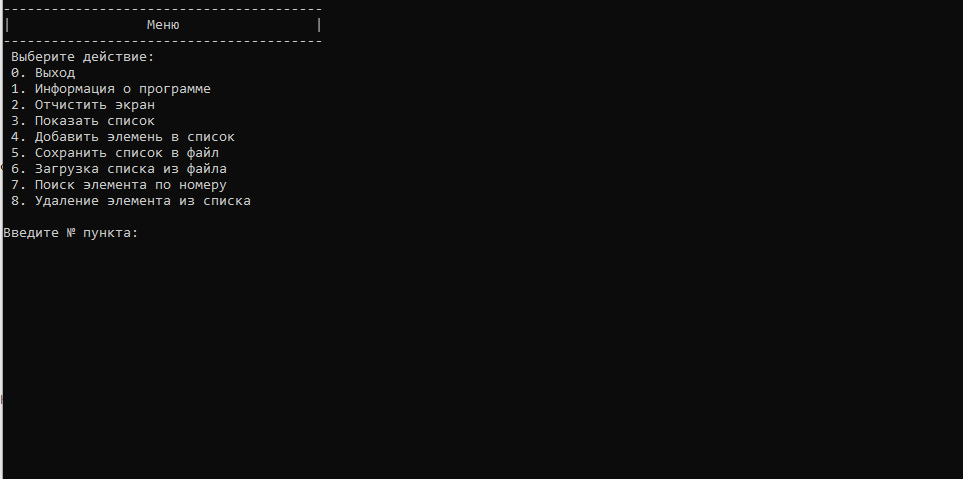
# Руководство пользователя

# Инструкция по использованию программы:

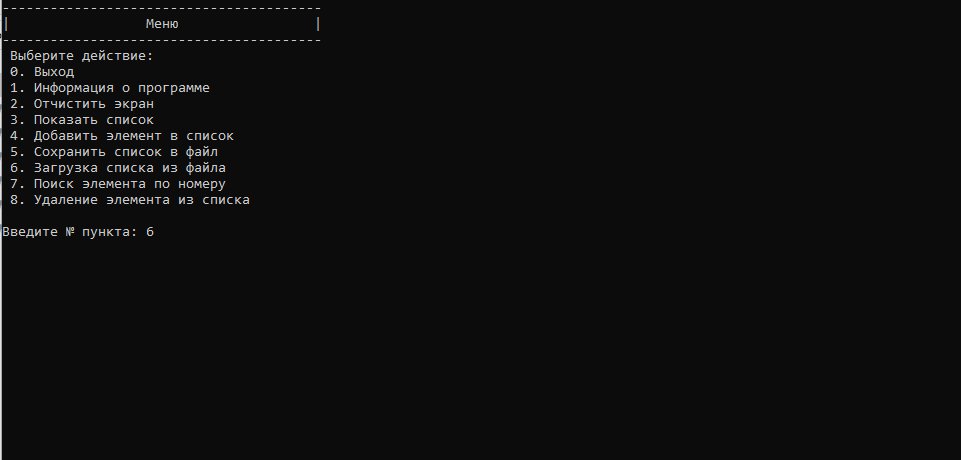
1. Скопировать папку «Для пользователя» на компьютер.

2. Запустить .exe файл программы, либо открыть проект, откомпилировать его и запустить

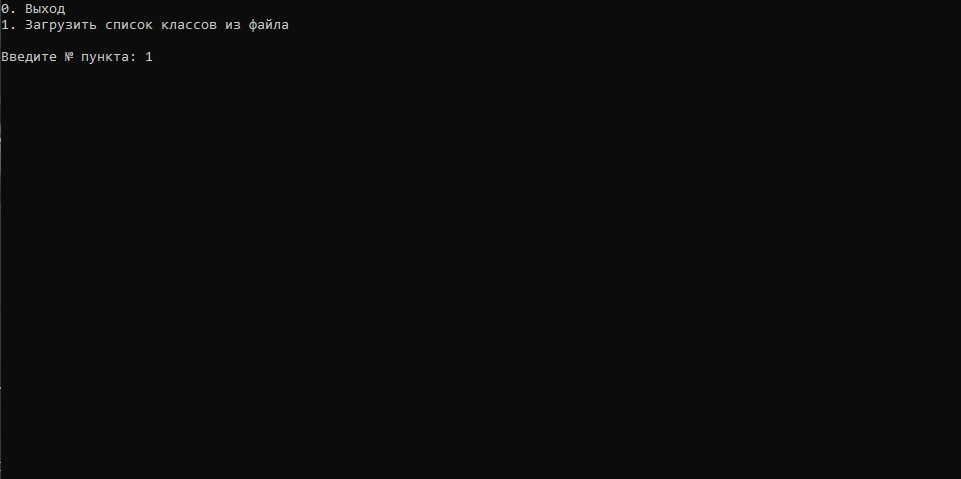
3. Перед вами появится главное меню программы и будет предложено выбрать один из 9 предложенных пунктов программы. (Рис.1)



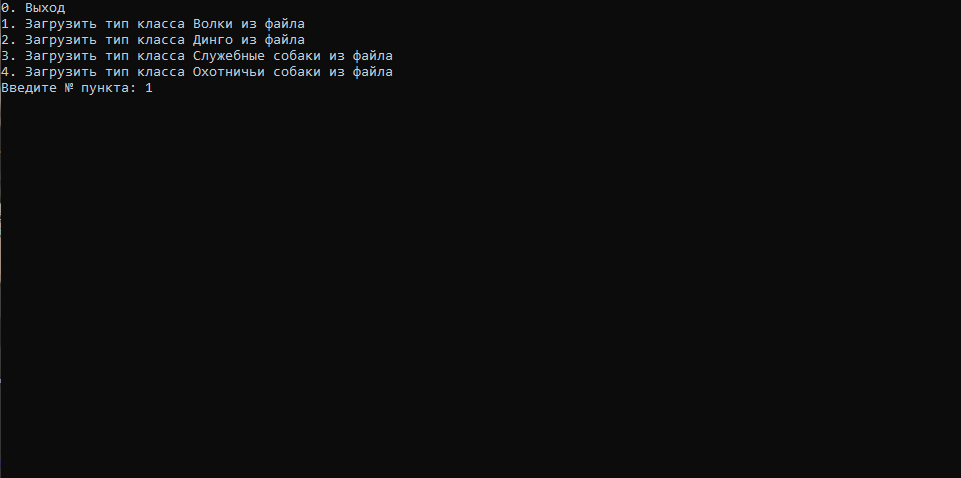
4. Для начала работы, например, загрузим список из файла. Для этого необходимо выбрать пункт № 6.



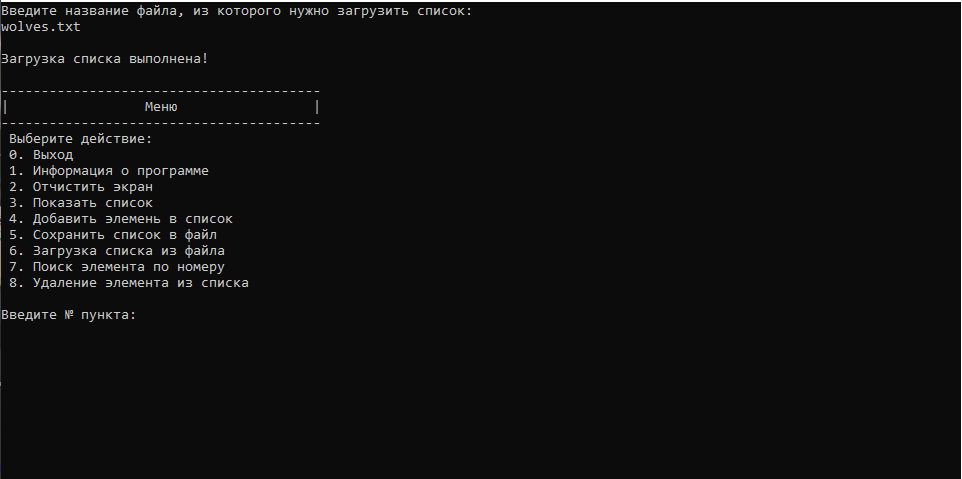
5. Далее выбираем пункт № 1, чтобы начать действия загрузки из файла.   
Если вы передумали загружать, то можно нажать пункт № 0.



6. Далее необходимо выбрать тип класс, который вы хотите загрузить из файла. К примеру, класс Волки. Выбираем пункт № 1.



7. Далее необходимо будет ввести имя файла с расширением . Если файл с данным именем существует, то выведется сообщение: «Загрузка списка выполнена !». После этого снова появится меню.



Аналогичным образом работают и другие пункты меню. Для того, чтобы просмотреть все объекты списков, пользователь должен в основном меню выбрать раздел «Показать список», а то есть ввести № пункта - 3.   Что касается поиска объекта, то здесь для начала пользователь вводит класс нужного элемента, затем индекс нужного элемента (начиная с 0) и программа выводит его на экран.   
 Также пользователь может удалять любой объект из списка по его индексу, при успешной операции пользователь получит об этом уведомление, иначе увидит ошибку, что такого элемента нет.

# Заключение

# В данной курсовой работе был реализован односвязный список , соответствующий техническим требованиям. Было разработано: сохранения и чтения из файла, вывода списка на экран, добавления, поиска и удаления элемента. В ходе процесса написания курсовой работы были изучены следующие темы языка C++, так как вся работа была выполнена на этом языке программирования: «Потоки ввода и вывода», «Перегрузка функций», «Инкапсуляция», «Наследование», «Полиморфизм», «Перегрузка операторов» и «Шаблоны». Таким образом, я считаю, что цель курсовой работы достигнута.

# 

# 

# 