МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ”ЛЬВІВСЬКА ПОЛІТЕХНІКА”

Кафедра інформаційних

систем та мереж

Звіт з лабораторної роботи №7

з дисципліни "Технології програмування в розподілених інформаційних системах"

на тему:

**"Дослідження проблем обміну даними, синхронізації й швидкодії**

**багатопотокової програми"**

Варіант №24

Виконав:

Студент групи КН-38

Стасишин О.З.

Прийняв:

доцент каф. ІСМ, к.т.н.

Щербак С. С.

ЛЬВІВ – 2018

**Мета роботи:** набуття навичок програмування дерев.

**Завдання на роботу:**

Розробити багатопотокові засоби динамічного збереження дерев та виконання дій над ними згідно варіанту.

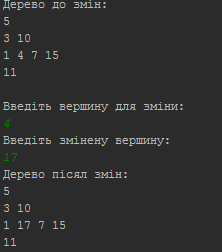
**Індивідуальне завдання:**



**Текст програми на мові Java:**

import java.util.LinkedList;  
import java.util.Queue;  
  
public class BTree {  
 private BTNode root;  
  
 public BTree(int value) {  
 this.root = new BTNode(value);  
 }  
  
  
 public BTree insertBST(int value) {  
 BTree tree = this;  
 tree.root = addBST(tree.root, value);  
 return tree;  
 }  
  
 private BTNode addBST(BTNode currentNode, int value) {  
  
 if (currentNode == null) {  
 return new BTNode(value);  
 }  
 else {  
 if (value <= currentNode.value) {  
 currentNode.left = addBST(currentNode.left, value);  
 }  
 else {  
 currentNode.right = addBST(currentNode.right, value);  
 }  
 }  
 return currentNode;  
 }  
  
  
  
 public void printLinebyLine(BTree root){  
 if(root == null){  
 return;  
 }  
 final Queue<BTNode> queue = new LinkedList<>();  
 queue.add(root.root);  
 while(!queue.isEmpty()){  
 final int size = queue.size();  
 for (int i = 0; i < size; i++) {  
 BTNode node = queue.remove();  
 System.*out*.print(node.value + " ");  
 if(node.left != null){  
 queue.add(node.left);  
 }  
 if(node.right != null){  
 queue.add(node.right);  
 }  
 }  
 System.*out*.println();  
 }  
 }  
  
 public void levelOrderTraversal(BTree root) {  
 Queue<BTNode> queue=new LinkedList<>();  
 queue.add(root.root);  
 while(!queue.isEmpty())  
 {  
 BTNode tempNode=queue.poll();  
 System.*out*.print(tempNode.value + " ");  
 if(tempNode.left!=null)  
 queue.add(tempNode.left);  
 if(tempNode.right!=null)  
 queue.add(tempNode.right);  
 }  
 }  
  
  
  
 public void search(int value , int changeValue){  
 search(root , value , changeValue);  
 }  
 private void search(BTNode node, int value , int changeValue) {  
 if (node != null) {  
 if (node.value == value) {  
 node.value = changeValue;  
 }  
 else {  
 search(node.left, value , changeValue);  
 search(node.right, value , changeValue);  
 }  
 }  
 }  
  
  
 private class BTNode {  
 int value;  
 BTNode left;  
 BTNode right;  
  
 public BTNode(int value) {  
 this.value = value;  
 left = null;  
 right = null;  
 }  
  
 }  
}

**Результати виконання програми:**



*Рис. 1. Заповнення дерева та виведення вершин, значення яких менше на одиницю за корінь*

**Висновок:** програма призначена для роботи з деревом. Програма реалізує усі необхідні. На цій лабораторній роботі було розробити засоби динамічного збереження дерев та реалізовано виконання дій над ними згідно з варіантом. Програма написана на мові Java у середовищі IntelliJ.