**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**«РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ»**

Факультет физико-математических и естественных наук

Кафедра информационных технологий

«УТВЕРЖДАЮ»  
Заведующий кафедрой  
информационных технологий  
д.ф.-м.н., проф.

\_\_\_\_\_\_\_\_\_\_\_\_Ю.Н. Орлов   
« » 2023г.

**ОТЧЕТ  
по лабораторной работе №1**

по дисциплине «Технология программирования»

Выполнил:

Студент группы НПИбд-02-22

Студенческий билет № 1132226528

Тарутина Кристина Олеговна

(Подпись)

«22» Февраля 2023г.

Москва 2023

**Цель**

Написать компьютерную программу, содержащую

 - Описание структуры, содержащей поля типа string, int, double;

 - Набор функций для работы со списком на базе этой структуры:

     - Добавление элемента в начало списка;

     - Добавление элемента в конец списка;

     - Добавление элемента в список после заданного элемента;

     - Добавление элемента в список перед заданным элементом;

     - Удаление из списка элемента с заданным именем;

     - Вывод содержания списка на экран;

 - Функцию main, содержащую сценарий работы со списком, использующий разработанный инструментарий.

**Ход лабораторной работы**

**Шаг 1: Описание структуры**

Сначала я описываю структуру для списка студентов РУДН, где будут храниться имя, фамилия, номер студенческого билета, количество баллов, а также оценки в русском и английском формате(рис 1)

Изображение выглядит как текст

Автоматически созданное описание

Рис 1

**Шаг 2: Создание функций**

После этого я создаю функцию для добавления элемента в начало списка. Сперва я объявляю указатель для описанной мной ранее структуры, и заполняю поля из структуры значениями, которые передаю при вызове функции. В поле next я вкладываю значение головы списка, после чего заменяю значение головы на нынешний элемент.(рис 2)

Изображение выглядит как текст

Автоматически созданное описание

Рис 2

Следующей я создаю функцию для добавления элемента в конец списка. Здесь я объявляю указатель временной переменной, внутрь которого вкладываю значение "головы" списка. Сразу проверяю не равна ли голова списка нулю, ибо тогда можно просто вызвать функцию добавления элемента в начало списка и не утруждаться далее.

Если же голова списка не равна нулю, то в цикле перемещаемся по списку с помощью поля next, после чего повторяем действия из прошлого цикла, также создавая указатель для описанной структуры и заполняя поля, за одним небольшим исключением. Теперь в поле next у нас будет храниться значение NULL, так как после на данный момент следующей структуры нет.

Изображение выглядит как текст

Автоматически созданное описание

Рис 3

Следующим реализуем функцию вставки элемента после заданного. Для этого помимо привычного набора данных, мы передаём в функцию также номер студенческого билета студента, после которого в список мы хотим добавить нового студента.

Сразу проверяем, чтобы список не был пустым, сверяясь не лежит ли в значении головы списка NULL и объявляем указатель, после чего в цикле, до тех пор, пока мы не дойдём до конца или пока не найдём нужного студента по номеру, мы переходим на следующий элемент списка. Снова проверяем не дошли ли мы до конца списка, и если нет, то добавляем в список нового студента. (рис 4)

Изображение выглядит как текст

Автоматически созданное описание

Рис 4

Теперь приступаем к функции вставки до указанного элемента. Во многом эта функция похожа не предыдущую, за небольшим исключением. Теперь мы также в начале проверяем не только, что список пустой, но и не является ли нужный нам студент началом данного списка. Если это действительно так, то мы просто пользуемся функцией вставки элемента в начало списка. После этого мы объявляем два указателя, один на голову списка, другой на последующий за головой элемент. Цикл почти такой же, как в предыдущей программе, за исключением того, что передвигаем мы не один, а два указателя. (рис 5)

Изображение выглядит как текст

Автоматически созданное описание

Рис 5

Для функции по удалению предмета нам нужно лишь знать голову списка и номер студенческого билета того, кого мы хотим удалить.

Сразу же проверяем, что наш список не пустой, после чего объявляем указатель и, если нужный студент в начале списка, удаляем его, переменной Head назначая следующий элемент.

Если студент не в начале списка, то объявляем ещё один указатель. Он будет содержать голову списка, в то время как предыдущий указатель последующий элемент.

Снова реализуем цикл поиска студента по номеру его студенческого билета (рис 6)

Изображение выглядит как текст

Автоматически созданное описание

Рис 6

Последней создаём функцию вывода списка на экран. Проверяем, что список не пуст, и если нет, то уже использованным до этого методом проходимся по всем элементам списка до самого конца.(рис 7)

Изображение выглядит как текст

Автоматически созданное описание

Рис 7

**Шаг 3: Функция main**

C помощью условных операторов, цикла while, а также операций ввода/вывода, я реализую возможность пользователю самому создавать, изменять и выводить список студентов, а также удалять конкретных лиц по желанию(рис 8-10)

Изображение выглядит как текст

Автоматически созданное описание

Рис 8

Изображение выглядит как текст

Автоматически созданное описание

Рис 9



Рис 10

Вывод

Выполнение лабораторной работы 1 прошло успешно, все цели выполнены, полностью рабочая программа и отчёт по ней выложены на гитхаб.