ПРИКЛАД ІНТЕРФЕЙСАМИ І ШАБЛОНАМИ

#pragma once

template<class MyType>

\_\_interface IComparable {

int compareTo(MyType\* obj);

};

#pragma once

#include "IComparable.h"

#include <string>

using namespace std;

class Book:public IComparable<Book>

{

public:

string Author;

string Publish;

int Year;

Book(string author, string publish, int year) :Author(author), Publish(publish), Year(year)

{}

int compareTo(Book\* obj) //Порівнюємо за роком видання

{

return Year - obj->Year;

}

string toString() { return Author + ", " + Publish + "," + to\_string(Year); }

};

#pragma once

#include "IComparable.h"

#include <string>

using namespace std;

class Product: public IComparable<Product>

{

public:

string Title;

double Price;

int Count;

Product(string title, double price, int count) :Title(title), Price(price), Count(count)

{}

int compareTo(Product\* obj)

{

if (Price == obj->Price)

{

return 0;

}

if (Price > obj->Price)

{

return 1;

}

return -1;

}

string toString() { return Title + ", " + to\_string(Price) + "-" + to\_string(Count); }

};

// ConsoleApplication42.cpp : Defines the entry point for the console application.

//

#include "stdafx.h"

#include <iostream>

using namespace std;

#include "Product.h"

#include "IComparable.h"

#include "Book.h"

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

template<class AnyType>

IComparable<AnyType> \* getMaxItem(IComparable<AnyType> \* \* items, int itemsCount)

{

IComparable\* max = items[0];

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

{

if (max->compareTo(items[i])<0)

{

max = items[i];

}

}

return max;

}

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

int main()

{

Book\*\* books = new Book\*[3];

books[0] = new Book("A1", "V1", 1991);

books[1] = new Book("A2", "V2", 1998);

books[2] = new Book("A3", "V3", 1990);

Book\* maxBook = (Book\*)getMaxItem<Book>((IComparable<Book>\*\*) books, 3);

printf("Max= %s\n", maxBook->toString().data());

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

Product\*\* products = new Product\*[4];

products[0] = new Product("T1", 11, 34);

products[1] = new Product("T2", 22, 12);

products[2] = new Product("T3", 89, 84);

products[3] = new Product("T1", 78, 120);

Product\* maxProduct = (Product\*)(getMaxItem<Product>((IComparable<Product>\*\*)products, 4));

printf("Max product= %s\n", maxProduct->toString().data());

system("pause");

return 0;

}