

Міністерство освіти і науки України
Національний технічний університет України «Київський політехнічний
інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки

Кафедра ІІІ

Звіт

з лабораторної роботи №5 з дисципліни
«Алгоритми та структури даних 2. Структури даних»

„Деревовидні структури даних”

Виконав ІІ-45 Янов Богдан Євгенійович

Перевірів Соколовський Владислав Володимирович

Київ 2025

Лабораторна робота №5

Деревовидні структури даних

Мета: вивчити основні підходи формалізації та імплементації алгоритмів побудови та обробки базових деревовидних структур даних.

Дерево - це рекурсивна структура даних, що складається з вузлів (вершин). Кожен вузол може мати підвузли (нащадки). Один вузол є коренем (root), а вузли без нащадків називаються листям (термінальні вершини).

Варіант 27 - Побудувати дерево, елементами якого є символи. Визначити і вивести на друк усі термінальні вершини (листя) цього дерева.

Псевдокод алгоритмів

```
struct Node {
    value
    children
}

add_child(node, child) {
    push(node.children, child)
}

print_leaves(node) {
    if len(node.children) > 0 {
        print(node.value)
    } else {
        for each child in node.children {
            print_leaves(child)
        }
    }
}
```

Вихідний код

```
struct Node {
    value: char,
    children: Vec<Node>,
}

impl Node {
    fn new(value: char) -> Self {
        Node {
            value,
```

```

        children: Vec::new(),
    }
}

fn add_child(&mut self, child: Node) {
    self.children.push(child);
}

fn print_leaves(&self) {
    if self.children.is_empty() {
        print!("{}", self.value);
    } else {
        for child in &self.children {
            child.print_leaves();
        }
    }
}
}

fn main() {
    //      'A'
    //      /  \
    //      'B'  'C'
    //      / \   \
    //      'D' 'E'  'F'

    let mut root = Node::new('A');

    let mut b = Node::new('B');
    b.add_child(Node::new('D'));
    b.add_child(Node::new('E'));

    let mut c = Node::new('C');
    c.add_child(Node::new('F'));

    root.add_child(b);

```

```
root.add_child(c);  
  
root.print_leaves();  
}
```

Приклад роботи

```
λ Partur lab5 → λ git master* → cargo run  
Finished `dev` profile [unoptimized + debuginfo] target(s) in 0.00s  
Running `target/debug/lab5`  
D E F
```

Висновок

У ході лабораторної роботи було побудовано дерево з символів та розроблено функцію для знаходження всіх термінальних вершин (листя). Робота дозволила закріпити навички роботи з рекурсією, списками та обробкою деревоподібних структур у функціональному стилі.