**ЛАБОРАТОРНА РОБОТА №14**

## Програмування задач з використанням дерев.

**Мета:**

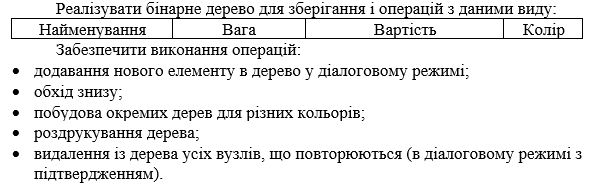
* Метою даної лабораторної роботи є закріплення та доповнення лекційного матеріалу, та набуття студентами практичних навичок вирішення основних завдань програмування задач на С++ з використанням динамічної структури - дерево.

**Оформити звіт:**

1. Тема.
2. Мета.
3. Постановка задачі.
4. Програма.
5. Результати роботи.
6. Висновки.

**Задача 1 завдання 9**

**Умови завдання:**

****

**3)Лістинг програми;**

#include <iostream>

#include <iomanip>

#include <string.h>

#include <Windows.h>

using namespace std;

struct Ware

{

string name\_tovar;

int weight;

int cost\_tovar;

string color;

Ware\* left = NULL;

Ware\* right = NULL;

Ware\* parent = NULL;

}; Ware\* koren = nullptr;

struct Color

{

string name\_tovar;

int weight;

int cost\_tovar;

string color;

Color\* left1 = NULL;

Color\* right1 = NULL;

Color\* parent1 = NULL;

}; Color\* koren1 = nullptr;

void Header();

void Menu();

void Input\_one\_value(Ware\* a0);

void display\_all(Ware\* koren);

void display\_level(Ware\* koren\_d, int& i, int level);

void LEVEL(Ware\* koren\_d, int& i, int& max\_level);

void postorderTraversal(Ware\* root);

void dfs(Ware\* koren);

void deleteTree\_base(Ware\* koren);

void deleteTree(Color\* koren);

void removeDuplicates(Ware\*& ware);

bool isDuplicate(Ware\* koren, string name, int P);

void input\_Ware(Ware\* koren);

void Add\_new\_Ware(Ware\* koren, Ware\* new\_code);

void insert\_Color(Color\*& root, string name\_tovar, int weight, int cost\_tovar, string color);

void Create\_Tree\_with\_color(Ware\* root, string cur\_color);

int main()

{

setlocale(LC\_ALL, "ukr");

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

int k;

Ware\* a0 = new Ware;

Input\_one\_value(a0);

bool t = true;

while (t)

{

Menu();

cout << "Виберiть режим роботи: " << endl;

k = 0;

cin >> k;

cin.ignore();

switch (k)

{

case 1:

{

Ware\* a1 = new Ware;

input\_Ware(koren);

Add\_new\_Ware(koren, a0);

cout << endl;

break;

}

case 2:

{

cout << "Обхід графа знизу: " << endl;

Header();

postorderTraversal(koren);

cout << endl;

break;

}

case 3:

{

string cur\_color;

cout << "Ведіть колір, який хочете додати: ";

cin >> cur\_color;

cout << "Дерево було сформовано:" << endl;

Header();

Create\_Tree\_with\_color(koren, cur\_color);

deleteTree(koren1);

cout << "Дерево з кольорами було видалено" << endl;

break;

}

case 4:

{

display\_all(koren); break;

}

case 5:

{

removeDuplicates(koren);

cout << "Дублікати були видалені" << endl;

break;

}

case 6:

{

t = 0;

break;

}

}

}

deleteTree\_base(koren);

cout << "Дерево було видалено" << endl;

}

void Create\_Tree\_with\_color(Ware\* root, string cur\_color)

{

int i = 0;

string name\_tovar, color;

int weight, cost\_tovar;

if (root == NULL)

{

return;

}

if (root->color == cur\_color)

{

Color\* new\_el = new Color;

name\_tovar = root->name\_tovar;

weight = root->weight;

cost\_tovar = root->cost\_tovar;

color = root->color;

insert\_Color(new\_el, name\_tovar, weight, cost\_tovar, color);

cout << root->name\_tovar << setw(8) << root->weight << setw(8) << root->cost\_tovar << setw(8) << root->color << endl;

}

Create\_Tree\_with\_color(root->left, cur\_color);

Create\_Tree\_with\_color(root->right, cur\_color);

}

void insert\_Color(Color\*& root, string name\_tovar, int weight, int cost\_tovar, string color)

{

if (root == nullptr)

{

root = new Color{ name\_tovar, weight, cost\_tovar, color };

koren1 = root;

return;

}

if (name\_tovar < root->name\_tovar)

{

if (root->left1 == nullptr)

{

root->left1 = new Color{ name\_tovar, weight, cost\_tovar, color };

root->left1->parent1 = root;

}

else

{

insert\_Color(root->left1, name\_tovar, weight, cost\_tovar, color);

}

}

else

{

if (root->right1 == nullptr)

{

root->right1 = new Color{ name\_tovar, weight, cost\_tovar, color };

root->right1->parent1 = root;

}

else

{

insert\_Color(root->right1, name\_tovar, weight, cost\_tovar, color);

}

}

}

void Add\_new\_Ware(Ware\* koren, Ware\* new\_code)

{

if (koren->cost\_tovar > new\_code->cost\_tovar && koren->left != NULL)

Add\_new\_Ware(koren->left, new\_code);

if (koren->cost\_tovar < new\_code->cost\_tovar && koren->right != NULL)

Add\_new\_Ware(koren->right, new\_code);

if (koren->cost\_tovar > new\_code->cost\_tovar && koren->left == NULL)

koren->left = new\_code;

if (koren->cost\_tovar < new\_code->cost\_tovar && koren->right == NULL)

koren->right = new\_code;

}

void input\_Ware(Ware\* koren)

{

Ware\* a1 = new Ware;

cout << "Введіть назву товару: ";

cin >> a1->name\_tovar;

cout << "Введіть вагу товару: ";

cin >> a1->weight;

cout << "Введіть вартість товару: ";

cin >> a1->cost\_tovar;

cout << "Введіть колір товару: ";

cin >> a1->color;

if (koren == NULL) { koren = a1; }

Add\_new\_Ware(koren, a1);

}

bool isDuplicate(Ware\* koren, string name, int P)

{

if (koren == nullptr)

{

return false;

}

if (koren->name\_tovar == name)

{

P++;

if (P > 1)

{

return true;

}

}

if (isDuplicate(koren->left, name, P))

{

return true;

}

if (isDuplicate(koren->right, name, P))

{

return true;

}

return false;

}

void removeDuplicates(Ware\*& ware)

{

if (ware == nullptr)

{

return;

}

if (isDuplicate(koren, ware->name\_tovar, 0))

{

cout << "Ware " << ware->name\_tovar << " is a duplicate. Delete? (y/n) ";

char answer;

cin >> answer;

if (answer == 'y')

{

ware = nullptr;

delete ware;

}

}

removeDuplicates(ware->left);

removeDuplicates(ware->right);

}

void deleteTree(Color\* koren)

{

if (koren == nullptr) {

return;

}

deleteTree(koren->left1);

deleteTree(koren->right1);

delete koren;

}

void deleteTree\_base(Ware\* koren)

{

if (koren == nullptr) {

return;

}

deleteTree\_base(koren->left);

deleteTree\_base(koren->right);

delete koren;

}

void dfs(Ware\* koren)

{

if (koren == NULL)

{

return;

}

dfs(koren->left);

dfs(koren->right);

}

void postorderTraversal(Ware\* root)

{

if (root == NULL)

return;

cout << root->name\_tovar << setw(8) << root->weight << setw(8) << root->cost\_tovar << setw(8) << root->color << endl;

postorderTraversal(root->left);

postorderTraversal(root->right);

}

void LEVEL(Ware\* koren\_d, int& i, int& max\_level)

{

if (koren\_d->left != NULL)

{

i++;

if (max\_level < i)

max\_level = i;

LEVEL(koren\_d->left, i, max\_level);

i--;

}

if (koren\_d->right != NULL)

{

i++;

if (max\_level < i)

max\_level = i;

LEVEL(koren\_d->right, i, max\_level);

i--;

}

}

void display\_level(Ware\* koren\_d, int& i, int level)

{

if (i == level)

{

cout << " ";

cout << koren\_d->name\_tovar << setw(10) << koren\_d->weight << setw(10) << koren\_d->cost\_tovar << setw(10) << koren\_d->color << endl;

}

if (koren\_d->left != NULL)

{

i++;

display\_level(koren\_d->left, i, level);

i--;

}

if (koren\_d->right != NULL)

{

i++;

display\_level(koren\_d->right, i, level);

i--;

}

}

void display\_all(Ware\* koren)

{

int i = 0;

int max\_level = 0;

LEVEL(koren, i, max\_level);

cout << "max рівень = " << max\_level << endl;

cout << " "; Header();

for (int level = 0; level <= max\_level; level++)

{

if (level == 0)

cout << "0 рівень: " << endl;

else

{

cout << "(" << level << " рівень): " << endl;

}

display\_level(koren, i, level);

}

cout << endl;

}

void Input\_one\_value(Ware\* a0)

{

cout << "Введіть назву товару: ";

cin >> a0->name\_tovar;

cout << "Введіть вагу товару: ";

cin >> a0->weight;

cout << "Введіть вартість товару: ";

cin >> a0->cost\_tovar;

cout << "Введіть колір товару: ";

cin >> a0->color;

koren = a0;

}

void Menu()

{

cout << "1 - додавання нового елементу в дерево у діалоговому режимі;" << endl;

cout << "2 - обхід знизу;" << endl;

cout << "3 - побудова окремих дерев для різних кольорів;" << endl;

cout << "4 - роздрукування дерева;" << endl;

cout << "5 - видалення із дерева усіх вузлів, що повторюються (в діалоговому режимі з підтвердженням)" << endl;

cout << "6 - вихiд;" << endl;

cout << endl;

}

void Header()

{

cout << "Назва товару" << setw(10) << "Вагa" << setw(10) << "Вартість" << setw(10) << "Колір" << endl;

}

**5)Результати**

Введіть назву товару: Монітор

Введіть вагу товару: 424

Введіть вартість товару: 2000

Введіть колір товару: Синій

1 - додавання нового елементу в дерево у діалоговому режимі;

2 - обхід знизу;

3 - побудова окремих дерев для різних кольорів;

4 - роздрукування дерева;

5 - видалення із дерева усіх вузлів, що повторюються (в діалоговому режимі з підтвердженням)

6 - вихiд;

Виберiть режим роботи:

1

Введіть назву товару: Піч

Введіть вагу товару: 748

Введіть вартість товару: 1900

Введіть колір товару: Чорний

1 - додавання нового елементу в дерево у діалоговому режимі;

2 - обхід знизу;

3 - побудова окремих дерев для різних кольорів;

4 - роздрукування дерева;

5 - видалення із дерева усіх вузлів, що повторюються (в діалоговому режимі з підтвердженням)

6 - вихiд;

Виберiть режим роботи:

1

Введіть назву товару: Мишка

Введіть вагу товару: 1990

Введіть вартість товару: 1990

Введіть колір товару: Червоний

1 - додавання нового елементу в дерево у діалоговому режимі;

2 - обхід знизу;

3 - побудова окремих дерев для різних кольорів;

4 - роздрукування дерева;

5 - видалення із дерева усіх вузлів, що повторюються (в діалоговому режимі з підтвердженням)

6 - вихiд;

Виберiть режим роботи:

1

Введіть назву товару: Процесор

Введіть вагу товару: 7283

Введіть вартість товару: 2200

Введіть колір товару: Чорний

1 - додавання нового елементу в дерево у діалоговому режимі;

2 - обхід знизу;

3 - побудова окремих дерев для різних кольорів;

4 - роздрукування дерева;

5 - видалення із дерева усіх вузлів, що повторюються (в діалоговому режимі з підтвердженням)

6 - вихiд;

Виберiть режим роботи:

1

Введіть назву товару: Процесор

Введіть вагу товару: 7823

Введіть вартість товару: 9823

Введіть колір товару: Чорний

1 - додавання нового елементу в дерево у діалоговому режимі;

2 - обхід знизу;

3 - побудова окремих дерев для різних кольорів;

4 - роздрукування дерева;

5 - видалення із дерева усіх вузлів, що повторюються (в діалоговому режимі з підтвердженням)

6 - вихiд;

Виберiть режим роботи:

1

Введіть назву товару: телефон

Введіть вагу товару: 932

Введіть вартість товару: 1950

Введіть колір товару: Червоний

1 - додавання нового елементу в дерево у діалоговому режимі;

2 - обхід знизу;

3 - побудова окремих дерев для різних кольорів;

4 - роздрукування дерева;

5 - видалення із дерева усіх вузлів, що повторюються (в діалоговому режимі з підтвердженням)

6 - вихiд;

Виберiть режим роботи:

2

Обхід графа знизу:

Назва товару Вагa Вартість Колір

Монітор 424 2000 Синій

Піч 748 1900 Чорний

Мишка 1990 1990Червоний

телефон 932 1950Червоний

Процесор 7283 2200 Чорний

Процесор 7823 9823 Чорний

1 - додавання нового елементу в дерево у діалоговому режимі;

2 - обхід знизу;

3 - побудова окремих дерев для різних кольорів;

4 - роздрукування дерева;

5 - видалення із дерева усіх вузлів, що повторюються (в діалоговому режимі з підтвердженням)

6 - вихiд;

Виберiть режим роботи:

4

max рівень = 3

Назва товару Вагa Вартість Колір

0 рівень:

Монітор 424 2000 Синій

(1 рівень):

Піч 748 1900 Чорний

Процесор 7283 2200 Чорний

(2 рівень):

Мишка 1990 1990 Червоний

Процесор 7823 9823 Чорний

(3 рівень):

телефон 932 1950 Червоний

1 - додавання нового елементу в дерево у діалоговому режимі;

2 - обхід знизу;

3 - побудова окремих дерев для різних кольорів;

4 - роздрукування дерева;

5 - видалення із дерева усіх вузлів, що повторюються (в діалоговому режимі з підтвердженням)

6 - вихiд;

Виберiть режим роботи:

3

Ведіть колір, який хочете додати: Чорний

Дерево було сформовано:

Назва товару Вагa Вартість Колір

Піч 748 1900 Чорний

Процесор 7283 2200 Чорний

Процесор 7823 9823 Чорний

Дерево з кольорами було видалено

1 - додавання нового елементу в дерево у діалоговому режимі;

2 - обхід знизу;

3 - побудова окремих дерев для різних кольорів;

4 - роздрукування дерева;

5 - видалення із дерева усіх вузлів, що повторюються (в діалоговому режимі з підтвердженням)

6 - вихiд;

Виберiть режим роботи:

5

Ware Процесор is a duplicate. Delete? (y/n) n

Ware Процесор is a duplicate. Delete? (y/n) n

Дублікати були видалені

1 - додавання нового елементу в дерево у діалоговому режимі;

2 - обхід знизу;

3 - побудова окремих дерев для різних кольорів;

4 - роздрукування дерева;

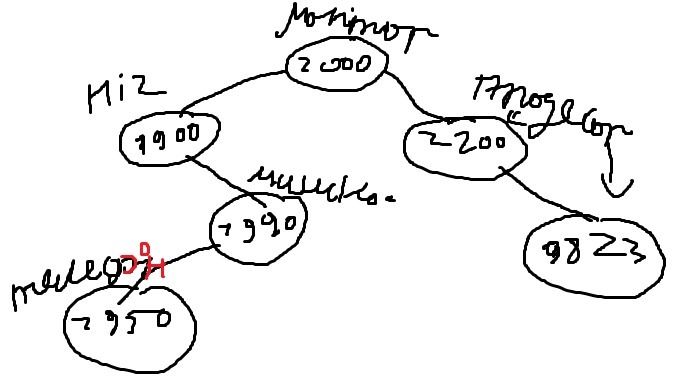
5 - видалення із дерева усіх вузлів, що повторюються (в діалоговому режимі з підтвердженням)

6 - вихiд;

Виберiть режим роботи:

6

Дерево було видалено

****

**Висновок:** Закріпив та доповнив лекційний матеріал, та набув практичних навичок вирішення основних завдань програмування задач на С++ з використанням динамічної структури - дерево.