

Основи програмування 2. Модульне програмування

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»

Варіант 29

Виконав студент

ПІ-15 Рибалка Ілля Сергійович
(шифр, прізвище, ім'я, по батькові)

Перевірів

Всечерковська Анастасія Сергіївна
(прізвище, ім'я, по батькові)

Київ 2022

Лабораторна робота 5

Дерева

Мета - вивчити особливості організації і обробки дерев.

Індивідуальне завдання

Варіант 29

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

Розв'язання

Код

main.cpp

```
#include "Tree.h"

int main()
{
    std::string str;

    std::cout << "Введіть набір символів - ";

    std::getline(std::cin, str);

    while(str.size() < 1)
    {
        std::cout << "Введіть набір символів - ";

        std::getline(std::cin, str);
    }

    Tree tree(str);

    std::cout << "Сформоване дерево на основі введених символів:" << std::endl;

    tree.print();
}
```

```

std::cout << std::endl;

std::cout << "Термінальні вершини цього дерева:" << std::endl;

tree.printleafs();

std::cout << std::endl;

return 0;

}

```

Tree.h

```

#pragma once

#include <iostream>

#include <iomanip>

#include <string>

struct Branch

{

    char Data;

    Branch *Left;

    Branch *Right;

    Branch(char (ch)): Data(ch), Left(NULL), Right(NULL) {};

};

class Tree

{

    Branch *root;

    Branch *add(char, Branch *);

    void printleafs(Branch *);

```

```

void print(Branch *);

void printbylvl(int, Branch *);

void remove(Branch *);

public:

    Tree(std::string);

    int height(Branch *);

    void printleaves();

    void print();

    ~Tree();

};

```

Tree.cpp

```

#include "Tree.h"

Tree::Tree(std::string data)
{
    this->root = new Branch(data[0]);

    if (data.size() > 1) for (int i = 1; i < data.size(); i++) this->root = add(data[i],
this->root);
}

Branch *Tree::add(char tData, Branch *node)
{
    if (node)
    {
        if (node->Data >= tData)
        {

```

Основи програмування 2. Модульне програмування

```
        if (!node->Left) node->Left = new Branch(tData);

        else add(tData, node->Left);

    }

    else

    {

        if (!node->Right) node->Right = new Branch(tData);

        else add(tData, node->Right);

    }

}

else node = new Branch(tData);

return node;

}

int Tree::height(Branch *node)

{

    int ln = 0, rn = 0;

    if(node)

    {

        if(node->Left) ln = height(node->Left);

        if(node->Right) rn = height(node->Right);

    }

    return std::max(ln,rn)+1;

}

void Tree::print()
```

```
{

    print(this->root);

}

void Tree::printleafs()

{

    printleafs(this->root);

}

void Tree::print(Branch *node)

{

    int h = height(node);

    int i;

    for (i=1; i<=h; i++)

    {

        printbylvl(i, node);

        std::cout << std::endl;

    }

}

void Tree::printbylvl(int lvl, Branch *node)

{

    if(node)

    {

        if (lvl == 1) std::cout << node->Data << " ";

    }

}
```

```

        else if (lvl > 1)

        {

            printbylvl(lvl-1, node->Left);

            printbylvl(lvl-1, node->Right);

        }

    }

}

```

```
void Tree::printleafs (Branch *node)
```

```

{

    if (node)

    {

        printleafs (node->Left);

        if (!node->Left && !node->Right)

            std::cout << node->Data << " ";

        printleafs (node->Right);

    }

}

```

```
void Tree::remove (Branch *node)
```

```

{

    if (node)

    {

        remove (node->Left);

        remove (node->Right);

    }

}

```

```
        delete node;

    }

}

Tree::~~Tree()

{

    remove(this->root);

}
```

Тестування

```
Введіть набір символів - 1234
Сформоване дерево на основі введених символів:
1
2
3
4

Термінальні вершини цього дерева:
4
MacBook-Pro-Mac:Lab6 mac$ cd "/Volumes/files/S2
Введіть набір символів - qwerty
Сформоване дерево на основі введених символів:
q
e w
r u
t

Термінальні вершини цього дерева:
e t u
```

Висновок

Я вивчив особливості організації і обробки дерев. В ході роботи над лабораторною роботою було створено програму для обробки бінарного дерева, що складається з символів і виводу його листків.