МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

**Український державний університет науки і технологій**

Кафедра «Комп’ютерні інформаційні технології»

**Лабораторна робота №4**

**з дисципліни «Операції перетворення»**

**на тему: «Створення та використання простих класів. Перевантаження операцій»**

Виконал: студент гр. ПЗ2112

Субботін С.П.

Прийняла:   
Демидович І.М.

Дніпро, 2022

**Тема.** Операції перетворення

**Мета роботи**. Отримати практичні навички створення та використання простих класів та перевантаження операцій перетворення.

**Постановка завдання:** Варіант 6





Написати програму, що реалізує і демонструє роботу операцій перетворення типів.

**Текст програми:**

**1)**

#pragma once

#include <iostream>

#include <string>

#include <Windows.h>

using namespace std;

struct part {

int cost = 0;

string name = "default";

};

class furniture

{

private:

part data[5];

int value;

public:

furniture()

{

part nft[] = { {3999,"Шафа"}, {4599,"Диван"}, {1999,"Тумба"}, {2399,"Люстра"}, {15999,"Глеб"} };

for (int i = 0; i < 5; i++) {

data[i] = nft[i];

}

cout << "Конструктор без параметров\n";

}

furniture(furniture& c)

{

for (int i = 0; i < 5; i++) {

data[i] = c.data[i];

}

cout << "Конструктор копирования\n";

}

furniture(int a)

{

part nft[] = { {3999,"Шафа"}, {4599,"Диван"}, {1999,"Тумба"}, {2399,"Люстра"}, {15999,"Глеб"} };

for (int i = 0; i < 5; i++) {

data[i].name = nft[i].name;

data[i].cost = nft[i].cost \* a;

}

cout << "Конструктор преобразования\n";

}

friend part getData(int i, furniture& r);

friend void putIn(int i, furniture& r);

furniture operator ++(int z) {

furniture old = \*this;

for (int round = 0; round < 5; round++) {

data[round].cost++;

}

return old;

}

operator char\* ()

{

char\* str\_value = new char[10];

snprintf(str\_value, 10, "%d", value);

return str\_value;

}

operator int()

{

return value;

}

operator double()

{

return value;

}

furniture operator+() {

cout << " Назва Ціна" << endl;

for (int j = 0; j < 5; j++) {

cout << " " << data[j].name << " " << data[j].cost << endl; // to do

}

return \*this;

}

furniture operator --(int z) {

furniture old = \*this;

for (int round = 0; round < 5; round++) {

data[round].cost--;

}

return old;

}

furniture operator !=(string z) {

furniture old = \*this;

cout << " Назва Ціна" << endl;

for (int round = 0; round < 5; round++) {

if (data[round].name != z) {

cout << " " << data[round].name << " " << data[round].cost << endl; // to do

}

}

return old;

}

~furniture() {}

};

part getData(int i, furniture& r) {

return r.data[i];

}

void putIn(int i, furniture& r) {

r.value = i;

}

**2)**

#include "data.h"

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

int i = 1;

char swtch;

cout << "1 - конструктор копіювання." << endl;

cout << "2 - конструктор з параметрами." << endl;

cout << "Натисніть будь-яку кнопку для того щоб обрати конструктор за замовченням." << endl;

cout << "Оберіть параметр запуску: ";

cin >> swtch;

cout << endl;

furniture a1;

switch (swtch) {

case '1': {

furniture a2;

furniture a1 = a2;

break; };

case '2': {

furniture a1(4);

break; };

default: {

furniture a1;

};

}

system("cls");

while (i != 0) {

cout << "1 - Вивести на єкран " << endl;

cout << "2 - збільшення ціни на задану величину" << endl;

cout << "3 - не дорівнює за назвами" << endl;

cout << "4 - зменшення на задану величину" << endl;

cout << "5 - змінити тип данних" << endl;

cout << "6 - Вийти з програми" << endl;

cout << "Виберіть варіант: ";

cin >> swtch;

cout << endl;

switch (swtch) {

case '1': {

+a1;

system("pause");

}

break;

case '2': {

int grade;

cout << "Введіть величину на яку хочете збільшити ціну: ";

cin >> grade;

cout << endl;

for (int h = 0; h < grade; h++) {

a1++;

}

system("cls");

cout << "Величина була збільшена!" << endl;

system("pause");

}

break;

case '3': {

string OG;

cout << "Введіть назву: ";

cin >> OG;

cout << endl;

a1 != OG;

system("pause");

}

break;

case '4': {

int grade;

cout << "Введіть величину на яку хочете зменшити ціну: ";

cin >> grade;

cout << endl;

for (int h = 0; h < grade; h++) {

a1--;

}

system("cls");

cout << "Величина була зменшена!" << endl;

system("pause");

}

break;

case '6': {

cout << "Ви точно бажаєте вийти? (y/n) ";

cin >> swtch;

if ((swtch == 'y') || (swtch == 'Y')) {

i = 0;

}

cout << endl;

system("pause");

}

break;

case '5': {

bool ground = false;

string toFind;

part zxc;

do {

system("cls");

+a1;

cout << "Виберіть назву числа, яке потрібно перетворити: ";

cin >> toFind;

cout << endl;

for (int l = 0; l < 5; l++) {

zxc = getData(l, a1);

if (toFind == zxc.name) {

ground = true;

putIn(zxc.cost, a1);

double d = a1; //преобразование к типу double

char\* str = a1; //преобразование к типу char\*

int i = a1; //преобразование к типу int

cout << "as double: " << d << endl;

cout << "as string: " << str << endl;

cout << "as int: " << i << endl;

}

}

if (ground == false) {

cout << "Такої назви не існує! " << endl;

system("pause");

}

} while (ground == false);

system("pause");

}

break;

default: {

cout << "Неправильно обрана опція!" << endl;

}

}

system("cls");

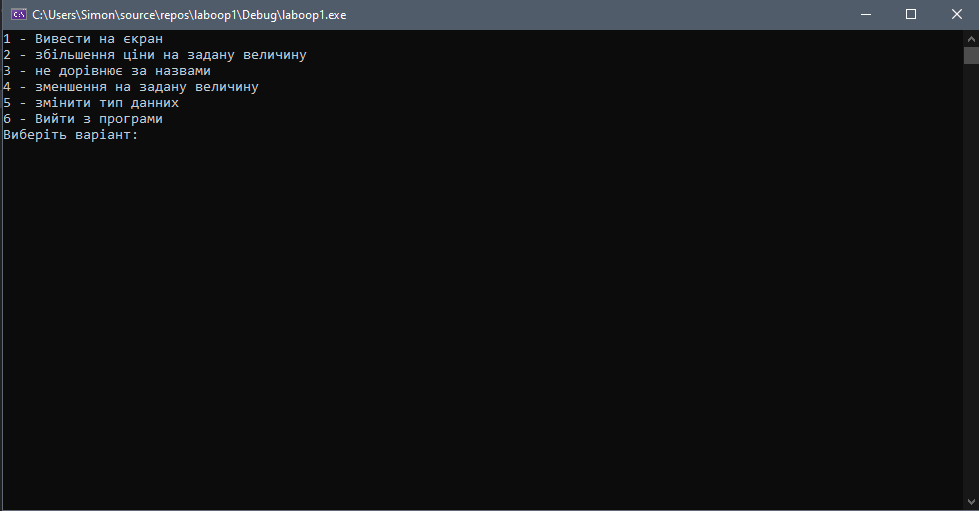
}

return 1;

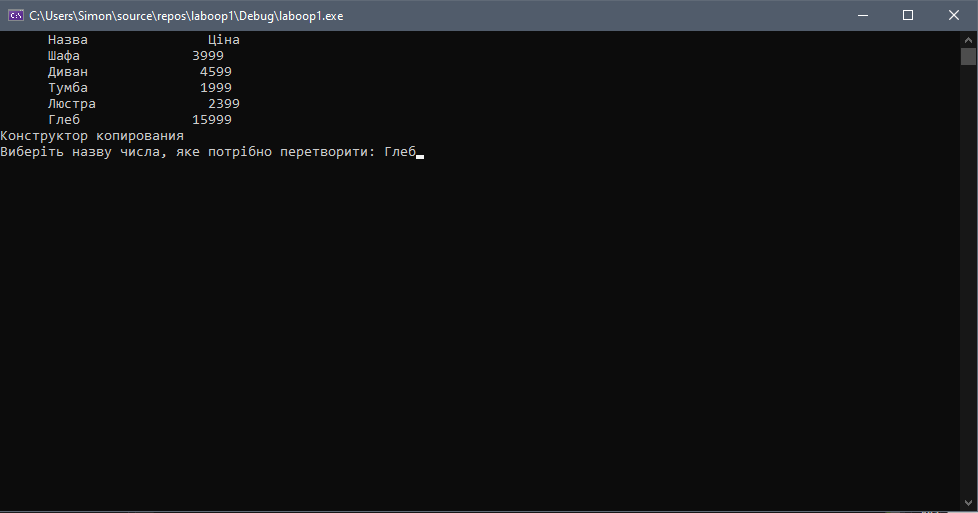
}

**Скріншоти тестів:**

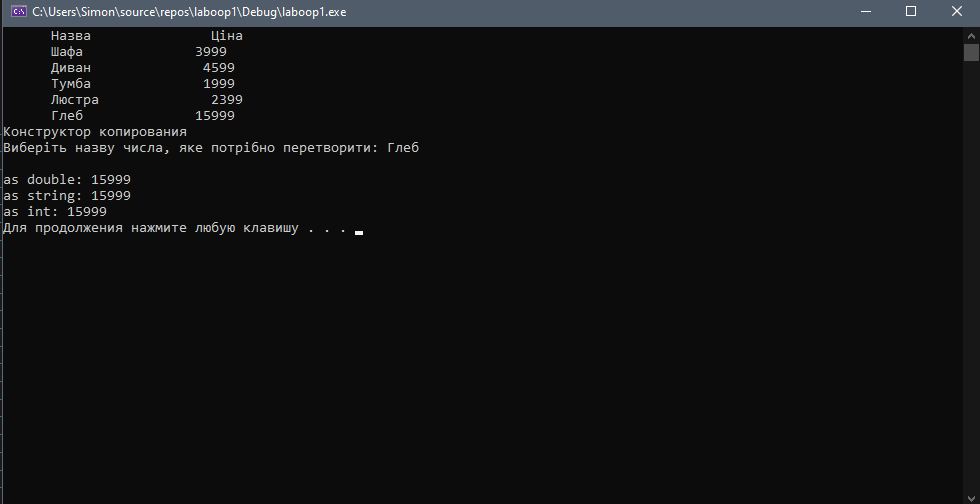
Принцип роботи програми аналогічний програмі з лабораторної роботи 2(див. Субботін Семен Лабораторна робота 2) за вийнятком нової функції – змінити тип даних.



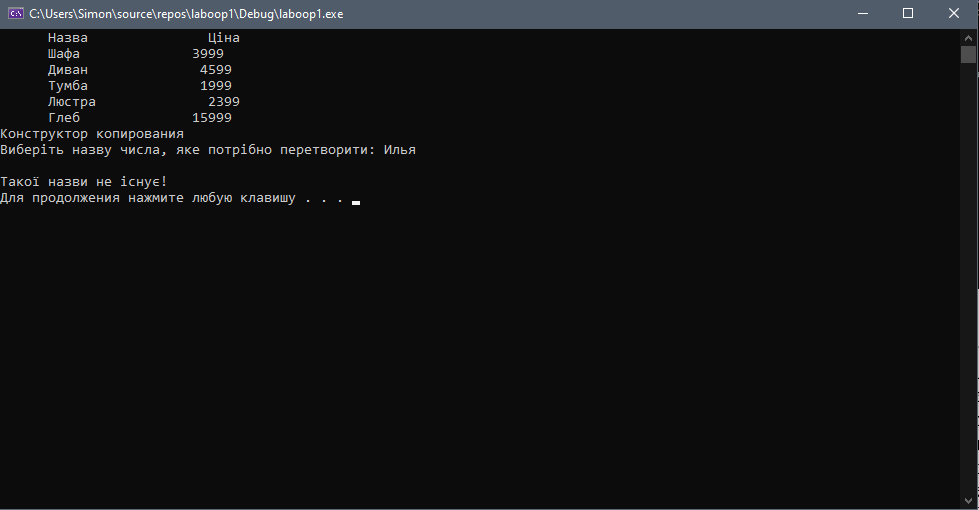
При виборі перетворення числа програма дає обрати число з заданих на вибір. Зробити це продукт пропонує вводом назви предмету

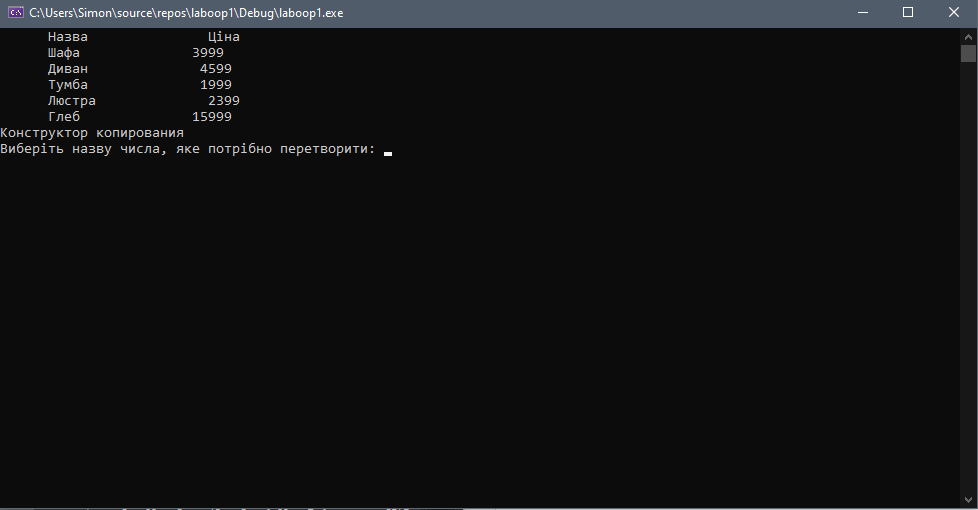


Після вибору предмету опрацьовується ціна, перетворюється до різних типів та виводиться на екран.



Якщо ввести неправильну назву, тоді програма запропонує спробувати ще раз.





**Висновок:** отримали практичні навички розробки програм за методом  
використання простих класів, деструкторів та конструкторів. Також опрацювали вміння правильно перевантажувати функції. При використанні простих класів – реалізація, зручність та швидкість написання продукту зростає, так як ми розбиваємо рішення завдання на класи, які пишуться окремо та які можна писати різним людям одночасно, що є набагато зручнішим за написання коду в одній функції. Також зростає безпечність продукту завдяки запобіганню витоку даних використовуючи деструктор. За допомогою перевантаження спростили виконання операцій з класами. Використання перегрузки – набагато зручніше, ніж використання звичних нам функцій. Навчились використовувати статичні об’єкти, змінні та функції. Це зручно для глобальних змінних одного типу класу. Перетворили змінну типу int у тип double та char всередині класу.