Львівський національний університет імені Івана Франка

Факультет електроніки та комп’ютерних технологій

**Звіт**

Про виконання лабораторної роботи №1

**Динамічні структури даних. Нелінійні списки (дерева)**

Виконав

Студент групи ФеП-21

Берніш Микола

Перевірив:

Зіновій Любунь

Львів 2020

**Мета** роботи: навчитися працювати з бінарними деревами, та бінарними деревами пошуку

**Частина 1**

Хід роботи

1. Створимо бібліотеку Tree.py, у ній ми оголосимо клас “Node”(рис.1)



Рис.1

1. У бібліотеці Tree.py реалізуємо функції CreateTree() та ShowTree(), для створення та відображення бінарного дерева(Рис.2)

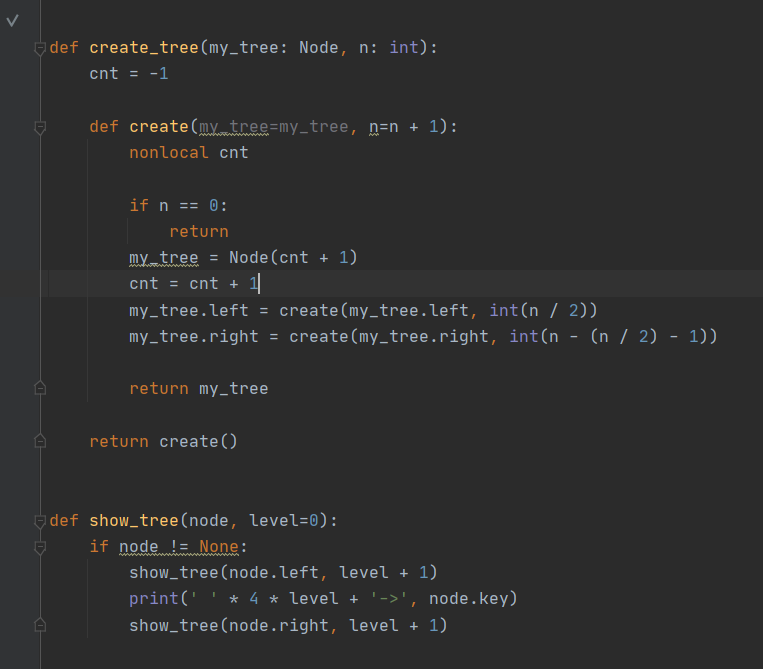


Рис.2

1. У бібліотеці Tree.py реалізуємо функції PrefixOrder(), PostfixOrder() та InfixOrder для обходу бінарних дерев(Рис.3)

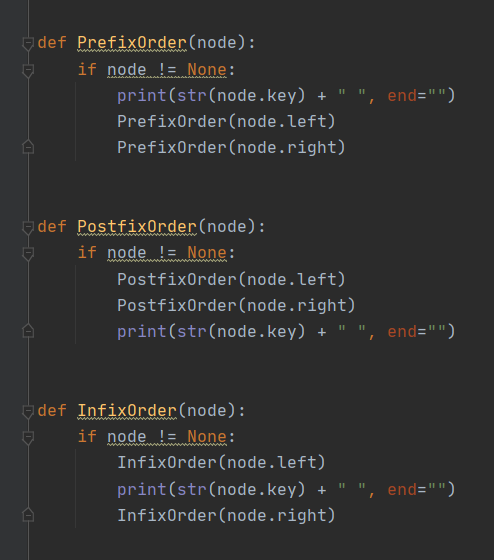


Рис.3

1. В окремому проекті підключимо бібліотеку Tree.py -створимо дерево, та обійдемо його (Рис.4), результат – (Рис.5)

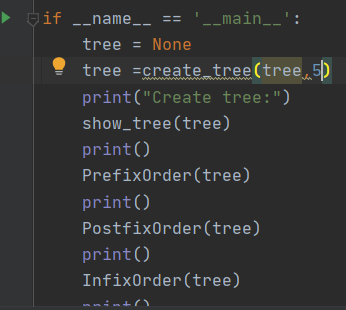


Рис.4

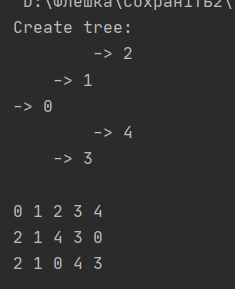


Рис.5

**Частина 2**

Хід роботи

1. У бібліотеці Tree, згідно описаних вище алгоритмів, реалізуємо функції SearchNodeBST(…), CreateRootBST(…), InsertNodeBST(…), DeleteNodeBST(…), SuccessorNodeBST(…), PredecessorNodeBST(…) для виконання операцій пошуку даних, створення кореневого вузла, додавання, видалення та пошуку термінальних вузлів у бінарному дереві пошуку (Рис.6, Рис.7,Рис.8,Рис.9)

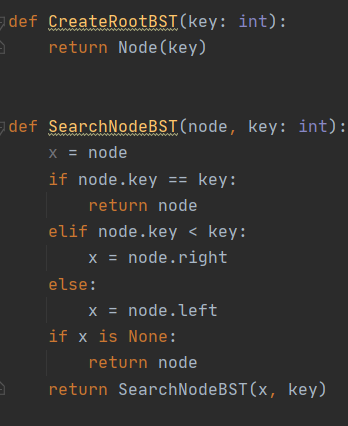
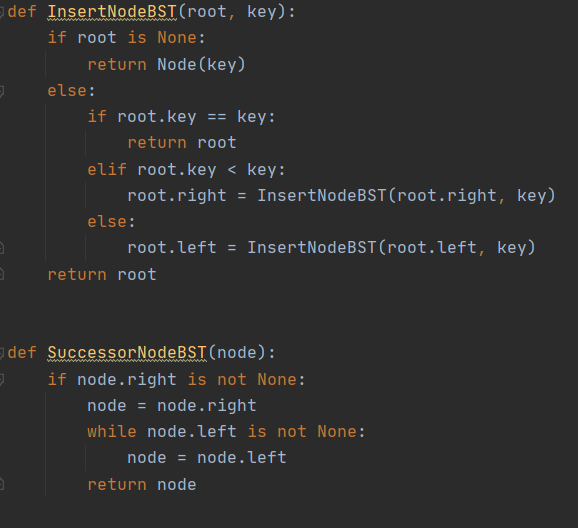
 

Рис.6 Рис.7

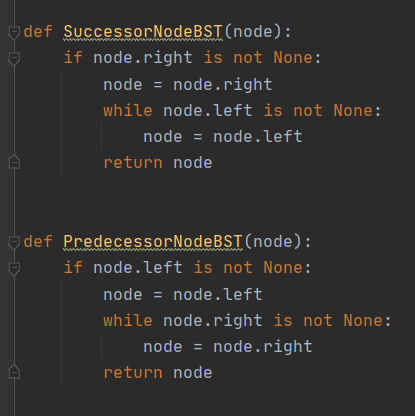
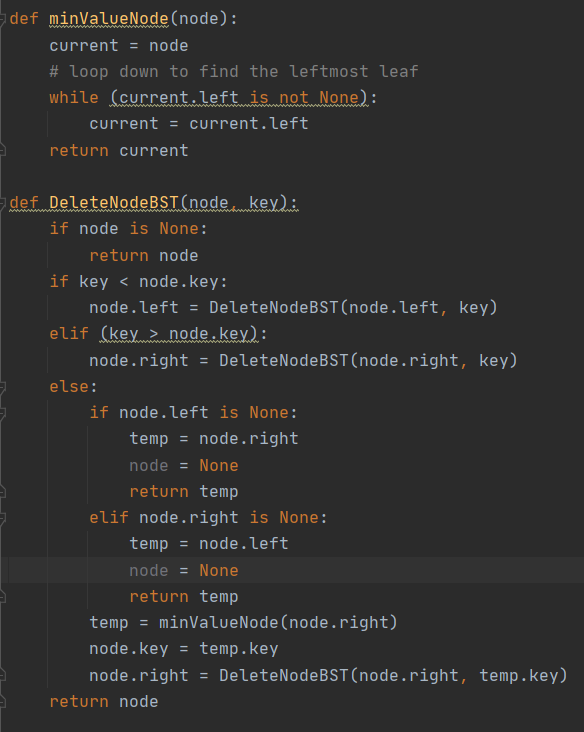
 

Рис.8 Рис.9

1. В окремому проекті підключимо бібліотеку Tree.py – створимо бінарне дерево пошуку та використаємо створений функціонал (Рис.10) та результат (Рис.11)

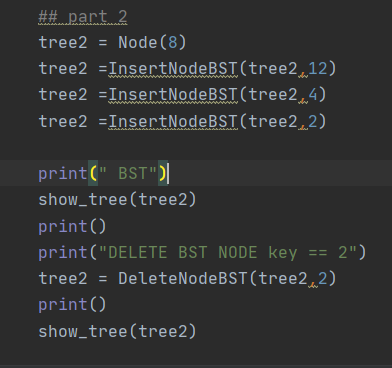


Рис.10

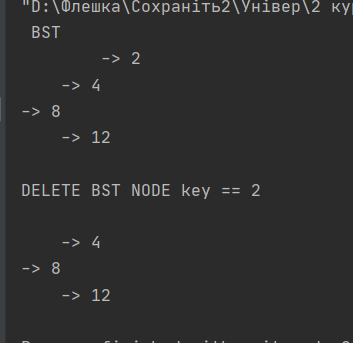


Рис.11

**Висновок:** Отже я навчився створювати та працювати з бінарними деревами, та бінарними деревами пошуку