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 № 31**

**Инструкционно-технологическая карта**

Тема: Разработка алгоритмов и программ с использованием структуры «бинарное дерево».

Цель: Научиться разрабатывать алгоритмы и программы с использованием структуры «бинарное дерево».

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**

**Структура бинарного дерева**

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

*Дерево* – структура данных, представляющая собой древовидную структуру в виде набора связанных узлов. Дерево является продолжением развития темы динамического списка, но у дерева каждый узел может хранить указатели на два или более других узлов, которые связаны между собой и отходят от одного корневого узла. Если графически изобразить элементы такой структуры данных и связи между ними, то это напомнит куст или дерево, что и дало название данной конструкции. Корневой узел обычно изображают сверху. Если элемент дерева может содержать максимум два указателя на последующие узлы, то такое дерево называют бинарным деревом.

*Бинарное дерево* – это конечное множество элементов, которое либо пусто, либо содержит элемент (*корень*), связанный с двумя различными бинарными деревьями, называемыми *левым и правым поддеревьями*. Каждый элемент бинарного дерева называется *узлом*. Связи между узлами дерева называются его *ветвями*.

Способ представления бинарного дерева:

*struct BinareTree // структура узла бинарного дерева*

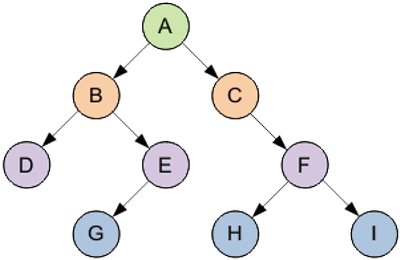
*{*

*DataField data\_field; //поле для данных (информационной части)*

*BinareTree\* left = nullptr; // левый потомок*

*BinareTree\* right = nullptr; // правый потомок*

*};*



A – корень дерева;

В – корень левого поддерева;

С – корень правого поддерева.

Корень дерева расположен на уровне с минимальным значением. Узел D, который находится непосредственно под узлом B, называется *потомком* B. Если D находится на уровне i, то B – на уровне i-1. Узел B называется *предком* D.

Максимальный уровень какого-либо элемента дерева называется его *глубиной* или *высотой*. Если элемент не имеет потомков, он называется *листом* или *терминальным узлом* дерева. Остальные элементы – *внутренние узлы* (узлы ветвления).

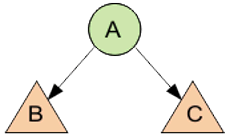
Число потомков внутреннего узла называется его *степенью*. Максимальная степень всех узлов есть степень дерева. Число ветвей, которое нужно пройти от корня к узлу x, называется *длиной пути* к x. Корень имеет длину пути равную 0; узел на уровне i имеет длину пути равную i.

Бинарное дерево применяется в тех случаях, когда в каждой точке вычислительного процесса должно быть принято одно из двух возможных решений.

Имеется много задач, которые можно выполнять на дереве.

Распространенная задача – выполнение заданной операции **p** с каждым элементом дерева. Здесь p рассматривается как параметр более общей задачи посещения всех узлов или задачи обхода дерева. Если рассматривать задачу как единый последовательный процесс, то отдельные узлы посещаются в определенном порядке и могут считаться расположенными линейно.

Способы обхода дерева. Пусть имеем дерево, где A – корень, B и C – левое и правое поддеревья.



Существует три способа обхода дерева:

1. Обход дерева сверху вниз (в прямом порядке): A, B, C – префиксная форма.
2. Обход дерева в симметричном порядке (слева-направо): B, A, C – инфиксная форма.
3. Обход дерева в обратном порядке (снизу-вверх): B, C, A – постфиксная форма.

*Бинарное (двоичное) дерево поиска* – это бинарное дерево, для которого выполняются следующие дополнительные условия (свойства дерева поиска):

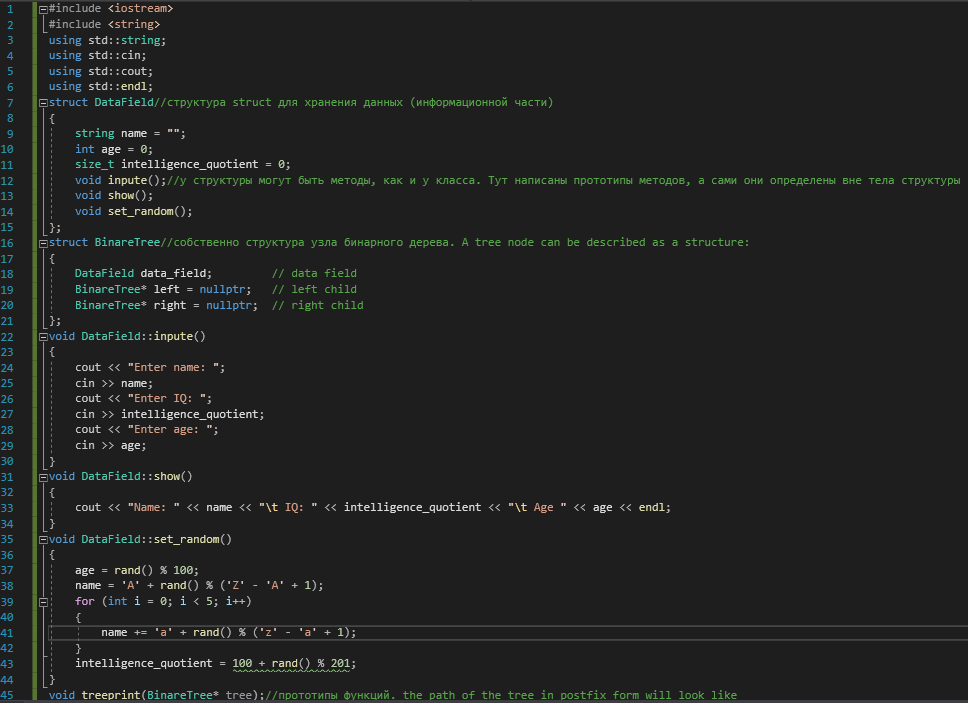
1. оба поддерева – левое и правое, являются двоичными деревьями поиска;
2. у всех узлов левого поддерева произвольного узла X значения ключей данных меньше, чем значение ключа данных самого узла X;
3. у всех узлов правого поддерева произвольного узла X значения ключей данных не меньше, чем значение ключа данных узла X.

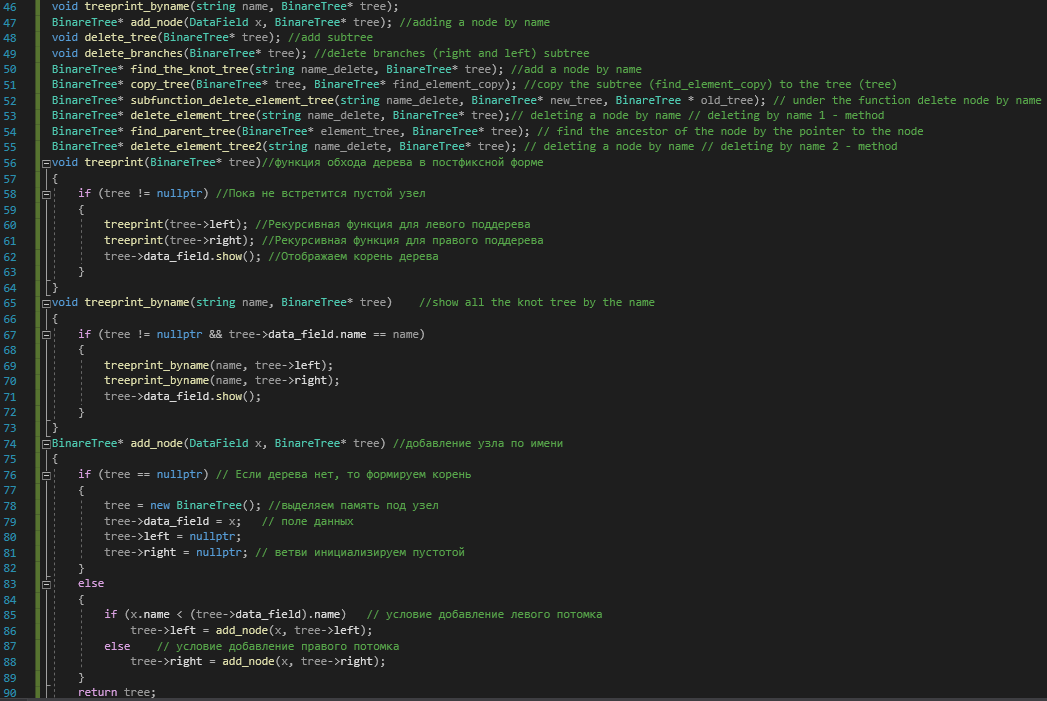
Данные в каждом узле должны обладать ключами, на которых определена операция сравнения меньше (**<**).

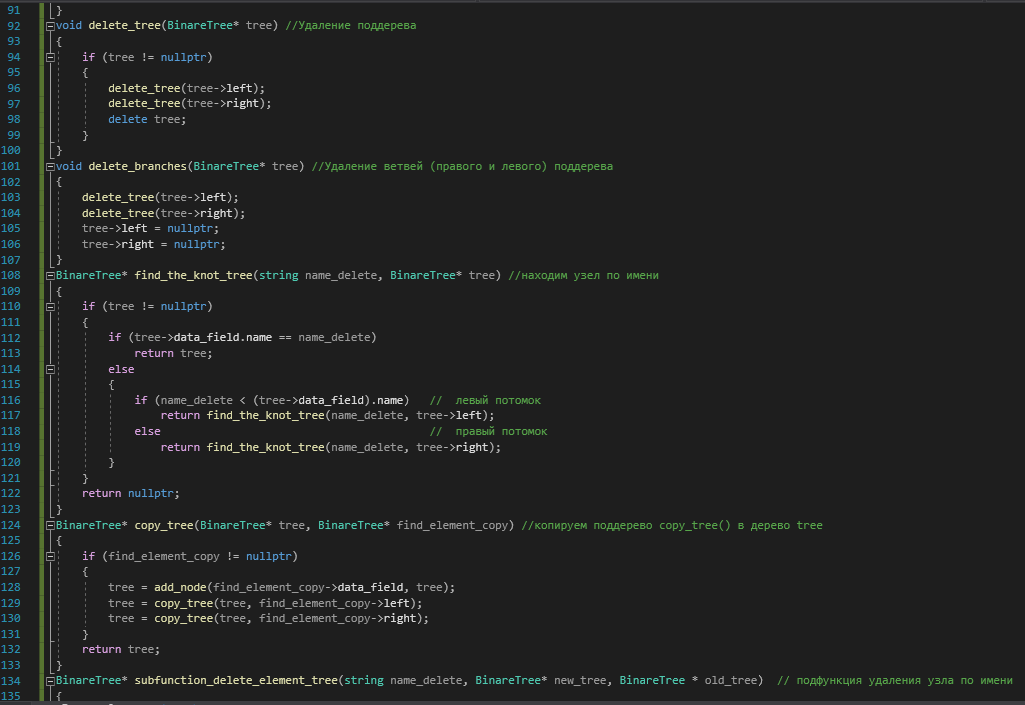
Как правило, информация, представляющая каждый узел, является записью (структурой struct), а не единственным полем данных.

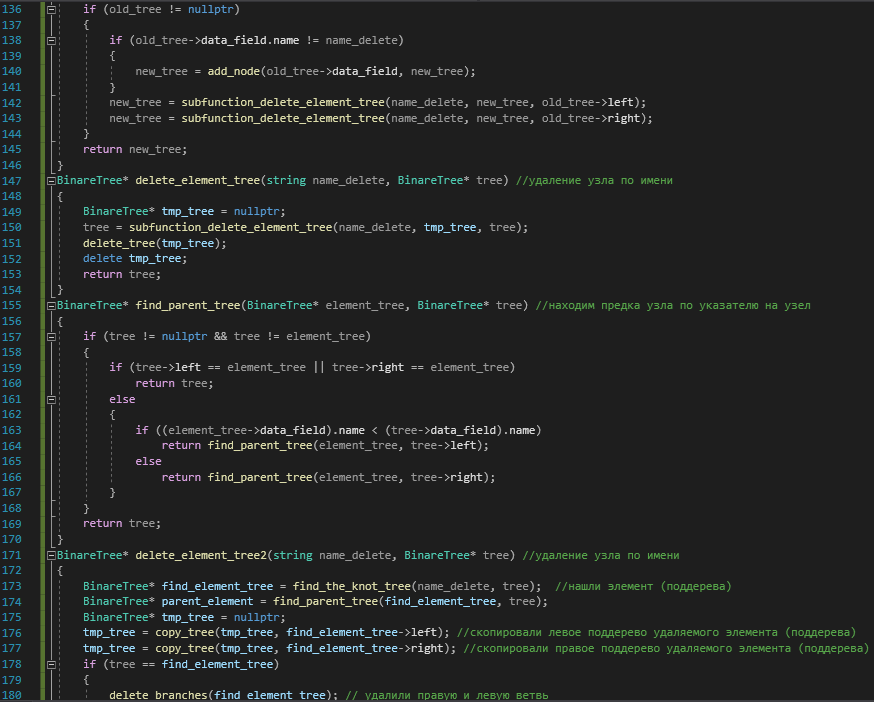
1. **Пример выполнения программы**

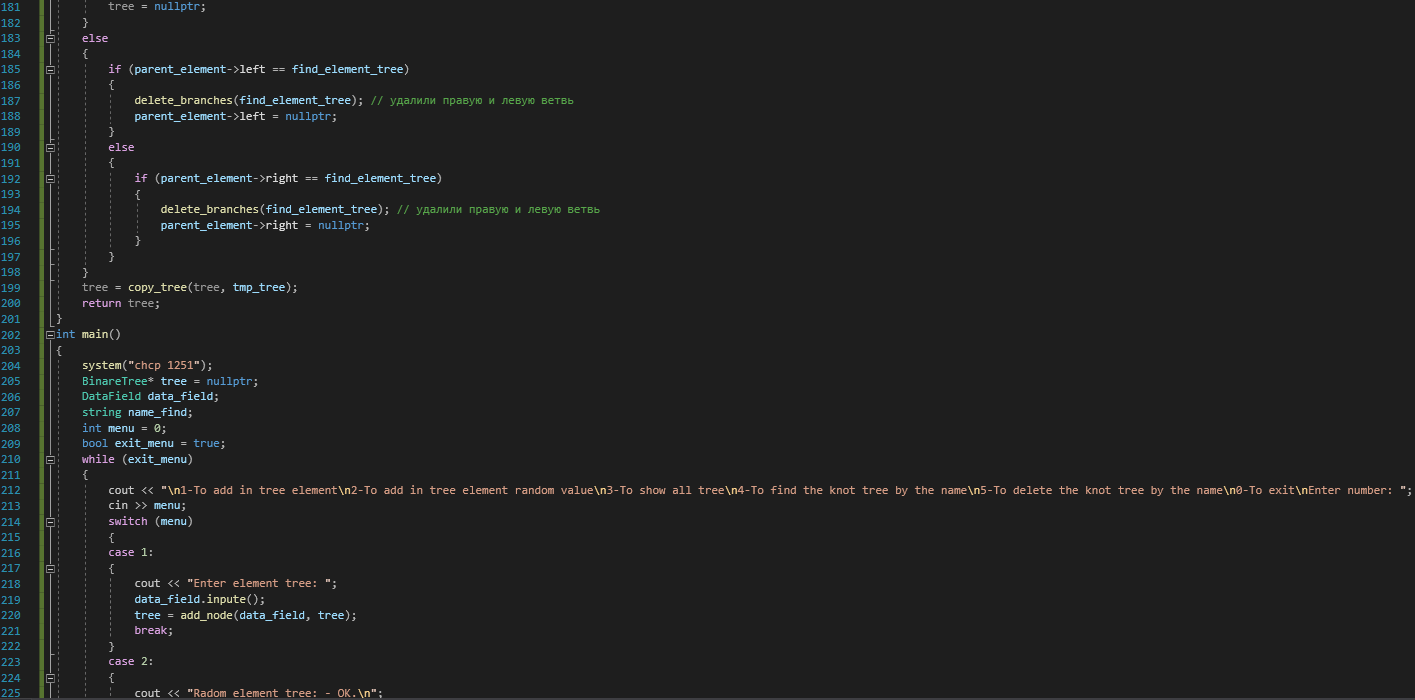
В примере рассмотрено бинарное дерево, хранящее данные о людях (имя, возраст, IQ) и функции добавления, удаления и поиска узла бинарного дерева по имени.

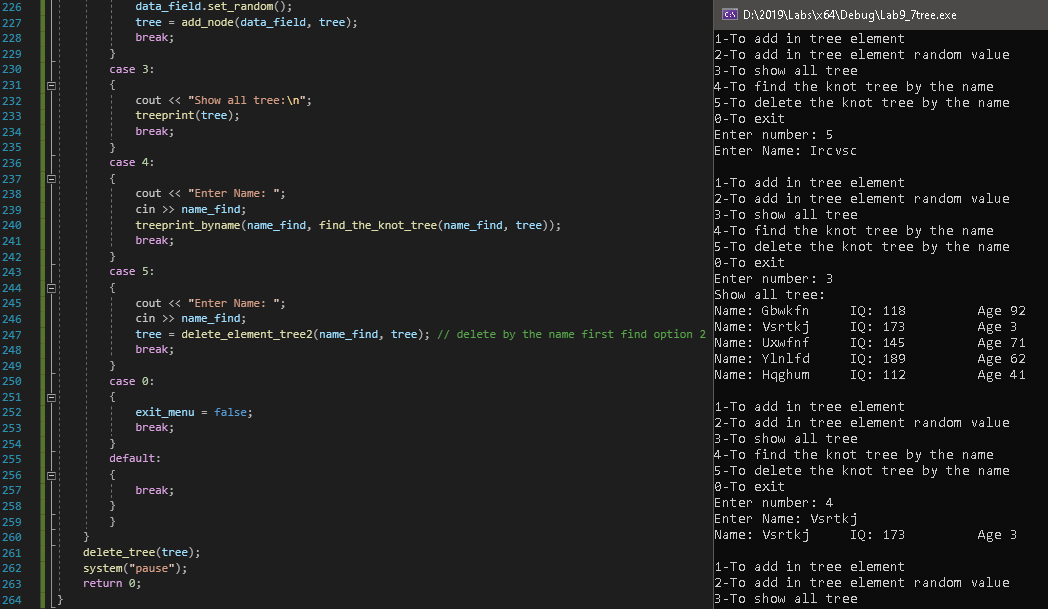












1. **Задания по вариантам**

В вашей программе должны быть реализованы функции:

1. Добавление элемента в Бинарное дерево.
2. Изменение элемента в бинарном дереве (по выбранному параметру).
3. Удаление элемента в бинарном дереве (по выбранному параметру).
4. Подсчет элементов в бинарном дереве.
5. Ваша программа должна иметь консольное меню.

Варианты индивидуальных заданий:

1. Создать бинарное дерево, содержащее информацию об ученике: фамилия, группа, отметка по физике, информатике, истории. Определить средний бал оценок по всем предметам.
2. Создать бинарное дерево, содержащее информацию о клиенте банка: фамилия И.О., номер счета, сумма на счете, процент. Посчитать сумму после начисления процентов.
3. Создать бинарное дерево, содержащее информацию о списке постояльцев гостиницы: паспортные данные, даты приезда и отъезда, номер. Посчитать количество дней пребывания.
4. Сформировать бинарное дерево, содержащее сведения о количестве изделий, собранных сборщиками цеха за неделю. Структурный тип 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.
5. Создать бинарное дерево, содержащее информацию о покупателе: ФИО, домашний адрес покупателя, суммы покупок, размер предоставляемой скидки. Вывести информацию о покупателе и сумме покупок с учетом скидки.
6. Создать бинарное дерево, содержащее информацию о складе, которая включает в себя: наименование товара, количество единиц товара, цену единицы и дату поступления товара на склад. Вывести информацию о товаре и его стоимости.
7. Создать бинарное дерево, содержащее 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 заданным временем.
8. Создать бинарное дерево, содержащее 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.
9. Создать бинарное дерево, содержащее информацию об участнике спортивных соревнований: содержит наименование страны, название команды, ФИО игрока, игровой номер, возраст, рост и вес. Вывести информацию о ИМТ игрока.
10. Создать бинарное дерево, содержащее 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.
11. Создать бинарное дерево, содержащее информацию о книге: регистрационный номер книги, автор, название, год издания, издательство, количество страниц. Вывести возраст книги.
12. Создать бинарное дерево, содержащее информацию о сотруднике предприятия: содержит ФИО, номер отдела, должность, дату начала работы. Вывести стаж сотрудника.
13. Создать бинарное дерево, содержащее информацию: в справочной 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 заданным временем.
14. Создать бинарное дерево, содержащее информацию: радиоателье хранят 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.
15. Создать бинарное дерево, содержащее информацию о горожанине: фамилия И.О., дата рождения, адрес. Посчитать возраст.
16. **Контрольные вопросы**
17. Что за структура данных бинарное дерево?
18. Для моделирования каких данных целесообразно использовать бинарное дерево?
19. Как получить доступ к элементу бинарного дерева?
20. В чем разница левых потомков от правых?
21. По какому принципу добавляться новые элементы в бинарное дерево
22. Как происходит обход всего бинарного дерева?
23. В чем разница между постфиксной, инфиксной и префиксной форме обхода дерева?
24. Как происходит удаление узла бинарного дерева?

**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 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |