Міністерство освіти і науки України

Центральноукраїнський національний технічний університет

Механіко-технологічний факультет

Кафедра кібербезпеки та програмного забезпечення

**ЛАБОРАТОРНА РОБОТА №3**

**Тема:** Успадкування, агрегація та композиція

з дисципліни: Об’єктно-орієнтоване програмування

|  |
| --- |
| Виконав: ст. гр. КН-23 |
| Царенко С. О. |
| Перевірив: асистент  Козірова Н. Л. |
|  |

Кропивницький

2024

**Тема:** Успадкування, агрегація та композиція

**Мета:** ознайомитись з основними поняттями успадкування, агрегація та композиція в ООП та навчитись їх програмно реалізовувати мовою С++.

**Варіант 5**

**Завдання 1**

Створіть використовуючи композицію два класи, які матимуть свої властивості, перший реалізуйте за допомогою включення (composition), а другий за допомогою посилання (reference), контейнером для цих класів використайте клас з лабораторної роботи №1, створіть також третій – похідний клас, використовуючи наслідування, базовим класом може бути або клас з лр№1, або класи які ви створили за допомогою композиції. За бажанням можете зобразити схематично як ваші класи залежать один від одного.

**Завдання 2**

1. Створіть клас "Телефон", який містить такі властивості:

• Марка телефону

• Модель телефону

• Екран (використовуйте композицію)

• Камера (використовуйте композицію)

1. Створіть клас "Екран", який містить такі властивості:

• Розмір екрану

• Тип екрану (LCD, OLED і т.д.)

1. Створіть клас "Камера", який містить такі властивості:

• Розмір матриці камери

• Роздільна здатність камери

4. Реалізуйте методи для встановлення та отримання інформації про телефон, екран та камеру.

5. Напишіть демонстраційну функцію main(), в якій створюються об'єкти класів "Телефон", "Екран" та "Камера". Встановіть значення для властивостей та виведіть інформацію про телефон.

6. Продемонструйте успадкування, створивши похідний клас від "Телефон" (наприклад, "Смартфон") з додатковими властивостями та методами.

7. Розширте функціональність, додавши додаткові методи та властивості до класів за власним бажанням. Не забудьте додати коментарі та пояснення до коду, щоб роз'яснити його функціональність та зрозумілість.

**Завдання 1.**

***Student.h:***

#pragma once

#include<string>

#include<iostream>

using namespace std;

class Student

{

private:

string name;

int age;

string major;

public:

void SetName(string newName);

void SetAge(int newAge);

void SetMajor(string newMajor);

string GetName();

int GetAge();

string GetMajor();

};

***Student.cpp:***

#include "Student.h"

using namespace std;

void Student::SetName(string newName)

{

this->name = newName;

}

void Student::SetAge(int newAge)

{

this->age = newAge;

}

void Student::SetMajor(string newMajor)

{

this->major = newMajor;

}

string Student::GetName()

{

return name;

}

int Student::GetAge()

{

return age;

}

string Student::GetMajor()

{

return major;

}

***Address.h:***

#pragma once

#include <string>

using namespace std;

class Address {

private:

string street;

string city;

public:

Address();

Address(string street, string city);

void SetStreet(string street);

void SetCity(string city);

string GetStreet();

string GetCity();

};

***Address.cpp:***

#include "Address.h"

Address::Address()

{

street = "Yanovskoho";

city = "Kropyvnytskyi";

}

Address::Address(string street, string city)

{

this->street = street;

this->city = city;

}

void Address::SetCity(string city) {

this->city = city;

}

void Address::SetStreet(string street) {

this->street = street;

}

string Address::GetCity() {

return city;

}

string Address::GetStreet() {

return street;

}

***University.h:***

#pragma once

#include <string>

using namespace std;

class University {

private:

string name;

public:

University();

University(string name);

string GetName();

void SetName(string name);

};

***University.cpp:***

#include "University.h"

University::University()

{

name = "CNTU";

}

University::University(string name)

{

this->name = name;

}

void University::SetName(string name) {

this->name = name;

}

string University::GetName() {

return name;

}

***Details.h:***

#pragma once

#include "Student.h"

#include "Address.h"

#include "University.h"

using namespace std;

class Details : public Student {

private:

Address address;

University& university;

public:

Details();

Details(Address address, University& university);

void SetAddress(Address address);

Address GetAddress();

University& GetUniversity();

};

***Details.cpp:***

#include "Details.h"

Details::Details(Address address, University& university) : address(address), university(university) {}

void Details::SetAddress(Address address) {

this->address = address;

}

Address Details::GetAddress() {

return address;

}

University& Details::GetUniversity() {

return university;

}

***Task1.cpp:***

#include <iostream>

#include "Details.h"

#include "Address.h"

#include "University.h"

using namespace std;

int main() {

string name, major, street, city, universityName;

int age;

cout << "Enter student name: ";

cin >> name;

cout << "Enter student age: ";

cin >> age;

cout << "Enter student major: ";

cin >> major;

cout << "Enter address (street): ";

cin >> street;

cout << "Enter address (city): ";

cin >> city;

cout << "Enter university name: ";

cin >> universityName;

Address address(street, city);

University university(universityName);

Details details(address, university);

details.SetName(name);

details.SetAge(age);

details.SetMajor(major);

cout << "\nStudent Details:" << endl;

cout << "Name: " << details.GetName() << endl;

cout << "Age: " << details.GetAge() << endl;

cout << "Major: " << details.GetMajor() << endl;

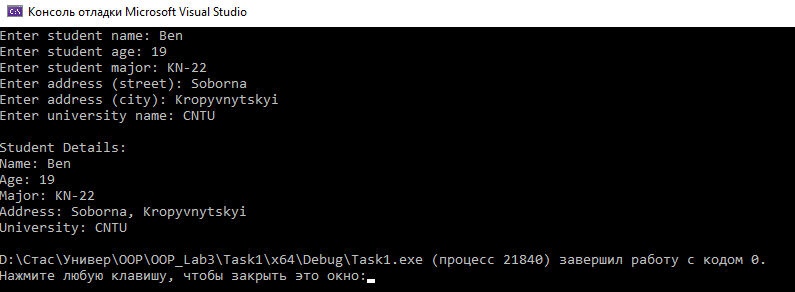
cout << "Address: " << details.GetAddress().GetStreet() << ", " << details.GetAddress().GetCity() << endl;

cout << "University: " << details.GetUniversity().GetName() << endl;

return 0;

}

***Скріншот:***

****

**Завдання 2.**

***Screen.h:***

#pragma once

#include<iostream>

#include<string>

using namespace std;

class Screen

{

private:

int size;

string type;

public:

Screen();

Screen(int size, string type);

void SetScreen(int size, string type);

int GetSize();

string GetType();

};

***Screen.cpp:***

#include "Screen.h"

Screen::Screen() {

size = 4;

type = "LCD";

}

Screen::Screen(int size, string type) {

this->size = size;

this->type = type;

}

void Screen::SetScreen(int size, string type) {

this->size = size;

this->type = type;

}

int Screen::GetSize() {

return size;

}

string Screen::GetType() {

return type;

}

***Camera.h:***

#pragma once

#include<iostream>

using namespace std;

class Camera

{

private:

float size;

int resolution;

public:

Camera();

Camera(float size, int resolution);

void SetCamera(float size, int resolution);

float GetSize();

int GetResolution();

};

***Camera.cpp:***

#include "Camera.h"

Camera::Camera() {

size = 0.5;

resolution = 12;

}

Camera::Camera(float size, int resolution) {

this->size = size;

this->resolution = resolution;

}

void Camera::SetCamera(float size, int resolution) {

this->size = size;

this->resolution = resolution;

}

float Camera::GetSize() {

return size;

}

int Camera::GetResolution() {

return resolution;

}

***Telephone.h:***

#pragma once

#include "Camera.h"

#include "Screen.h"

#include<iostream>

#include<string>

using namespace std;

class Telephone

{

private:

string brand;

string model;

Camera camera;

Screen screen;

public:

Telephone();

Telephone(string band, string model, Screen screen, Camera camera);

void SetBrand(string brand);

void SetModel(string model);

void SetTelephoneCamera(float size, int resolution);

void SetTelephoneScreen(int size, string type);

string GetBrand();

string GetModel();

Camera GetCamera();

Screen GetScreen();

};

***Telephone.cpp:***

#include "Telephone.h"

Telephone::Telephone() {

brand = "Apple";

model = "13 pro max";

camera = Camera();

screen = Screen();

}

Telephone::Telephone(string brand, string model, Screen screen, Camera camera) {

this->brand=brand;

this->model=model;

this->screen = screen;

this->camera = camera;

}

void Telephone::SetBrand(string brand) {

this->brand = brand;

}

void Telephone::SetModel(string model) {

this->model = model;

}

void Telephone::SetTelephoneCamera(float size, int resolution) {

camera.SetCamera(size,resolution);

}

void Telephone::SetTelephoneScreen(int size, string type) {

screen.SetScreen(size,type);

}

string Telephone::GetBrand() {

return brand;

}

string Telephone::GetModel() {

return model;

}

Camera Telephone::GetCamera() {

return camera;

}

Screen Telephone::GetScreen() {

return screen;

}

***Smartphone.h:***

#pragma once

#include"Telephone.h"

using namespace std;

class Smartphone : public Telephone

{

private:

int memory;

int RAM;

public:

Smartphone();

Smartphone(string brand, string model, Screen screen, Camera camera, int memory, int RAM);

void SetMemory(int memory);

void SetRam(int RAM);

void ShowInfo();

int GetMemory();

int GetRam();

};

***Smartphone.cpp:***

#include "Smartphone.h"

#include <iostream>

Smartphone::Smartphone() : Telephone() {

memory = 128;

RAM = 6;

}

Smartphone::Smartphone(string brand, string model, Screen screen, Camera camera, int memory, int RAM) : Telephone(brand, model, screen, camera) {

this->memory = memory;

this->RAM = RAM;

}

void Smartphone::SetMemory(int memory) {

this->memory = memory;

}

void Smartphone::SetRam(int RAM) {

this->RAM = RAM;

}

int Smartphone::GetMemory() {

return memory;

}

int Smartphone::GetRam() {

return RAM;

}

void Smartphone::ShowInfo() {

cout << "Smartphone info: " << endl;

cout << "Brand: " << GetBrand() << endl;

cout << "Model: " << GetModel() << endl;

cout << "Screen size: " << GetScreen().GetSize() << endl;

cout << "Screen type: " << GetScreen().GetType() << endl;

cout << "Camera size: " << GetCamera().GetSize() << endl;

cout << "Camera resolution: " << GetCamera().GetResolution() << endl;

cout << "Memory: " << GetMemory() << " Gb" << endl;

cout << "RAM: " << GetRam() << " Gb" << endl;

}

***Task2.cpp:***

#include<iostream>

#include "Telephone.h"

#include "Smartphone.h"

using namespace std;

int main() {

Telephone telephone;

cout << "Telephone info: " << endl;

cout << "Brand: " << telephone.GetBrand() << endl;

cout << "Model: " << telephone.GetModel() << endl;

cout << "Screen size: " << telephone.GetScreen().GetSize() << endl;

cout << "Screen type: " << telephone.GetScreen().GetType() << endl;

cout << "Camera size: " << telephone.GetCamera().GetSize() << endl;

cout << "Camera resolution: " << telephone.GetCamera().GetResolution() << endl;

Screen scr2(5, "LED");

Camera cam2(1.2, 14);

Telephone tel2("Samsung", "Galaxy", scr2, cam2);

cout << "Telephone info: " << endl;

cout << "Brand: " << tel2.GetBrand() << endl;

cout << "Model: " << tel2.GetModel() << endl;

cout << "Screen size: " << tel2.GetScreen().GetSize() << endl;

cout << "Screen type: " << tel2.GetScreen().GetType() << endl;

cout << "Camera size: " << tel2.GetCamera().GetSize() << endl;

cout << "Camera resolution: " << tel2.GetCamera().GetResolution() << endl;

Screen smartscr(6, "OLED");

Camera smartcam(1.6, 24);

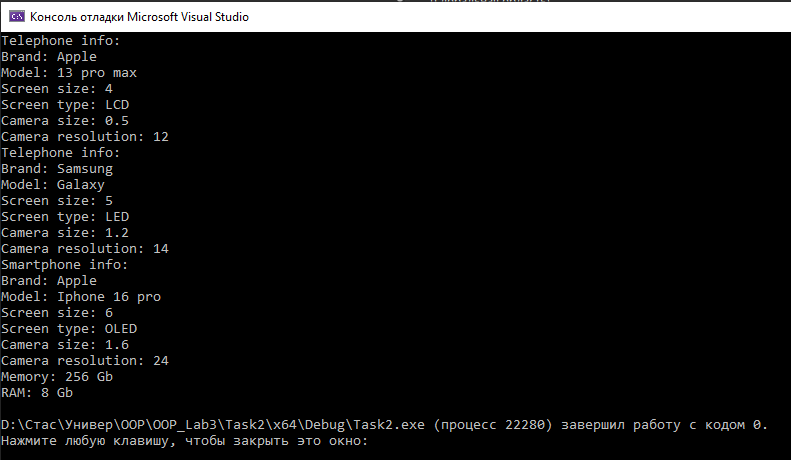
Smartphone newSmartphone("Apple", "Iphone 16 pro", smartscr, smartcam, 256, 8);

newSmartphone.ShowInfo();

return 0;

}

***Скріншот:***

****

**Висновок:**

У ході виконання лабораторної роботи №3 “ Успадкування, агрегація та композиція ”. Я ознайомився з основними поняттями успадкування, агрегація та композиція в ООП та навчився їх програмно реалізовувати мовою С++.