Приклад. Простий приклад використання інтерфейсу

// ConsoleApplication100.cpp : Defines the entry point for the console application.

//

#include "stdafx.h"

#include<iostream>

using namespace std;

//--- ОПИС ІНТЕРФЕЙСУ

\_\_interface IMyInterface

{

int method1();

void method2();

};

//-- ПРИКЛАД КЛАСУ, ЩО РЕАЛІЗОВУЄ ІНТЕРФЕЙС

class MyClass :public IMyInterface

{

public:

int method1();

void method2();

};

//--- ЩЕ ОДИН ПРИКЛАД КЛАСУ, ЩО РЕАЛІЗОВУЄ ІНТЕРФЕЙС

class MyClass2 :public IMyInterface

{

public:

int method1();

void method2();

};

//------------------------

// -- фунція, у яку можна передати адресу будь-якого класу,

// який реалізовує інтерфейс IMyInterface

void showMessage(IMyInterface\* m)

{

m->method2();

}

int main()

{

MyClass c1;

showMessage(&c1);

showMessage( new MyClass2());

system("pause");

return 0;

}

int MyClass::method1()

{

return 78;

}

void MyClass::method2()

{

cout << "hello" << endl;

}

int MyClass2::method1()

{

return 22;

}

void MyClass2::method2()

{

cout << "Hi" << endl;

}

//=============================

ПРИКЛАД. Розробити клас, що представляє працівника і реалізовує інтерфейс IComparable з методом compareTo для порівняння двох працівників.

//-- Опис інтерфейса

#pragma once

\_\_interface IComparable

{

int compareTo(void\* obj);

};

//---------------------

// Опис класу, що реалізовує інтерфейс

#pragma once

#include "IComparable.h"

#include<string>

using namespace std;

class Employee :

public IComparable

{

public:

int ID;

string Name;

double Salary;

Employee(int id,string name, double salary);

Employee();

int compareTo(void\* obj); //-- метод, що реалізовує

};

//------------------------------

#include "stdafx.h"

#include "Employee.h"

Employee::Employee(int id, string name, double salary)

{

ID = id;

Name = name;

Salary = salary;

}

Employee::Employee():Employee(0,"no name",0)

{

}

int Employee::compareTo(void \* obj)

{

Employee\* otherEmployee = (Employee\*)obj;

if (Salary>otherEmployee->Salary)

{

return 1;

}

if (Salary<otherEmployee->Salary)

{

return -1;

}

return 0;

}

=======================================================================================

=======================================================================================

=======================================================================================

=======================================================================================

=======================================================================================

=======================================================================================

//----------------------

// ConsoleApplication100.cpp : Defines the entry point for the console application.

//

#include "stdafx.h"

#include<iostream>

#include "IComparable.h"

#include "Employee.h"

using namespace std;

\_\_interface IMoveAble

{

void move();

};

\_\_interface IShootAble

{

int shoot(int shootPosition);

};

\_\_interface \_\_ICreature :public IMoveAble, public IShootAble

{

};

//--------------------------

//-- Функція, у яку для сортування можна передати масив покажчиків довільних класів, які

//-- реалізовують інтерфейс IComparable

void sort(IComparable\*\* dataArray,int length)

{

bool zamina;

do

{

zamina = false;

for (int i = 1; i < length; i++)

{

if (dataArray[i-1]->compareTo(dataArray[i])>0)

{

//swap(dataArray[i - 1], dataArray[i]);

IComparable\* temp = dataArray[i - 1];

dataArray[i - 1] = dataArray[i];

dataArray[i] = temp;

zamina = true;

}

}

} while (zamina);

}

//--------------------------

int main()

{

Employee\*\* group = new Employee\*[4];

group[0] = new Employee(1,"Ivan",3200);

group[1] = new Employee(3, "Petro", 10200);

group[2] = new Employee(2, "Stepan", 1200);

group[3] = new Employee(67, "Dmutro", 8200);

sort((IComparable\*\*)group, 4);

for (int i = 0; i < 4; i++)

{

cout << group[i]->Name.data() << "-" << group[i]->Salary << endl;

}

system("pause");

return 0;

}