Міністерство освіти і науки України

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»



Звіт

З лабораторної роботи № 6, варіант 9

З дисципліни « ПРОГРАМУВАННЯ, ЧАСТИНА 2 (ОБ’ЄКТНООРІЄНТОВАНЕ ПРОГРАМУВАННЯ) »

На тему: «Спадкування»

Виконав: ст. гр. КІ-15

Горбачук В.Р.

Перевірив: викладач

Козак Н.Б.

Львів – 2020

**Завдання.** Створити абстрактний базовий клас і похійдний від нього клас, які реалізують модель предметної області згідно варіанту. Кожен клас має мати мінімум 3 власні елементи даних один з яких створюється динамічно, методи встановлення і читання характеристик елементів-даних класу (Set і Get), та мінімум 2 абстрактні методи обробки даних і мінімум 2 методи обробки даних у похідному класі. Крім цього клас має містити перевантаження оператора присвоєння, конструкторів по замовчуванню і копіювання та віртуальний деструктор. Для розроблених класів реалізувати програму-драйвер, яка демонструє роботу класів.

9 Рослина Дерево

**Код програми:**

//Plant.cpp

#include "pch.h"

#include "Plant.h"

#include <iostream>

using namespace std;

Plant::Plant() { // Конструктор по замовчуванню

Name = "undefined";

population = 0;

lifeCycleAge = 0;

}

Plant::Plant(string name) { // Конструктор з одним параметром

this->Name = name;

population = 1;

lifeCycleAge = 0;

}

Plant::Plant(Plant &newPlant) { // Конструктор копіюваня

this->Name = newPlant.Name;

this->population = newPlant.population;

this->lifeCycleAge = newPlant.lifeCycleAge;

}

Plant::~Plant() {

// Деструктор не робить нічого так як немає динамічно виділених обєктів які потрібно видалити

}

void Plant::SetPopulation(int population) {

this->population = population;

}

void Plant::SetlifeCycleAge(int lifeCycleAge) {

this->lifeCycleAge = lifeCycleAge;

}

int Plant::getPopulation()

{

return this->population;

}

int Plant::getLifeCycleAge()

{

return this->lifeCycleAge;

}

void Plant::ShowInfo() {

cout << endl << ""

<< ">Назва рослини: " << Name << endl << "--------------------------" << endl << ">Популяцiя рослини: " << population <<

endl << ">Життєвий цикл рослини: " << lifeCycleAge << endl << "--------------------------" << endl;

}

//Tree.cpp

#include "pch.h"

#include "Tree.h"

#include <iostream>

Tree::Tree() : Plant() // Конструктор по замовчуванню

{

height = 0;

stemRadius = 0;

age = 0;

}

Tree::Tree(int height, int stemRadius, int age, string Name) : Plant(Name) // Конструктор з 3 параметрами

{

this->height = height;

this->stemRadius = stemRadius;

this->age = age;

}

Tree::Tree(Tree & newTree) : Plant(Name) // Конструктор копіювання

{

this->age = newTree.age;

this->height = newTree.height;

this->lifeCycleAge = newTree.lifeCycleAge;

this->population = newTree.population;

this->stemRadius = newTree.stemRadius;

}

Tree::~Tree()

{

// деструктор не робить нічого так як немає динамічно виділеної памяті яку потрібно вивільняти

}

void Tree::setHeight(int height)

{

this->height = height;

}

void Tree::setStemRadius(int stemRadius)

{

this->stemRadius = stemRadius;

}

void Tree::setAge(int age)

{

this->age = age;

}

void Tree::FindTimeToDying()

{

cout << endl << ">Час до вмирання дерева: " << lifeCycleAge - age << endl;

}

void Tree::GetTreeVolume()

{

cout << endl << ">Радiус дерева: " << (double)stemRadius\*height\*3.14\*3.14 << endl;

cout << endl << "--------------------------" << endl;

}

//main.cpp

#include "pch.h"

#include <iostream>

#include "Tree.h"

using namespace std;

int main()

{

setlocale(LC\_CTYPE, "ukr");

cout << endl << "" << endl;

cout << " Показ програми";

cout << endl << "--------------------------" << endl;

Tree tree(25, 6, 4, "Липа"); // Використання конструктора

tree.SetlifeCycleAge(6); // Встановлення lifeCycleAge

tree.ShowInfo(); // Показати інформацію

tree.FindTimeToDying();

tree.GetTreeVolume();

system("pause");

return 0;

}

Вікно програми:

