**Міністерство освіти і науки України**

**Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського»**

**Факультет інформатики та обчислювальної техніки**

**Кафедра інформатики та програмної інженерії**

Звіт

з лабораторної роботи №6 з дисципліни

«Основи програмування

2. Модульне програмування»

«Дерева»

Виконав студент ІП-14 Нікулін Павло Юрійович

(шифр, прізвище, ім'я, по батькові)

Перевірив Вітковська Ірина Іванівна

(прізвище, ім'я, по батькові)

**Лабораторна робота №6**

**Дерева**

**Мета**: вивчити особливості організації і обробки дерев.

**Хід роботи**

**Задача**: Побудувати дерево наступного типу:

n

/ \

2 2

…

n-1 n-1

/ \ / \

n n n n

*Розв’язання*

1. **Постановка задачі**:

2. Побудова **математичної моделі**. Складемо таблицю імен змінних для двох мов.

**C++**

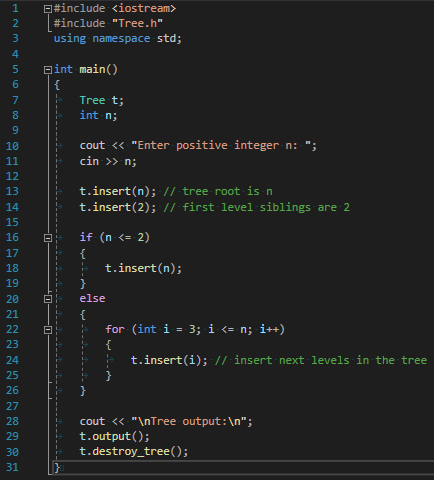
|  |  |  |  |
| --- | --- | --- | --- |
| **Змінна** | **Тип** | **Ім’я** | **Призначення** |
| Хедер до файлу з функціями | Файловий | TREE.H | Файл |
| Файл з функціями | Файловий | TREE.CPP | Файл |
| Об’єкт класу Tree | Tree | T | Результат |
| Корінь дерева | Цілий | N | Початкове дане |
| Вузол дерева | Структура | NODE | Проміжне дане |
| Значення вузла | Цілий | DATA | Проміжне дане |
| Ліве листя вузла | node\* | LEFT | Результат |
| Праве листя вузла | node\* | RIGHT | Результат |
| Стебло дерева | Структура | TRUNK | Проміжне дане |
| Конструктор класу Trunk | Конструктор Trunk | TRUNK | Метод |
| Попереднє стебло | Trunk\* | PREV\_TRUNK | Проміжне дане |
| З’єднання у дереві | Рядковий | CONNECTION | Проміжне дане |
| Дерево | Клас | TREE | Результат |
| Конструктор класу Tree | Конструктор Tree | TREE | Метод |
| Метод додавання вершини (для виклику) | Універсальний | INSERT | Метод |
| Метод знищення дерева (для виклику) | Універсальний | DESTROY\_TREE | Метод |
| Метод виведення дерева (для виклику) | Універсальний | OUTPUT | Метод |
| Метод додавання вершини | Універсальний | INSERT | Метод |
| Метод знищення дерева | Універсальний | DESTROY\_TREE | Метод |
| Метод стебла дерева | Універсальний | SHOW\_TRUNK | Метод |
| Метод виведення дерева | Універсальний | GRAPHICAL\_TREE | Метод |
| Корінь дерева | node\* | ROOT | Проміжне дане |
| Значення вершини | Цілий | NUM | Проміжне дане |
| Листя дерева | node\* | LEAF | Проміжне дане |
| Поточне стебло | Trunk\* | CURR\_TRUNK | Проміжне дане |
| Поточний вузол/корінь | node\* | CURR\_ROOT | Проміжне дане |
| Попереднє стебло | Trunk\* | PREVIOUS | Проміжне дане |
| Якщо рух по правому стеблу | Булевий | IS\_RIGHT | Проміжне дане |
| Попереднє з’єднання | Рядковий | PREV\_CONNECTION | Проміжне дане |

**Випробування коду**

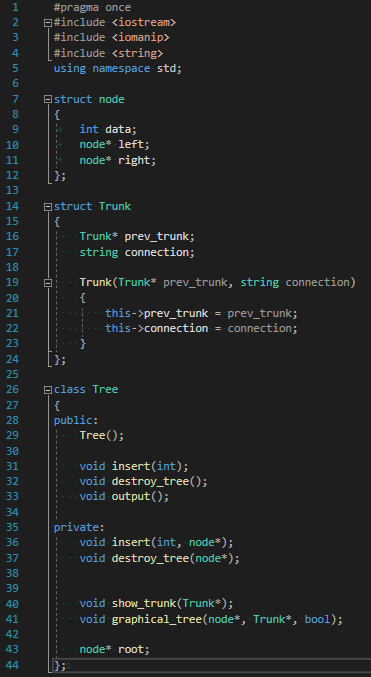
**C++**

*Код*

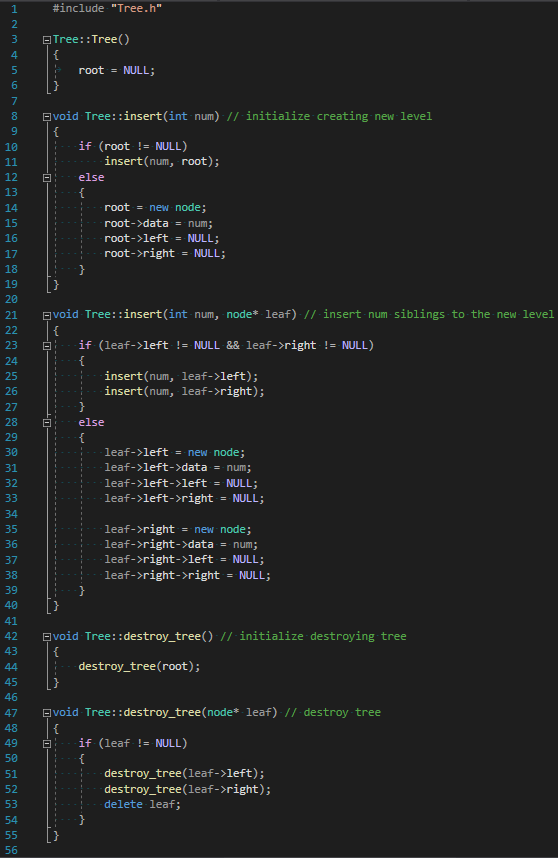
lab1.cpp

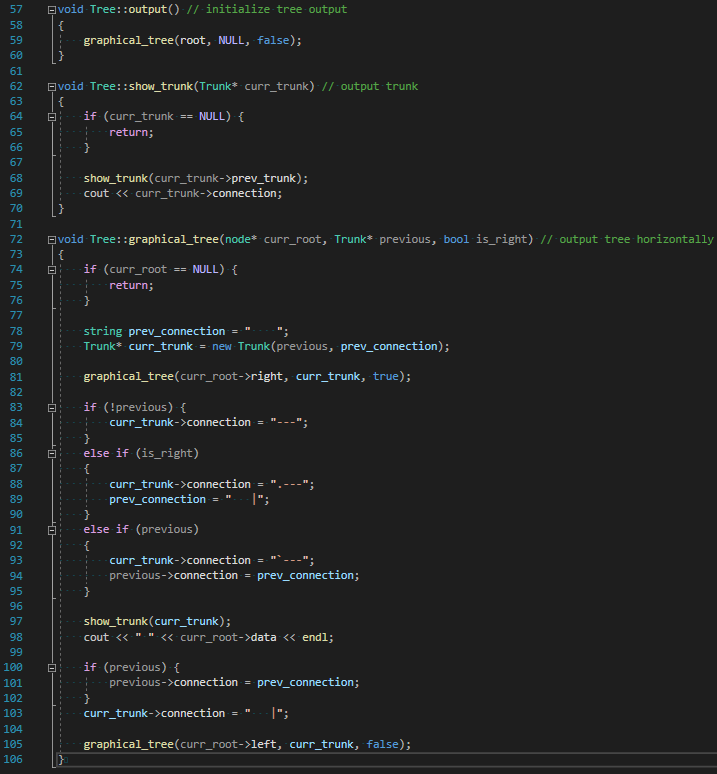


Tree.h

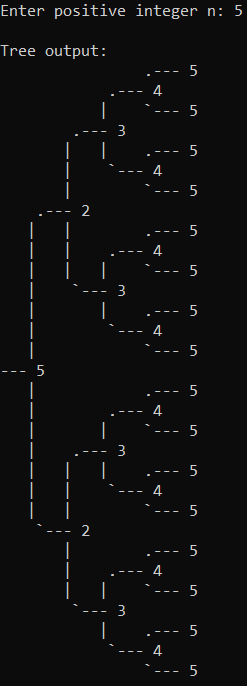


Tree.cpp





*Результат*



**Висновок**

Під час виконання лабораторної роботи було досліджено особливості організації і обробки дерев. Було створено бінарне дерево за макетом з умови. Коренем (нульовий рівень) є ціле число n, його дитини на першому рівні – число 2, наступні рівні – числа від 2 до n, і на останьому рівні всі вершини є числом n. Було реалізовано горизонтальне виведення дерева. Роботу було виконано на мові програмування С++ з використанням ООП (класи та об'єкти). Програма працює коректно, відповідно до умови задачі.