Міністерство освіти і науки України

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»



Звіт

З лабораторної роботи № 6, варіант 23

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

На тему: «Спадкування»

Виконав: ст. гр. КІ-15

Пилипів Р. С.

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.

|  |  |  |
| --- | --- | --- |
| Варіант | Завдання | |
| Абстрактний базовий клас | Похідний клас |
| 23 | Мікросхема | Процесор |

**Код програми:**

**//**microcircuit.h

#include <stdlib.h>

#include <string.h>

#include <iostream>

using namespace std;

class Micro

{

public:

Micro(char\* fName);

virtual ~Micro();

virtual bool On() = 0;

virtual bool Off() = 0;

virtual bool Execute(char\* cmd, void\* prm) = 0;

virtual bool Status(int ext = 0)

{

return m\_bIsOned;

}

protected:

char\* m\_pszMicroName;

char\* m\_pszFriendlyName;

bool m\_bIsOned;

};

**//**microcircuit.cpp

#include "microcircuit.h"

Micro::Micro(char\* fName)

{

m\_pszFriendlyName = new char[strlen(fName) + 1];

m\_pszMicroName = 0;

}

Micro::~Micro()

{

if (m\_pszMicroName != 0)

{

delete[] m\_pszMicroName;

}

if (m\_pszFriendlyName != 0)

{

delete[] m\_pszFriendlyName;

}

}

//processor.h

#include <stdlib.h>

#include <string.h>

#include "microcircuit.h"

#include <iostream>

using namespace std;

class Processor : public Micro

{

public:

Processor(char\* fName);

virtual ~Processor();

virtual bool On();

virtual bool Off();

virtual bool Execute(char\* cmd, void\* prm);

private:

char\* m\_pszScrBuf;

};

//processor.cpp

#include "microcircuit.h"

Micro::Micro(char\* fName)

{

m\_pszFriendlyName = new char[strlen(fName) + 1];

m\_pszMicroName = 0;

}

Micro::~Micro()

{

if (m\_pszMicroName != 0)

{

delete[] m\_pszMicroName;

}

if (m\_pszFriendlyName != 0)

{

delete[] m\_pszFriendlyName;

}

}

//main.cpp

#include "processor.h"

#include <iostream>

using namespace std;

int main(void)

{

char w[128] = { "processor" };

char v[128] = { "read" };

char b[128] = { "clear" };

char s[128] = { "scan" };

char\* strinf = new char[128];

Processor processor(w);

cout << "Enter processor's model: ";

processor.On();

cout << processor.Status() << endl;

processor.Execute(s, strinf);

cout << strinf << endl;

processor.Execute(v, strinf);

cout << strinf << endl;

processor.Execute(b, strinf);

processor.Execute(v, strinf);

cout << strinf << endl;

processor.Off();

cout << processor.Status() << endl;

}

