Частное учреждение образования

«Колледж бизнеса и права»

|  |  |
| --- | --- |
|  | УТВЕРЖДАЮ  Ведущий методист колледжа  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Е.В. Паскал  « \_\_\_\_ » \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2021 года |
| Специальность 2-40 01 01 «Программное обеспечение информационных технологий» | Учебная дисциплина «Основы алгоритмизации и программирование» |

**Лабораторная работа № 30**

**Инструкционно-технологическая карта**

Тема: Разработка алгоритмов и программ с использованием списковых структур данных.

Цель: Научиться разрабатывать алгоритмы и программы с использованием списковых структур данных.

Время выполнения: 2 часа

1. **Порядок выполнения работы**
2. Изучить теоретические сведения к лабораторной работе.
3. Разработать на языке С++ программу вывода на экран решения задачи в соответствии с вариантом индивидуального задания, указанным преподавателем.
4. Отлаженную, работающую программу сдать преподавателю. Работу программы показать с помощью самостоятельно разработанных тестов.
5. Ответить на контрольные вопросы.
6. **Теоретические сведения**

**Динамические структуры данных**

Структура – это составной тип данных, в котором под одним именем объединены данные различных типов. Отдельные данные структуры называются полями.

Списковые структуры строят из специальных элементов, включающих помимо информационной части (данных, которые собственно храним) еще и один или несколько указателей-адресов элементов или вложенных структур, на которые ссылаемся из этой структуры, а значит, к которым можем получить доступ из этой структуры (из этого элемента). Объявление структуры осуществляется с помощью ключевого слова *struct*, за которым идет ее имя и в фигурных скобках список элементов-полей и указателей, заключенных в фигурные скобки (и оператор «точка с запятой» в конце):

*struct имя*

{

*тип\_элемента\_1 имя\_элемента\_1*;

*тип\_элемента\_2 имя\_элемента\_2*;

...

*тип\_элемента\_n имя\_элемента\_n*;

*имя\_struct\* указатель\_на\_следующий\_элемент*;

*имя\_struct\* указатель\_на\_предыдущий\_элемент*;

};

Далее объявляете свою списковую структуру:

*имя\_структуры\* MyList = new имя\_структуры( );//с помощью указателя\* и оператора* ***new*** *выделяем динамическую память для одного элемента структуры, который создаем с помощью конструктора без параметров* ***имя\_структуры( )***

**Связный список** – одна из базовых структур данных. Ее часто сравнивают с массивом, так как многие другие структуры можно реализовать с помощью либо массива, либо связного списка. У этих двух типов есть преимущества и недостатки. Связный **однонаправленный список** состоит из группы узлов, которые вместе образуют последовательность. Каждый узел содержит две вещи: фактические данные, которые в нем хранятся (это могут быть данные любого типа) и указатель (или ссылку) на следующий узел в последовательности. Также существуют **двусвязные списки**: в них у каждого узла есть указатель и на следующий, и на предыдущий ему элемент в списке. Основные операции в связном списке включают добавление, удаление и поиск элемента в списке.

**Стек** – это базовая структура данных, которая позволяет добавлять или удалять элементы только в её начале. Она похожа на стопку книг: если вы хотите взглянуть на книгу в середине стека, сперва придется убрать лежащие сверху. Или сдаваемые учениками начальной школы тетради на проверку: если каждый ученик кладет свою тетрадь в стопку сверху, то преподаватель начнет проверку с самой последней сданной тетради, ведь она в стопке-стеке лежит сверху. Стек организован по принципу LIFO (Last Input – First Output, «последним пришёл – первым вышел»). Это значит, что самый последний элемент, который вы добавили в стек, первым выйдет из него. В стеках можно выполнять три операции: добавление элемента, удаление элемента и отображение (чтение) содержимого стека.

**Очередь** – эту структуру можно представить, как очередь в продуктовом магазине. Первым обслуживают того, кто пришёл в самом начале – всё как в жизни. Очередь устроена по принципу FIFO (First Input – First Output, «первым пришёл – первым уйдёт»). Это значит, что удалить очередной элемент можно только после того, как были убраны все ранее добавленные перед ним элементы. Очередь позволяет выполнять две основные операции: добавлять элементы в конец очереди и удалять первый элемент из очереди.

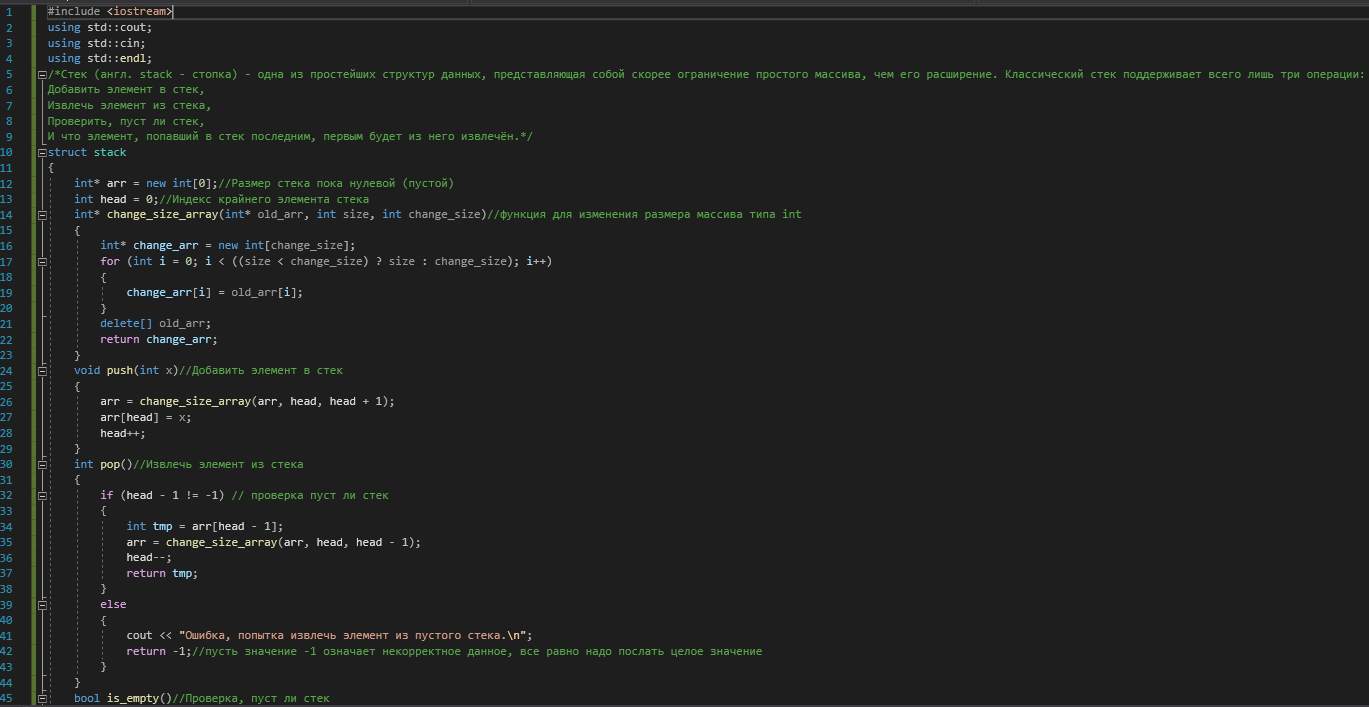
**Кольцевой список** (кольцо) – список, в котором последний элемент хранит указатель не на NULL, а на первый элемент этого же списка, в результате чего список становится «закольцованным» (циклическим, круговым) и по нему можно бесконечно ходить, поскольку у него конец переходит в начало. Круговой список может быть как однонаправленным (описан выше), так и двунаправленным, у которого в каждом элементе есть указатель на предыдущий и последующий элементы. Соответственно, в кольцевом двунаправленном списке первый элемент указывает на последний, а последний элемент указывает на первый, что позволяет ходить по кольцевому двунаправленному списку и слева направо, и справа налево.

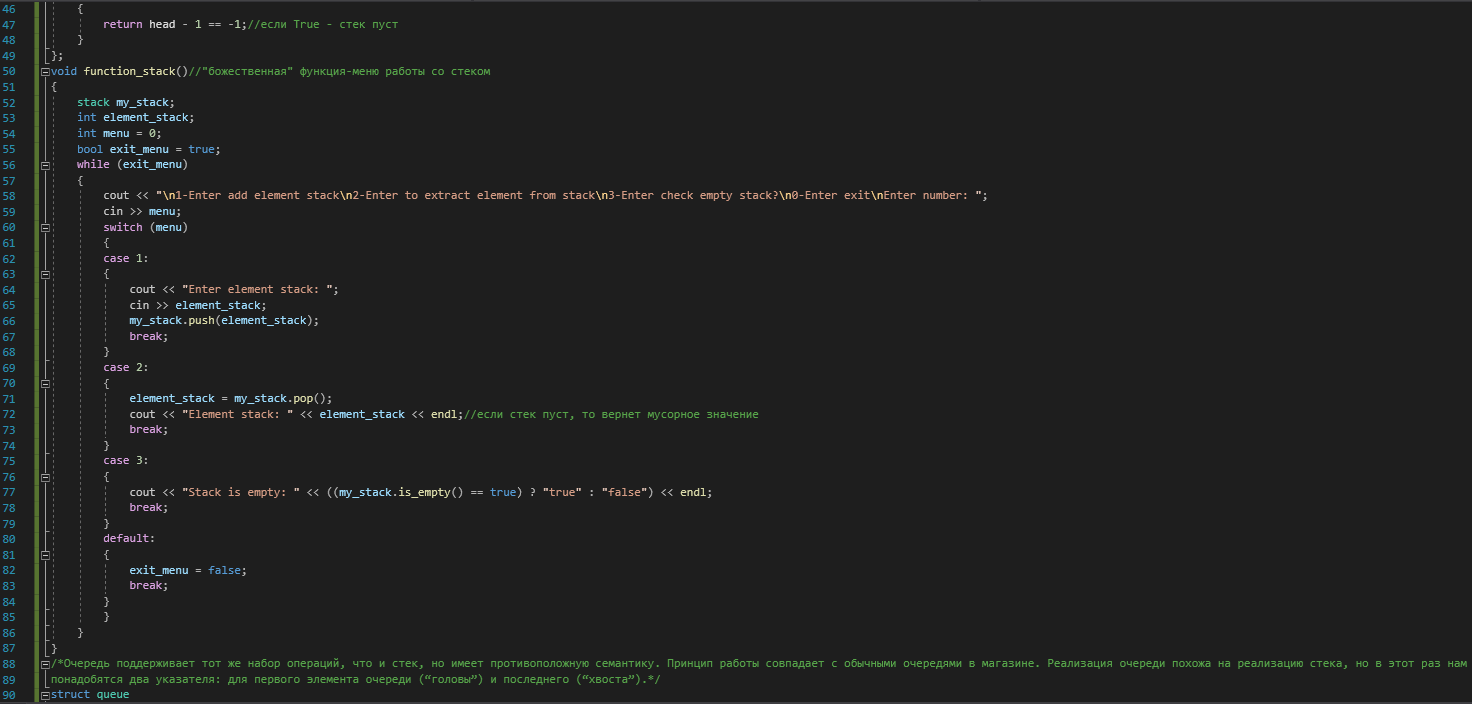
**Дек** (дека, двусторонняя очередь, DEque – Double Ended queue) – очередь с двумя концами и двумя началами одновременно; двусвязный список, в котором элементы можно добавлять и удалять с обеих сторон (и из начала, и из конца этого двунаправленного списка.

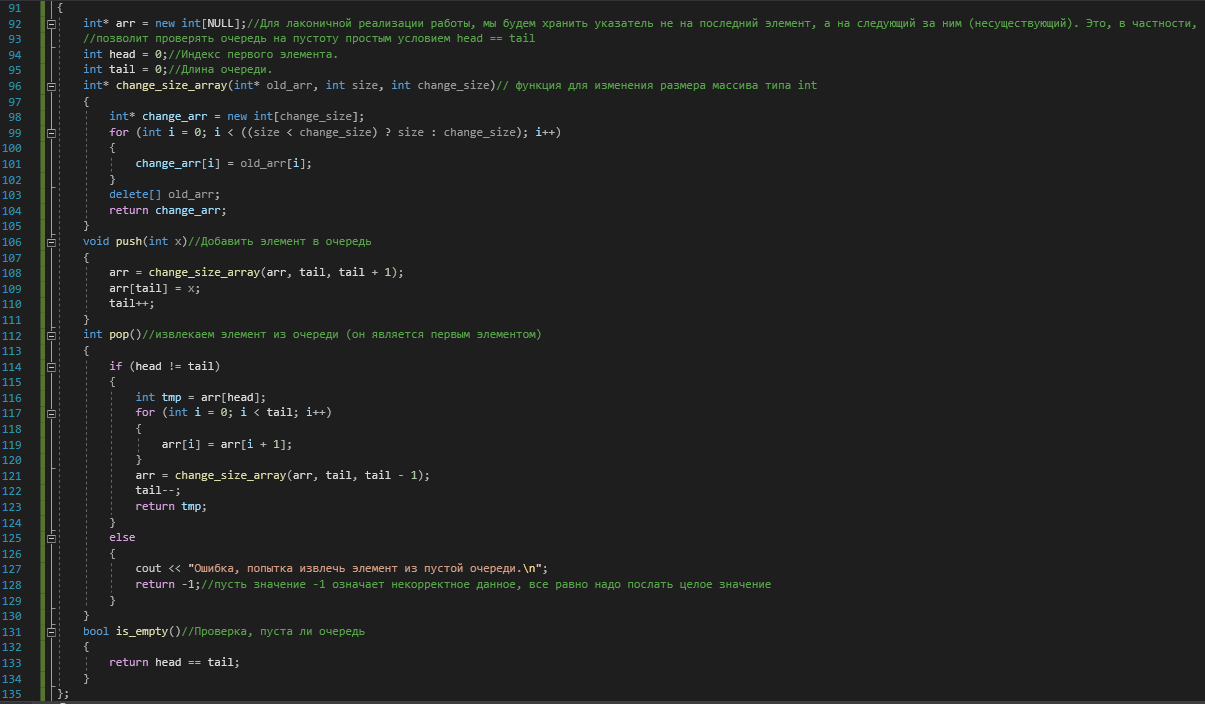
**Словарь** (dictionary, map) – это динамическая структура данных, в которой все элементы состоят из пары «ключ – хранимое значение», и хранятся в отсортированном по ключу порядке. Для получения значения, надо знать его ключ. Словарь map аналогичен ранее изученному множеству set (multiset), тоже автоматически сортируется (по ключу), но хранит не просто значения, а пары «ключ – значение». Значения должны иметь уникальные ключи и поиск значений осуществляется по ключу. Для использования словаря map необходимо подключить библиотеку #include <map>.

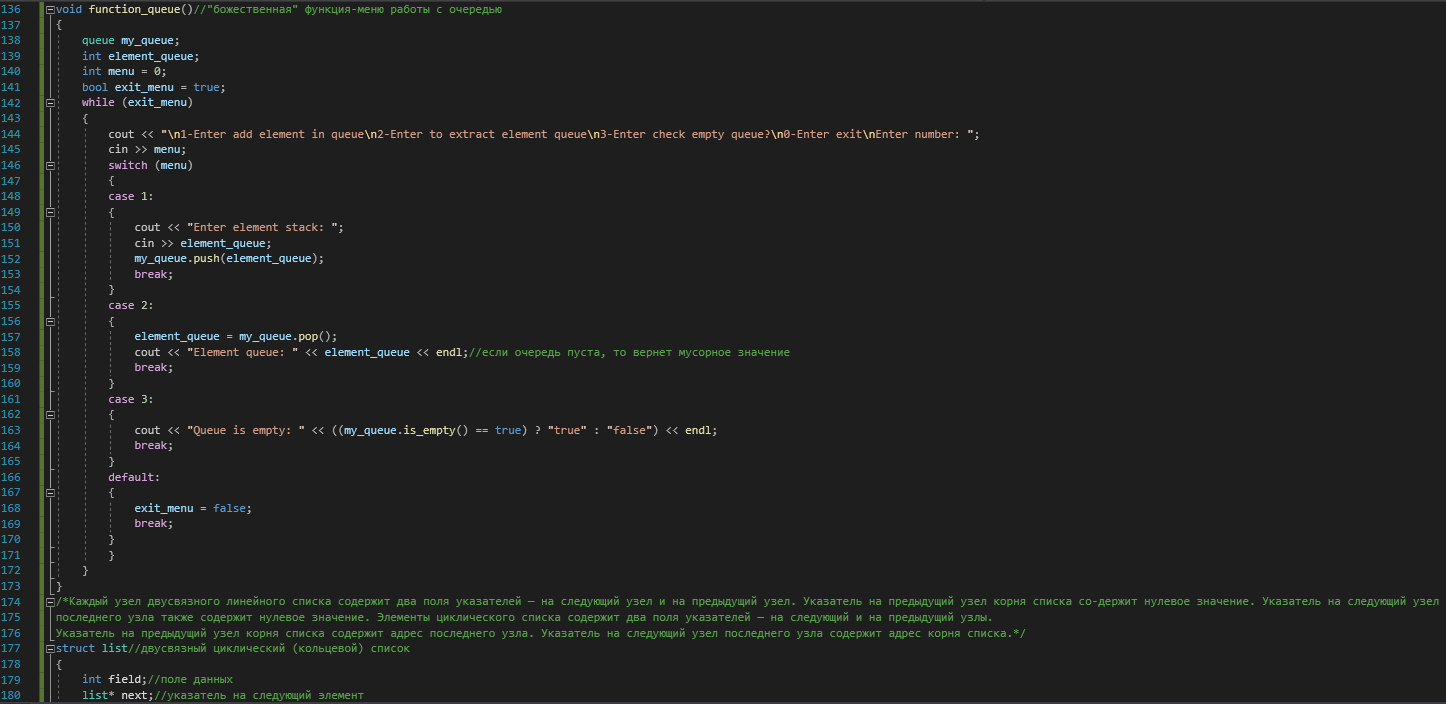
1. **Пример выполнения программы**

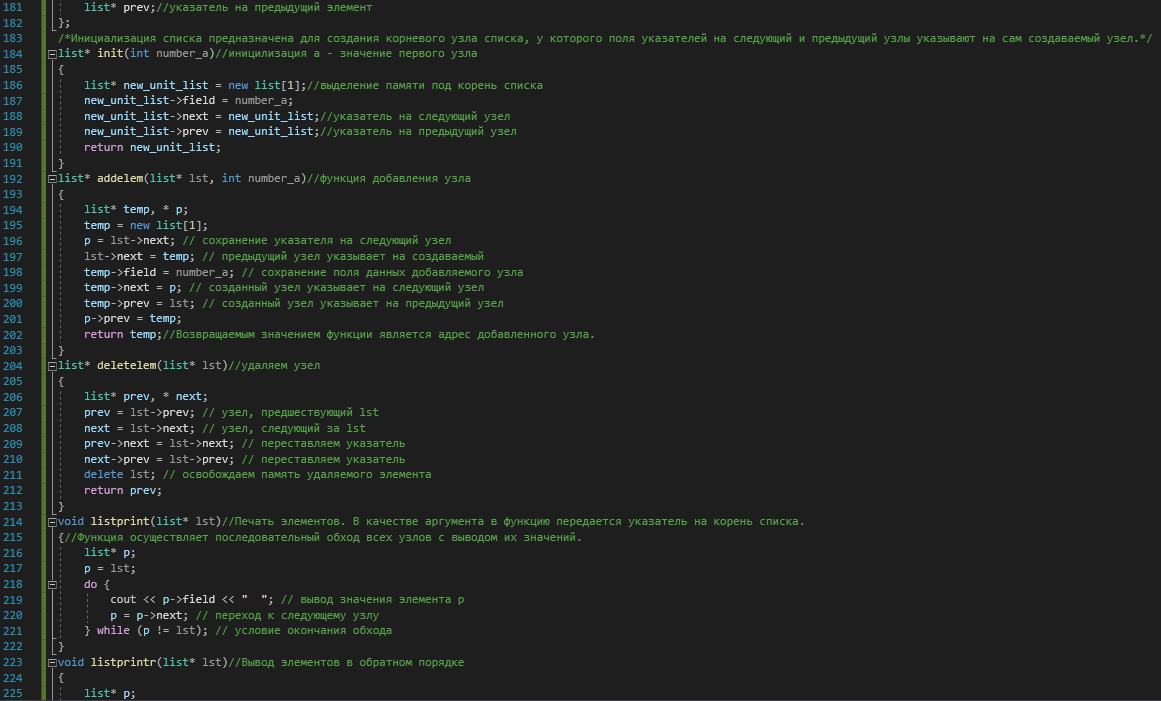
Рассмотрим один из способов реализации динамического стека, очереди и списка для одного целочисленного поля int (информационная часть).

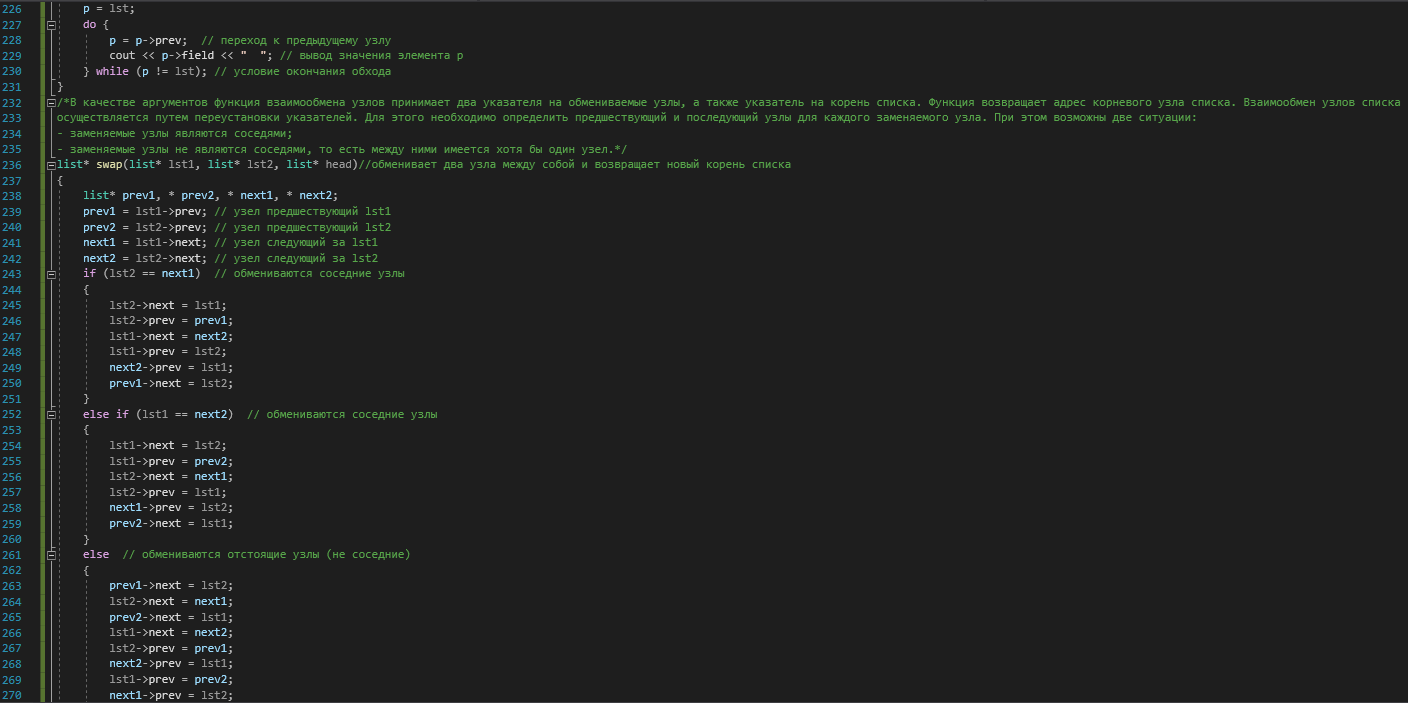


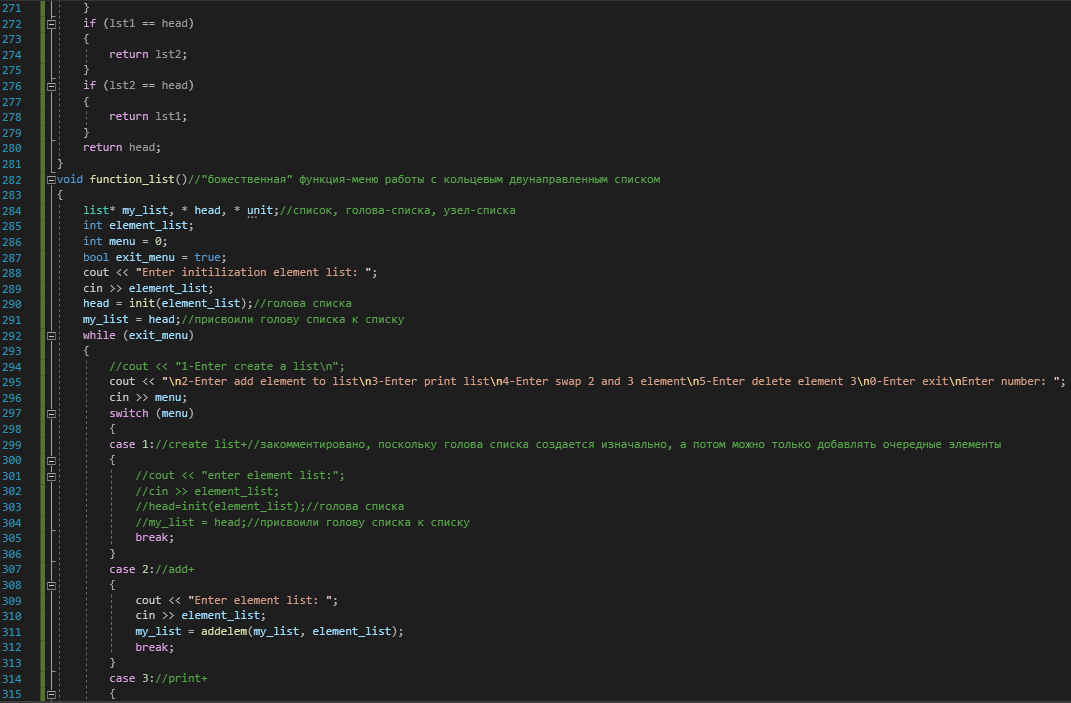


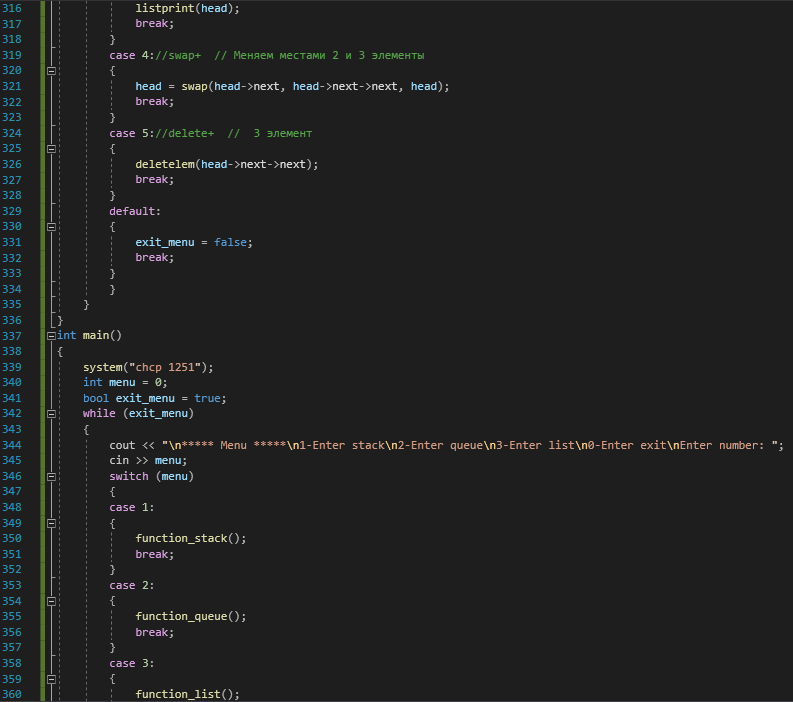


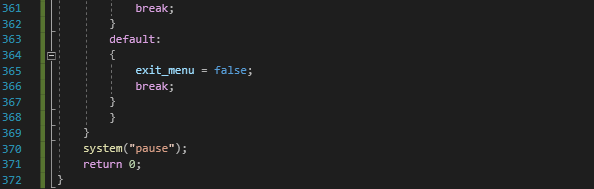












1. **Задания по вариантам**

Для вариантов 1, 5, 9, 13, 17, 21, 25 реализовать двусвязный линейный список.

Для вариантов 2, 6, 10, 14, 18, 22, 26 реализовать односвязный циклический (кольцевой) список.

Для вариантов 3, 7, 11, 15, 19, 23, 27 реализовать очередь.

Для вариантов 4, 8, 12, 16, 20, 24, 28 реализовать стек.

В вашей программе должны быть реализованные функции:

1. Добавление элемента в структуру данных.
2. Изменение элемента в структуре данных (по выбранному параметру).
3. Удаление элемента в структуре данных (по выбранному параметру).
4. Вывод одного элемента структуры данных на экран.
5. Вывод всей структуры данных на экран (всех элементов).
6. Подсчет количества всех элементов в структуре данных.

Варианты индивидуальных заданий:

1*.* Сформировать структуру данных, содержащую сведения о количестве изделий, собранных сборщиками цеха за неделю. Структурный список содержит поля: фамилия сборщика, количество изделий, собранных им ежедневно в течение шестидневной недели, т.е. раздельно в понедельник, вторник и т.д. Написать программу, выдающую на печать: фамилию сборщика и общее количество деталей, собранных им за неделю.

2. Сформировать структуру данных, содержащую информацию о покупателе: ФИО, домашний адрес покупателя, суммы покупок, размер предоставляемой скидки. Вывести информацию о покупателе и сумме покупок с учетом скидки.

3. Сформировать структуру данных, содержащую информацию о складе, включает в себя наименование товара, количество единиц товара, цену единицы и дату поступления товара на склад. Вывести информацию о товаре и его стоимость.

4. Сформировать структуру данных, содержащую информацию о рейсе указаны его номер, тип автобуса, пункт назначения, время отправления и прибытия. Вывести информацию о рейсе, и разницу между прибытием в пункт назначения и заданном времени.

5. Сформировать структуру данных, содержащую информацию междугородной АТС, которая содержит дату разговора, код и название города, время разговора, тариф, номер телефона в этом городе и номер телефона абонента. Вывести город, общее время разговоров с ним и сумму.

6. Сформировать структуру данных. Информация об участнике спортивных соревнований содержит наименование страны, название команды, ФИО игрока, игровой номер, возраст, рост и вес. Вывести информацию о ИМТ игрока.

7. Сформировать структуру данных, содержащую информацию о сотруднике фирмы: ФИО, табельный номер, количество отработанных часов за месяц, почасовой тариф. Рабочее время свыше 144 часов считается сверхурочным и оплачивается в двойном размере. Вывести размер заработной платы сотрудника фирмы за вычетом подоходного налога, который составляет 12 % от суммы заработка.

8. Сформировать структуру данных, содержащую информацию о книге: регистрационный номер книги, автор, название, год издания, издательство, количество страниц. Вывести возраст книги.

9. Сформировать структуру данных; информация о сотруднике предприятия содержит ФИО, номер отдела, должность, дату начала работы. Вывести стаж сотрудника.

10. Сформировать структуру данных. В справочнике аэропорта хранится расписание вылета самолетов. Для рейса указаны номер рейса, тип самолета, пункт назначения, время вылета. Вывести информацию о рейсе, и разницу между отправлением в пункт назначения и заданном времени.

11. Сформировать структуру данных. В радиоателье хранятся квитанции о сданной в ремонт радиоаппаратуре. Квитанция содержит следующую информацию: наименование группы изделий (телевизоры, радиоприемники и т. п.), марку изделия, дату приемки и время ремонта. Вывести информацию о состоянии заказа.

12. Сформировать структуру данных, содержащую информацию о горожанине: фамилия И.О., дата рождения, адрес. Посчитать возраст.

13. Сформировать структуру данных, содержащую информацию о постояльцах гостиницы: паспортные данные, даты приезда и отъезда, номер. Посчитать количество дней пребывания.

14. Сформировать структуру данных, содержащую информацию о клиенте банка: фамилия И.О., номер счета, сумма на счете, процент. Посчитать сумму после начисления процентов.

15. Сформировать структуру данных, содержащую информацию об ученике: фамилия, группа, отметки по физике, информатике, истории. Определить средний бал оценок по всем предметам.

1. **Контрольные вопросы**
2. Для моделирования каких данных целесообразно использовать массив структур?
3. Как получить доступ к элементу структуры?
4. Что за структура данных стек?
5. Что за структура данных очередь?
6. Что за структура данных список?
7. Что за структура данных кольцо (кольцевой список)?
8. Что за структура данных дека?
9. Что за структура данных словарь? Есть ли связь словаря и множества?
10. В чем разница между обычным списком и кольцом?
11. В чем разница между односвязным и двусвязным списками?

**Литература**

**Дейтел,** Х.М. Как программировать на С++ / Х.М. Дейтел, П.Дж. Дейтел . – М. : Бином-Пресс , 2018 . – 1456 с.

**Павловская**, Т. А. С++. Объектно-ориентированное программирование : практикум / Т.А. Павловская, Ю.А. Щупак . – СПб. : Питер , 2019 . – 265 с.

**Страуструп**, Б. Язык программирования С++ / Б. Страуструп . – СПб. : Бином-Пресс , 2019 . – 1054 с.

Преподаватель Шаляпин Ю.В.

|  |
| --- |
| Рассмотрено на заседании цикловой  комиссии ПОИТ № 10  Протокол №\_\_\_\_от «\_\_\_»\_\_\_\_\_\_\_\_\_\_2021 г.  Председатель ЦК \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |