**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**Технічний коледж**

**НУ «Львівська політехніка»**

**Відділення**

**Інформаційних технологій**

**та комп’ютерної техніки**

**Звіт до:**

Лабораторної роботи № 1

з предмету «НП з ООП»

**Тема:**

«**Створення класу C++**»

**Підготував**

Студент групи – 31-ПЗ

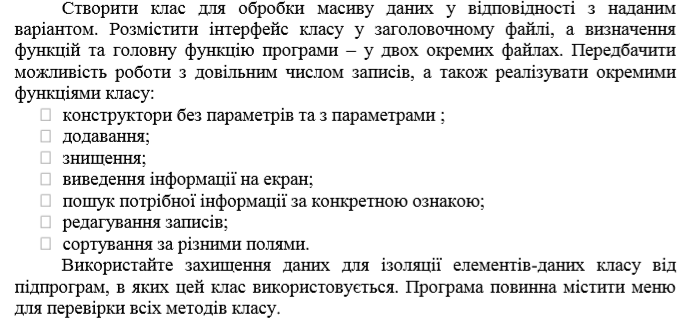
Дячок Остап

**Львів 2021**

**Мета роботи:** порівняння об’єктно-орієнтованого та функціонального підходів; початкове знайомство з класами, об’єктами та головними елементами об’єктного підходу.

**Постановка задачі:**

**Варіант 8**





**Код :**

[main.cpp](https://www.onlinegdb.com/online_c++_compiler#editor_1) :

#include "storage.h"

#include <iostream>

#include <string>

using namespace std;

int

main ()

{

int notes;

string method;

cout << " Enter the number of entries:" << endl;

cout << ">>";

cin >> notes;

storage variant8(true);

storage \*allNotes = new storage[notes];

for (;;)

{

cout << ">>";

cin >> method;

if (method == "add")

variant8.add (allNotes, notes);

if (method == "edit")

variant8.edit (allNotes);

if (method == "search")

variant8.search (allNotes, notes);

if (method == "triage")

variant8.triage (allNotes, notes);

if (method == "del")

variant8.del (allNotes);

if (method == "print")

variant8.print (allNotes, notes);

}

// variant8.del(allNotes,notes);

// variant8.print(allNotes,notes);

delete[]allNotes;

}

[storage.cpp](https://www.onlinegdb.com/online_c++_compiler#_editor_8188653561):

#include "storage.h"

#include <iostream>

#include <string>

#include <algorithm>

storage::storage ()

{

}

storage::storage (bool value)

{

cout << "------------------------" << endl;

cout << " list of methods:" << endl;

cout << "------------------------" << endl;

cout << " 1. add - adding a new record" << endl;

cout << " 2. edit - edit an already created record" << endl;

cout << " 3. search - search for an entry by name" << endl;

cout << " 4. triage - sort records by selected field" << endl;

cout << " 5. del - delete record by id" << endl;

cout << " 6. print - output of all records" << endl;

cout << "------------------------" << endl;

}

void

storage::add (storage \* arr, int notes)

{

std::cout << "------------------------" << std::endl;

std::cout << "Add()" << std::endl;

std::cout << "------------------------" << std::endl;

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

{

if (!arr[i].barCode)

{

std::cout << "Note id " << i << std::endl;

std::cout << "------------------------" << std::endl;

std::cout << "Enter bar code:";

std::cin >> arr[i].barCode;

std::cout << "Enter name:";

std::cin >> arr[i].name;

std::cout << "Enter weight:";

std::cin >> arr[i].weight;

std::cout << "Enter price:";

std::cin >> arr[i].price;

std::cout << "Enter number:";

std::cin >> arr[i].number;

std::cout << "------------------------" << std::endl;

break;

}

}

}

void

storage::del (storage \* arr)

{

std::cout << "------------------------" << std::endl;

std::cout << "Del()" << std::endl;

std::cout << "------------------------" << std::endl;

int id;

std::cout << "Enter id note :";

std::cin >> id;

storage sub;

arr[id] = sub;

arr[id].barCode = 0;

std::cout << "------------------------" << std::endl;

}

void

storage::print (storage \* arr, int notes)

{

std::cout << "------------------------" << std::endl;

std::cout << "Print()" << std::endl;

std::cout << "------------------------" << std::endl;

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

{

if (arr[i].barCode)

{

std::cout << "Note id " << i << std::endl;

std::cout << "------------------------" << std::endl;

std::cout << "bar code:" << arr[i].barCode << std::endl;

std::cout << "name:" << arr[i].name << std::endl;

std::cout << "weight:" << arr[i].weight << std::endl;

std::cout << "price:" << arr[i].price << std::endl;

std::cout << "number:" << arr[i].number << std::endl;

std::cout << "------------------------" << std::endl;

}

}

}

void

storage::search (storage \* arr, int notes)

{

std::cout << "------------------------" << std::endl;

std::cout << "Search()" << std::endl;

std::cout << "------------------------" << std::endl;

int code;

std::cout << "Enter bar code: ";

cin >> code;

std::cout << "------------------------" << std::endl;

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

{

if (code == arr[i].barCode)

{

std::cout << "Note id " << i << std::endl;

std::cout << "------------------------" << std::endl;

std::cout << "bar code:" << arr[i].barCode << std::endl;

std::cout << "name:" << arr[i].name << std::endl;

std::cout << "weight:" << arr[i].weight << std::endl;

std::cout << "price:" << arr[i].price << std::endl;

std::cout << "number:" << arr[i].number << std::endl;

std::cout << "------------------------" << std::endl;

}

}

}

void

storage::edit (storage \* arr)

{

std::cout << "------------------------" << std::endl;

std::cout << "Edit()" << std::endl;

std::cout << "------------------------" << std::endl;

int id;

std::cout << "Enter id note :";

std::cin >> id;

std::cout << "------------------------" << std::endl;

std::cout << "Enter bar code:";

std::cin >> arr[id].barCode;

std::cout << "Enter name:";

std::cin >> arr[id].name;

std::cout << "Enter weight:";

std::cin >> arr[id].weight;

std::cout << "Enter price:";

std::cin >> arr[id].price;

std::cout << "Enter number:";

std::cin >> arr[id].number;

std::cout << "------------------------" << std::endl;

}

void

storage::triage (storage \* arr, int notes)

{

std::cout << "------------------------" << std::endl;

std::cout << "Sort()" << std::endl;

std::cout << "------------------------" << std::endl;

string field;

float \*resultSortFloat = new float[notes];

string \*resultSortString = new string[notes];

storage \*resultSort = new storage[notes];

std::cout << "select the field:" << std::endl;

std::cout << "------------------------" << std::endl;

cout << "1. barCode" << endl;

cout << "2. name" << endl;

cout << "3. weight" << endl;

cout << "4. price" << endl;

cout << "5. number" << endl;

std::cout << "------------------------" << std::endl;

std::cin >> field;

std::cout << "------------------------" << std::endl;

if (field == "barCode")

{

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

resultSortFloat[i] = arr[i].barCode;

sort (resultSortFloat, resultSortFloat + notes);

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

for (int j = 0; j < notes; j++)

if (resultSortFloat[i] == arr[j].barCode)

resultSort[i] = arr[j];

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

arr[i] = resultSort[i];

}

if (field == "weight")

{

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

resultSortFloat[i] = arr[i].weight;

sort (resultSortFloat, resultSortFloat + notes);

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

for (int j = 0; j < notes; j++)

if (resultSortFloat[i] == arr[j].weight)

resultSort[i] = arr[j];

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

arr[i] = resultSort[i];

}

if (field == "price")

{

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

resultSortFloat[i] = arr[i].price;

sort (resultSortFloat, resultSortFloat + notes);

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

for (int j = 0; j < notes; j++)

if (resultSortFloat[i] == arr[j].price)

resultSort[i] = arr[j];

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

arr[i] = resultSort[i];

}

if (field == "number")

{

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

resultSortFloat[i] = arr[i].number;

sort (resultSortFloat, resultSortFloat + notes);

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

for (int j = 0; j < notes; j++)

if (resultSortFloat[i] == arr[j].number)

resultSort[i] = arr[j];

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

arr[i] = resultSort[i];

}

if (field == "name")

{

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

resultSortString[i] = arr[i].name;

sort (resultSortString, resultSortString + notes);

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

for (int j = 0; j < notes; j++)

if (resultSortString[i] == arr[j].name)

resultSort[i] = arr[j];

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

arr[i] = resultSort[i];

}

delete[]resultSortString;

delete[]resultSortFloat;

delete[]resultSort;

}

[storage.h](https://www.onlinegdb.com/online_c++_compiler#_editor_8192303411):

#include <string>

using namespace std;

class storage{

private:

float barCode;

string name;

float weight;

float price;

float number;

public:

storage();

storage(bool value);

void add(storage \*arr,int notes);

void del(storage \*arr);

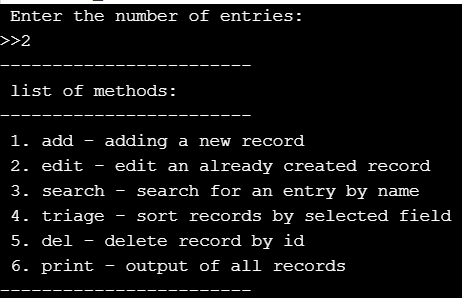
void print(storage \*arr,int notes);

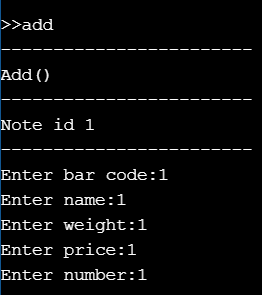
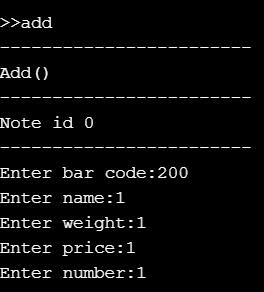
void search(storage \*arr,int notes);

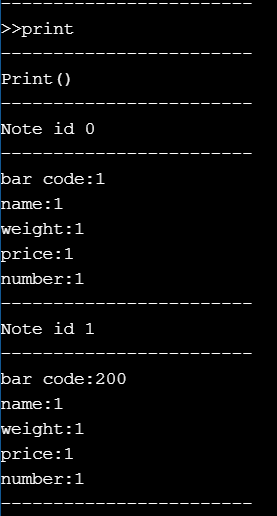
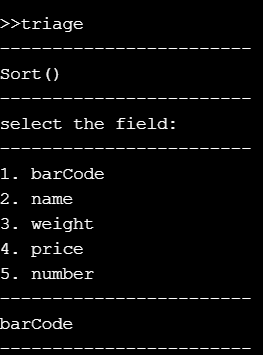
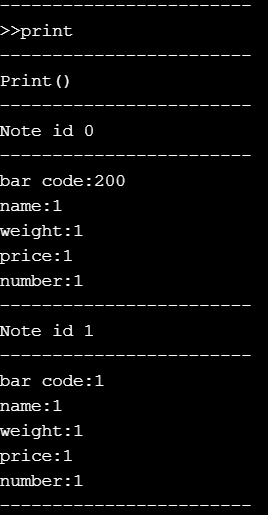
void edit(storage \*arr);

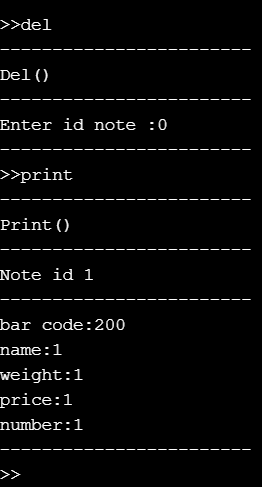
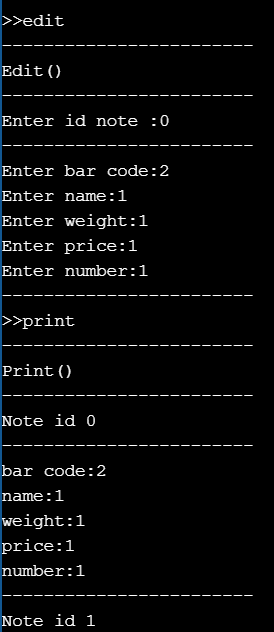
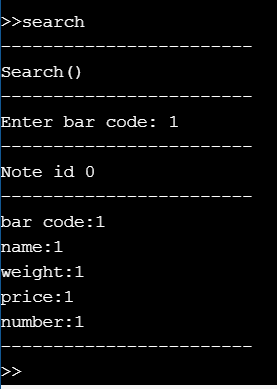
void triage(storage \*arr,int notes);

};









**Висновок:**

Завдяки виконанню лабораторної роботи 1 я порівняв об’єктно-орієнтований та функціональний підхід; ознайомився з класами, об’єктами та головними елементами об’єктного підходу.