

Додаток 1
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»
«Дерева»
Варіант 12

Виконав студент: ІП-13 Дойчев Костянтин Миколайович

Перевірила: Вечерковська Анастасія Сергіївна

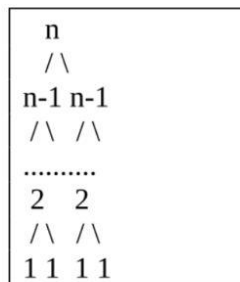
Київ 2021

Лабораторна робота №5

Тема: Дерева

Постановка задачі

12. Побудувати дерево наступного виду:



, де n - додатне ціле число

Код:

Файл - main.cpp

```
1 #include <iostream>
2 #include "Tree.h"
3
4 using namespace std;
5
6 int main() {
7     int rootNumber = 0;
8     bool isValid = false;
9     while (!isValid) {
10         try {
11             cout << "Enter the root number (an integer > 1): ";
12             cin >> rootNumber;
13             if (rootNumber > 1) {
14                 isValid = true;
15             }
16         } catch (exception &e) {
17             cout << "Invalid input. Try again." << e.what() << endl;
18         }
19     }
20
21     cout << "Building the tree... " << endl;
22
23     cout << "Tree: (left branch is blue, right branch is yellow)\n\n" << endl;
24
25     Tree tree(rootNumber);
26     cout << tree << endl;
27
28
29     return 0;
30 }
```

main

Файл: Node.h

```
#ifndef LAB_5_NODE_H
#define LAB_5_NODE_H

class Node {
private:
    Node* left;
    Node* right;
    int data;
public:
    Node(int d);
    Node* getLeft();
    Node* getRight();
    int getData();
    void setLeft(Node* node);
    void setRight(Node* node);
};

#endif //LAB_5_NODE_H
```

Файл: Node.cpp

```
main.cpp x Node.h x Node.cpp x Tree.cpp x CMakeLists.txt x
1 #include "Node.h"
2
3 Node::Node(int d): data(d), left(nullptr), right(nullptr) {};
4
5 Node* Node::getLeft() {
6     return left;
7 }
8
9 Node* Node::getRight() {
10    return right;
11 }
12
13 int Node::getData() {
14    return data;
15 }
16
17 void Node::setLeft(Node* node) {
18    left = node;
19 }
20
21 void Node::setRight(Node* node) {
22    right = node;
23 }
24
25
```

Файл: Tree.h

```
main.cpp x Node.h x Node.cpp x Tree.h x Tree.cpp x CMakeLists.txt x
1  #ifndef LAB_5_TREE_H
2  #define LAB_5_TREE_H
3
4  #define YELLOW  "\033[33m"
5  #define BLUE    "\033[34m"
6
7  #include <iostream>
8  #include <string>
9  #include "Node.h"
10
11 class Tree {
12     Node* root;
13 public:
14     Tree(int data);
15     Node* getRoot();
16
17 private:
18     void build(Node* node, int data);
19     void print(const std::string &prefix, Node* node, bool isLeftBranch);
20     friend std::ostream& operator << (std::ostream& os, Tree &tree);
21 };
22
23
24
25 #endif //LAB_5_TREE_H
26
```

Файл: Tree.cpp

```
#include "Tree.h"

using namespace std;

Tree::Tree(int data) {
    this->root = new Node(data);
    this->build(this->root, data - 1);
}

void Tree::build(Node *node, int data) {
    node->setLeft(new Node(data));
    node->setRight(new Node(data));

    if (data == 1) {
```

```

        return;
    }

    this->build(node->getLeft(), data - 1);
    this->build(node->getRight(), data - 1);
}

Node *Tree::getRoot() {
    return this->root;
}

void Tree::print(const string &prefix, Node *node, bool isLeftBranch) {
    if (node == nullptr) {
        return;
    }

    cout << prefix;
    cout << (isLeftBranch ? "├──" : "└──");

    cout << node->getData() << endl;
    string newPrefix = prefix + (isLeftBranch ? "│   " : "   ");
    this->print(BLUE + newPrefix, node->getLeft(), true);
    this->print(YELLOW + newPrefix, node->getRight(), false);
}

ostream &operator<<(ostream &os, Tree &tree) {
    tree.print("", tree.getRoot(), false);
    return os;
}

```

Дані і консоль:

```
/Users/Kostia/Documents/Programming/kpi/programming-basics/C-plus-plus/bachelor/year-1/semester-2/lab-5/cmake-build-debug/lab_5
```

```
Enter the root number (an integer > 1): -10
```

```
Enter the root number (an integer > 1): 0
```

```
Enter the root number (an integer > 1): 1
```

```
Enter the root number (an integer > 1): 1
```

```
Building the tree...
```

```
Tree: (left branch is blue, right branch is yellow)
```



```
Process finished with exit code 0
```