

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 № 5  
з дисципліни «Основи програмування – 2.  
Методології програмування»

«Успадкування та поліморфізм»

Варіант 13

Виконав студент ІП-13 Жмайло Дмитро Олександрович  
(шифр, прізвище, ім'я, по батькові)

Перевірів Вечерковська Анастасія Сергіївна  
(прізвище, ім'я, по батькові)

Київ 2022

## Лабораторна робота 5

### Дерева

### Варіант 13

### Код програми

Побудувати дерево для зберігання даних виду: деталь, її кількість, постачальник. Забезпечити виконання операцій додавання нового елемента у дерево в діалоговому режимі та визначення постачальника найбільшої кількості деталей.

C#

#### Program.cs

```
using System;

namespace Lab5
{
    internal class Program
    {
        static void Main(string[] args)
        {
            Input input = new Input();
            BinaryTree tree = input.InputTree();
            tree.Display();

            string ProviderOfMaxAmount = "";
            tree.FindProvider(ref ProviderOfMaxAmount);
            Console.WriteLine("Provider of max amount of details: " +
ProviderOfMaxAmount);
        }
    }
}
```

## Node.cs

```
using System;
using System.Collections.Generic;
using System.Text;

namespace Lab5
{
    internal class Node
    {
        private Detail _detail;
        private Node _leftNode;
        private Node _rightNode;
        public Node(Detail detail)
        {
            _detail = detail;
        }
        public void Add(Detail detail)
        {
            if (detail.GetAmount() >= _detail.GetAmount())
            {
                if (_rightNode == null)
                {
                    _rightNode = new Node(detail);
                }
                else
                {
                    _rightNode.Add(detail);
                }
            }
            else
            {
                if (_leftNode == null)
                {
                    _leftNode = new Node(detail);
                }
                else
                {
                    _leftNode.Add(detail);
                }
            }
        }
        public void PrintNode(int space)
        {
            if (_detail == null)
                return;

            if (_rightNode != null)
                _rightNode.PrintNode(++space);
            else
                space++;
            for (int i = 0; i < space; i++)
                Console.Write("\t");
            Console.WriteLine(_detail.ToString());
            if (_leftNode != null)
                _leftNode.PrintNode(space);
        }
        public void FindProvider(ref string provider)
        {
            if (_rightNode != null)
                _rightNode.FindProvider(ref provider);
            else
                provider = _detail.GetProvider();
        }
    }
}
```

## BinaryTree.cs

```
using System;
using System.Collections.Generic;
using System.Text;

namespace Lab5
{
    internal class BinaryTree
    {
        public Node _root;

        public void Add(Detail detail)
        {
            if (_root == null)
                _root = new Node(detail);
            else
                _root.Add(detail);
        }

        public void FindProvider(ref string provider)
        {
            if (_root != null)
                _root.FindProvider(ref provider);
        }

        public void Display()
        {
            int dashNumber = 50;
            Console.WriteLine("Your tree is: ");
            Console.WriteLine(new String('-', dashNumber) + "\n");

            _root.PrintNode(0);

            Console.WriteLine("\n" + new String('-', dashNumber));
        }
    }
}
```

## Detail.cs

```
using System;
using System.Collections.Generic;
using System.Text;

namespace Lab5
{
    internal class Detail
    {
        private string name;
        private int amount;
        private string provider;

        public Detail(string name, int amount, string provider)
        {
            this.name = name;
            this.amount = amount;
            this.provider = provider;
        }

        public string GetName()
        {
            return name;
        }
        public int GetAmount()
        {
            return amount;
        }
        public string GetProvider()
        {
            return provider;
        }

        public override string ToString()
        {
            return name + "(" + amount + ")";
        }
    }
}
```

## Input.cs

```
using System;
using System.Collections.Generic;
using System.Text;

namespace Lab5
{
    internal class Input
    {
        public BinaryTree InputTree()
        {
            BinaryTree tree = new BinaryTree();

            int nodesAmount = InputAmount("Enter nodes amount: ");
            Console.WriteLine();
            for (int i = 0; i < nodesAmount; i++)
            {
                Console.WriteLine($"{i+1}) detail: ");
                tree.Add(InputDetail());
                Console.WriteLine();
            }
            return tree;
        }

        private Detail InputDetail()
        {
            string name = InputString("Enter name of detail: ");
            int amount = InputAmount("Enter amount of details: ");
            string provider = InputString("Enter the provider of details: ");
            return new Detail(name, amount, provider);
        }

        private int InputAmount(string message)
        {
            int minAmount = 1;
            Console.Write(message);
            while (true)
            {
                if (int.TryParse(Console.ReadLine(), out int amount) && amount >=
minAmount)
                {
                    return amount;
                }
                else
                {
                    Console.Write("Wrong input. Try again: ");
                }
            }
        }

        private string InputString(string message)
        {
            Console.Write(message);
            while (true)
            {
                string input = Console.ReadLine();
                if (!string.IsNullOrEmpty(input))
                {
                    return input;
                }
                else
                {
                    Console.Write("Wrong input. Try again: ");
                }
            }
        }
    }
}
```

## Тестування:

```
Консоль отладки Microsoft Visual Studio
Enter nodes amount: паляниця
Wrong input. Try again: 5

1) detail:
Enter name of detail: ship
Enter amount of details: 2
Enter the provider of details: nike

2) detail:
Enter name of detail: jeep
Enter amount of details: 5
Enter the provider of details: adidas

3) detail:
Enter name of detail: jeans
Enter amount of details: 1
Enter the provider of details: sport

4) detail:
Enter name of detail: red
Enter amount of details: 7
Enter the provider of details: color

5) detail:
Enter name of detail: cherry
Enter amount of details: 10
Enter the provider of details: berry

Your tree is:
-----
                cherry(10)
               /
            red(7)
           /
        jeep(5)
       /
    ship(2)
   /
  jeans(1)
-----

Provider of max amount of details: berry
```

## Висновки:

На цій лабораторній роботі я застосував на практиці знання щодо реалізації такої динамічної структури даних як дерево. Розробив програму, використовуючи мову програмування C# для виконання поставленої задачі та зробив висновки.