Міністерство освіти і науки України

Івано-Франківський національний технічний університет нафти і газу

Кафедра ІПЗ

**Лабораторна робота №5**

**Тема: “Дерева ”**

Виконав

ст.гр. ІП-22-1

Токарєв С.О.

Перевірив:

Григорчук Л.І.

Івано-Франківськ

2023

**Теоретичні відомості:**

Дерева є важливими структурами даних, які широко використовуються в програмуванні. В Java дерева можна використовувати для вирішення різноманітних задач, таких як зберігання та пошук даних, сортування, обхід структур і багато інших.

Операції над деревами:

-Вставка (Insertion): Додавання нового елемента у дерево.

-Видалення (Deletion): Видалення вузла з дерева.

-Пошук (Search): Знаходження вузла з певним значенням.

-Обхід (Traversal): Проходження всіх вузлів дерева у певному порядку.

**Постановка задачі:**

1. Побудувати двійкове дерево пошуку з цілих чисел, що вводиться. Вивести його на екран у вигляді дерева. Знайти вершину, яка містить задане число. Визначити максимальний елемент в цьому дереві.

2. Побудувати двійкове дерево пошуку з букв рядка, що вводиться. Вивести його на екран у вигляді дерева. Знайти букви, що зустрічаються більше одного разу. Видалити з дерева ці літери. Вивести елементи дерева, що залишилися, при його постфіксному обході.

3. Побудувати двійкове дерево пошуку, в вершинах якого знаходяться слова з текстового файлу. Вивести його на екран у вигляді дерева. Визначити кількість вершин дерева, що містять слова, які починаються на зазначену букву. Видалити з дерева ці вершини.

**Код:**

package lab5;

import java.util.\*;

public class Main {

public static void menu() {

Scanner sc = new Scanner(System.***in***);

int k = 0;

do {

System.***out***.println("1.Task #1: ");

System.***out***.println("2.Task #2: ");

System.***out***.println("3.Task #3: ");

System.***out***.println("4.Exit: ");

k=sc.nextInt();

switch(k){

case 1:

IntTree intTree = new IntTree();

System.***out***.print("Enter the number amount: ");

int amount = sc.nextInt();

for(int i=0;i<amount; i++) {

System.***out***.print("Enter the number: ");

int num = sc.nextInt();

intTree.insert(num);

}

System.***out***.println("Tree:");

intTree.printTree();

System.***out***.println("Max number");

System.***out***.println(intTree.findMax());

break;

case 2:

CharTree charTree = new CharTree();

System.***out***.println("Enter the word");

sc.nextLine();

String word =sc.nextLine();

charTree.insertString(word);

charTree.printTree();

case 3:

StringTree stringTree = new StringTree();

System.***out***.print("Enter the words amount: ");

amount = sc.nextInt();

sc.nextLine();

for(int i=0;i<amount; i++) {

System.***out***.print("Enter the word: ");

word = sc.nextLine();

stringTree.insert(word);

}

stringTree.printTree();

System.***out***.print("Enter the starting letter:");

char l = sc.next().charAt(0);

stringTree.deleteNodesStartingWithLetter(l);

stringTree.printTree();

}

}while(k!=4);

sc.close();

}

public static void main(String[] args) {

*menu*();

}

}

public void printTree() {

printTree(root, "", false);

}

public void insertString(String word) {

for(char c: word.toCharArray()) {

insert(c);

}

private int countNodesStartingWithLetter(Node root, char letter) {

if (root == null) {

return 0;

}

int count = 0;

if (root.data.charAt(0) == letter) {

count = 1;

}

count += countNodesStartingWithLetter(root.left, letter);

count += countNodesStartingWithLetter(root.right, letter);

return count;

}

public int countNodesStartingWithLetter(char letter) {

return countNodesStartingWithLetter(root, letter);

}

private Node deleteNodesStartingWithLetter(Node root, char letter) {

if (root == null) {

return null;

}

if (root.data.charAt(0) < letter) {

root.right = deleteNodesStartingWithLetter(root.right, letter);

} else if (root.data.charAt(0) > letter) {

root.left = deleteNodesStartingWithLetter(root.left, letter);

} else {

root = deleteNode(root, letter);

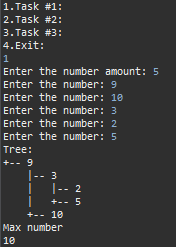
root = deleteNodesStartingWithLetter(root, letter); // рекурсивно видаляємо інші співпадаючі елементи

}

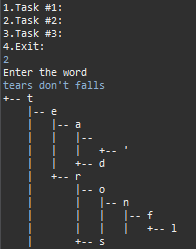
return root;

}

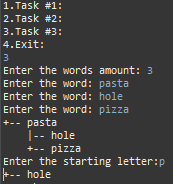
**Скріншоти:**



Завдання №1, додавання чисел в дерево



Завдання№2, додавання літер до дерева



Завдання №3, додавання слів до дерева і видалення їх

**Висновки:**

У цій лабораторній роботі я навчився працювати із деревами, використовувати стандартні методи, додавати різні типи даних до дерев і виводити дерева у правильному вигляді.