Министерство образования и науки РФ

Пермский национальный исследовательский политехнический университет

Электротехнический факультет

Кафедра Информационные технологии и автоматизированные системы

Лабораторная работа № 1

Тема: «Бинарные деревья»

Выполнил: студент группы РИС-22-2б

Мизёв В.А. ф

Проверил: доцент кафедры ИТАС

Полякова О.А.

г. Пермь – 2024

1. **Цель работы**

Получить практические навыки работы с бинарными деревьями.

1. **Постановка задачи**

- сформировать идеально сбалансированное бинарное дерево, тип информационного поля указан в варианте.

- распечатать полученное дерево.

- выполнить обработку дерева в соответствии с заданием, вывести полученный результат.

- преобразовать идеально сбалансированное дерево в дерево поиска.

- распечатать полученное дерево.

Реализовать операции вставки узла, удаления узла, поиск элемента по ключу.

Реализовать алгоритмы обхода дерева: прямой, симметричный, обратный.

Визуализировать бинарное дерево с использованием любой графической библиотеки.

Пользовательский интерфейс на усмотрение разработчика.

1. **Анализ задачи**

Для визуализации дерева будет использоваться SFML – это мультимедийная библиотека, которая предоставляет возможность работать с графикой, звуками и сетью в своей обёртке.

В качестве базы бинарного дерева взять АВЛ дерево. Данное дерево также будет являться деревом поиска.

Основная идея в реализации визуализации дерева состоит из написания следующих классов: Map и TreeRenderer. Класс Map реализует в себе АВЛ дерево с соответствующими методами. Класс TreeRenderer занимается отрисовкой дерева.

Основные методы класса Map продемонстрированы на рисунке 1

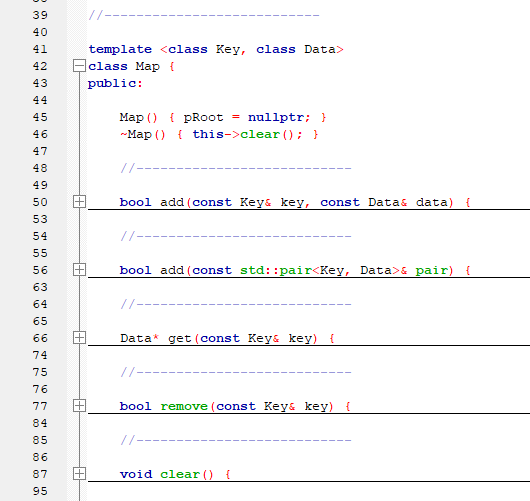


Рисунок 1 - Основные методы класса Map

Метод add() для добавления элементов в дерево, где key и data, ключ и данные соответственно.

Метод get() нужен для получения значения по ключу.

Метод remove() для удаления элемента по ключу.

Метод clear() для очистки бинарного дерева.

Также для классического бинарного дерева была описана следующая структура для узла, которая продемонстрирована на рисунке 2.

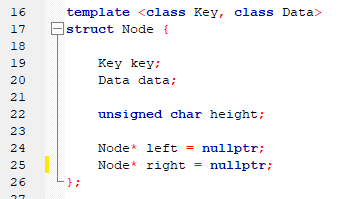


Рисунок 2 - Структура для узла дерева

Где key, data – ключ, значение соответственно, height высота поддерева с корнем в данном узле, left и right указатели на левое и правое поддеревья.

Также для прямого, обратного и симметричного обхода дерева были написаны методы: preorder, postorder и inorder.

Методы getPrintVertical() и getPrintHorizontal() – возвращают строку с вертикальной и горизонтальной печатью дерева.

Для связывания двух классов была придумана структура, приведённая на рисунке 3.

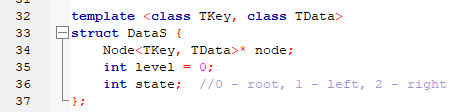


Рисунок 3 - Структура для отрисовки

Для связывания двух классов Map и TreeRenderer, у класса Map придуман метод getTree(), который возвращает вектор со всеми узлами, со структурой из рисунка 3. Сама структура состоит их указателя на узел, уровень узла и состояния, который показывает в каком поддереве данный узел существует, где 0 это корневой узел, 1 левое поддерево, 2 правое поддерево.

Основные методы класса TreeRenderer.

Существует метод setSize(), который позволяет создать область определённого размера для отрисовки содержимого дерева.

Метод moveSelection(), данный метод нужен для перемещения между узлами.

Метод buildFromVector(), строит дерево на базе данных класса Map, который связан с методом getTree().

Также для отрисовки отдельного элемента узла, была сформирована структура, приведённая на рисунке 4.

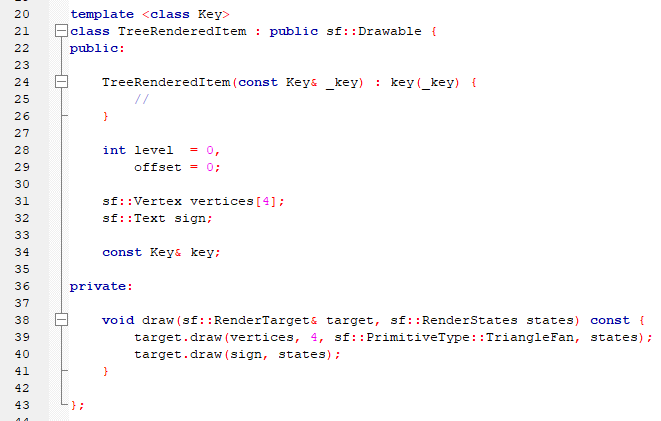


Рисунок 4 - Структура для отрисовки узла

Для удобства отрисовки и перемещения вида, класс TreeRenderer наследуется от класса sf::Drawable и sf::Transformable, где у класса sf::Drawable необходимо реализовать абстрактный метод draw(), а для корректной отрисовки устанавливать текущий transform в sf::RenderStates у метода draw(), который передаётся в качестве аргумента.

1. **Результаты работы программы**

Отрисовка дерева приведена на рисунке 5.

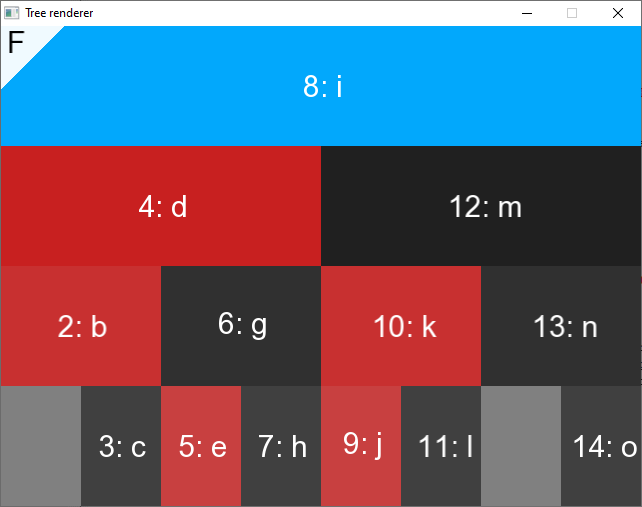


Рисунок 5 - Отрисовка дерева

Для вызова меню с помощью необходимо нажать на кнопку «F» на клавиатуре. Окно с помощью продемонстрировано на рисунке 6.

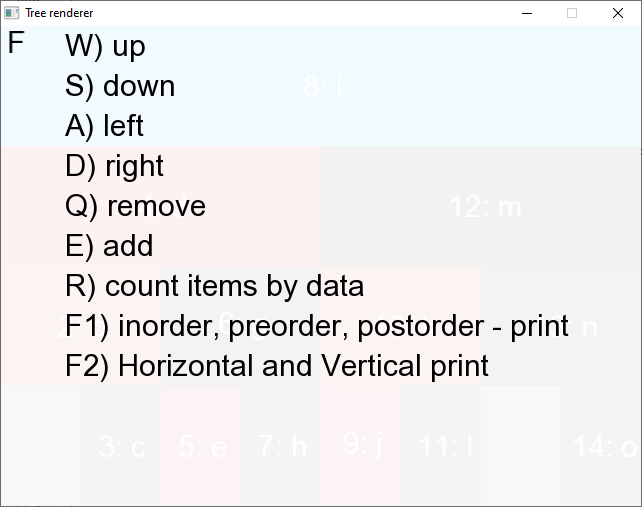


Рисунок 6 - Окно с помощью

Для перемещения по узлам дерева использовать кнопки «wasd», где w- вверх, s – вниз, a – влево, d – вправо.

Для удаления узла нажать на кнопку «Q».

Для добавления нового узла нажать кнопку E.

Для поиска количества заданных данных необходимо нажать кнопку «R»

Для печати обхода дерева использовать F1 или F2 (данные выводятся в консоль).

На рисунке 7 продемонстрирована обходы дерева inorder, preorder и postorder.

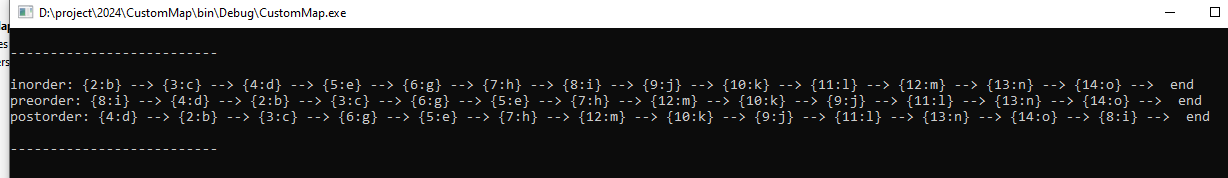


Рисунок 7 - inorder, preorder, postorder

На рисунке 8 продемонстрирована горизонтальная и вертикальная печать дерева.



Рисунок 8 - Горизонтальная и вертикальная печать дерева

1. **Тестирование алгоритмов. Задание 1. Вариант 1.**

Тип информационного поля char. Найти количество элементов с заданными данными.

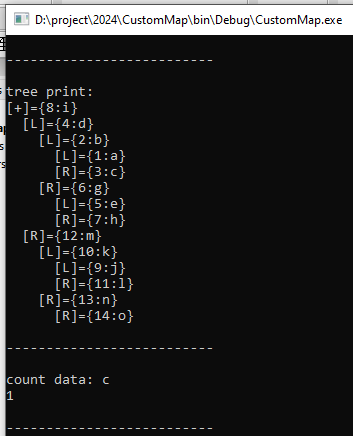


Рисунок 9 - Результат работы программы

**Заключение**

В результате работы был реализован визуализатор дерева, который позволяет добавлять узел, удалять узел по ключу, обходить дерево: прямой обход, симметричный обход, обратный обход, а также позволяет вывести горизонтальную печать и вертикальную печать дерева. Цель получить практический навык работы с деревьями достигнута.