**М і н і с т е р с т в о о с в і т и і н а у к и У к р а ї н и**

**Національний університет “Львівська політехніка”** Кафедра ЕОМ



**Курсова робота**

з предмету: «Програмування, частина 2 (Об’єктно-

орієнтоване програмування)»

на тему:

**«Базові принципи об’єктно-орієнтованого програмування»**

Індивідуальне завдання: «АЗС»

Виконав:

ст. гр. КІ-15

ГорбачукВ.Р.

Прийняв:

викладач

Козак Н.Б.

Львів-2020

Зміст

[АНОТАЦІЯ 2](#_Toc41039886)

[ЗАВДАННЯ НА КУРСОВУ РОБОТУ 3](#_Toc41039887)

[ВСТУП 4](#_Toc41039888)

[1.ОГЛЯД ТА ОБГРУНТУВАННЯ ВИБОРУ ТЕХНОЛОГІЙ ОБЄКТНО-ОРІЄНТОВАНОГО ПРОГРАМУВАННЯ 5](#_Toc41039889)

[2.АНАЛІЗ ТА РОЗРОБКА АЛГОРИТМУ ЗГІДНО ІНДИВІДУАЛЬНОГО ЗАВДАННЯ 7](#_Toc41039890)

[2.1.Опис алгоритму та його складності. 7](#_Toc41039891)

[2.2. Аналіз особливостей алгоритму. 8](#_Toc41039892)

[2.3. Блок-схема алгоритму 9](#_Toc41039893)

[3.ПРОГРАМНА РЕАЛІЗАЦІЯ АЛГОРИТМУ ЗГІДНО ІНДИВІДУАЛЬНОГО ЗАВДАННЯ 10](#_Toc41039894)

[3.1. Опис програмної реалізації алгоритму. 10](#_Toc41039895)

[3.1.1. Опис структур даних програми. 10](#_Toc41039896)

[3.1.2 Порядок викликів функцій 10](#_Toc41039897)

[3.1.3. Опис реалізаціх основних функцій/методів 11](#_Toc41039898)

[4. ВІДЛАГОЖДЕННЯ ТА ТЕСТУВАННЯ ПРОГРАМИ ЗГІДНО З ІНДИВІДУАЛЬНИМ ЗАВДАННЯМ 19](#_Toc41039899)

[4.1.Відлагодження та тестування певних частин програмою. 19](#_Toc41039900)

[4.2. Відлагождення та тестування програми вцілому. 19](#_Toc41039901)

[5. АНАЛІЗ ПРОДУКТИВНОСТІ(ЧАСУ РОБОТИ) ТА СТРУКТУРНОЇ СКЛАДНОСТІ РЕАЛІЗОВАНОЇ ПРОГРАМИ ЗГІДНО ІНДИВІДУАЛЬНОГО ЗАВДАННЯ 24](#_Toc41039902)

[5.1. Аналіз структурної складності виконання програми. 24](#_Toc41039903)

[ВИСНОВОК 25](#_Toc41039904)

АНОТАЦІЯ

Для розв’язання багатьох практичних задач з різних галузей, необхідно використовувати базові принципи об’єктно-орієнтованого програмування, оскільки вони можуть адаптувати вирішення поставлених задач до їх швидкої зміни в реальному часі. Одним з шляхів забезпечення таких вимог є організація об’єктно-орієнтованих процесів і відповідних технічних засобів їх реалізації. Причому, ефективність застосування об’єктно-орієнтованого програмування у вирішенні поставленого завдання залежить, як від продуктивності комп’ютерів, так і від розмірів і структури пам’яті, пропускної здатності каналів зв’язку, використаних об’єктно-орієнтованих мов програмування, компіляторів, операційних систем, чисельних методів та інших математичних досліджень. Такий широкий обсяг параметрів вимагає проведення досліджень на різних рівнях: на рівні опису алгоритмів, створення спеціальних мов програмування, компіляторів, багатопроцесорних систем, неоднорідних систем, кластерів. Для скорочення термінів розробки об’єктно-орієнтованих програм та дослідження їх роботи використовується моделювання. Метою виконання курсової роботи є засвоєння основних методів та алгоритмів об’єктно-орієнтованих структур, принципів побудови відповідних структур, набуття початкових практичних навиків проектування таких засобів. В результаті вивчення курсу студент повинен: знати: основні методи, алгоритми і засоби опрацювання інформації, засоби програмування об’єктно-орієнтованих структур, склад апаратних засобів та програмного забезпечення обчислювальних систем; вміти: виконувати елементарні вправи з об’єктно-орієнтованого програмування, описувати алгоритми поставленого перед ним завдання, використовувати класи та об'єкти, застосовувати модульне програмування.

ЗАВДАННЯ НА КУРСОВУ РОБОТУ

Індивідуальне завдання «АЗС».

Вхідні дані взято з сайту одного з АЗС WOG. А саме: принципи купівлі палива, автохімії і напоїв, принцип розподілу роботи між працівниками. На базі цієї інформації було створено алгоритм програми, яка буде реалізовувати ці дані. АЗС було поділено на продукти і персонал. Усе це реалізовано за допомогою класів та їх методів.

ВСТУП

У сучасному світі більшість людей живуть в жвавому ритмі життя тому їм потрібно все робити без зайвих затрат в часі тому сайт для АЗС допоможе їм зекономити час не чекаючи на каву або поки спакують ваші покупки. Людина придбавши щось на сайті може прибути на АЗС і забрати своє замовлення без зайвих витрат на час, показавши чек. Тому тема моєї курсової є досить актуальною.

Метою моєї курсової роботи є: оволодіти базовими елементами мови програмування С++, та за допомогою них написати програму.

1.ОГЛЯД ТА ОБГРУНТУВАННЯ ВИБОРУ ТЕХНОЛОГІЙ ОБЄКТНО-ОРІЄНТОВАНОГО ПРОГРАМУВАННЯ

Об'єктно-орієнтоване програмування (ООП) — одна з парадигм програмування, яка розглядає програму як множину «об'єктів», що взаємодіють між собою. Основу ООП складають три основні концепції: інкапсуляція, успадкування та поліморфізм. Одною з переваг ООП є краща модульність програмного забезпечення.

Успадкування (наслідування). Клас може мати «підкласи», спеціалізовані, розширені версії надкласу. Можуть навіть утворюватися цілі дерева успадкування. Підкласи успадковують атрибути та поведінку своїх батьківських класів, і можуть вводити свої власні. Успадкування може бути одиничне (один безпосередній батьківський клас) та множинне (кілька батьківських класів).

Інкапсуляція. Приховування деталей про роботу класів від об'єктів, що їх використовують або надсилають їм повідомлення. Інкапсуляція досягається шляхом вказування, які класи можуть звертатися до членів об'єкта. Як наслідок, кожен об'єкт представляє кожному іншому класу певний інтерфейс — члени, доступні іншим класам. Інкапсуляція потрібна для того, аби запобігти використанню користувачами інтерфейсу тих частин реалізації, які, швидше за все, будуть змінюватись. Це дозволить полегшити внесення змін, тобто, без потреби змінювати і користувачів інтерфейсу. Часто, члени класу позначаються як публічні (англ. public), захищені (англ. protected) та приватні (англ.private), визначаючи, чи доступні вони всім класам, підкласам, або лише до класу в якому їх визначено.

Поліморфізм. Поліморфізм означає залежність поведінки від класу, в якому ця поведінка викликається, тобто, два або більше класів можуть реагувати по-різному на однакові повідомлення. На практиці - це реалізовується шляхом реалізації ряду підпрограм (функцій, процедур, методів тощо) з однаковими іменами, але з різними параметрами. В залежності від того, що передається і вибирається відповідна підпрограма.

Нижче наведено основні принципи ООП. Вони є дуже важливими, адже дозволяють захищати дані в класі, дозволяють брати необхідну інформацію одним класам у інших. Це робить написання коду набагато зручнішим і ефективнішим. Тому я вважаю було доцільним використовувати ці принци під час написання програми.

2.АНАЛІЗ ТА РОЗРОБКА АЛГОРИТМУ ЗГІДНО ІНДИВІДУАЛЬНОГО ЗАВДАННЯ

2.1.Опис алгоритму та його складності.

Алгоритм розпочинається із основної конструкції switch, що утворює меню консольної програми. Це розгалуження відповідає за головне меню програми, яке має такі можливості:

1) перейти до роздiлiв товару

2) перегляд збережених товарiв

3) перегляду персоналу

4) перегляду iнформацiї про запрaвку

5)вийти

Обравши 1-ий варіант користувачу виводиться на екран список розділів товарів. І одна конструкція switch, за допомогою ми можемо обрати розділ бажаного товару, а саме:

A) перегляд палива

Б) перегляд напоїв

В) перегляд автохімії

Г) перейти до головного меню

Обравши варіант А користувачу виводиться таблиця з 6 типами палива в якому вказано його ціну і наявний об’єм палива на АЗС. Після чого користувач може обрати бажане паливо і потрібний об’єм. Далі його покаже ціну за куплене паливо і перенесе до меню з товарами.

Обравши варіант Б користувачу виводиться таблиця з 9 напоями в якому вказано його ціну і розмір склянки. Після чого користувачу запропонують обрати бажаний напій і кількість штук цього напою далі йому знов запропонують обрати напій і відмовившись йому покаже ціну за куплені напої

Обравши варіант В користувачу виводиться таблиця з 6 продуктами автохімії в якому вказано його ціну і об’єм продукта. Після чого користувачу запропонують обрати бажаний продукт автохімії і обрашви щось далі йому знов запропонують обрати продукт автохімії і відмовившись йому покаже ціну за куплену автохімію.

Обравши варіант Г користувач поветається до головного меню.

Обравши 2-ий варіант користувачу виводиться на екран список його покупок і загальна ціна всіх покупок. Користувач може очистити список покупок за вибором і повернутися до головного меню.

Обравши 3-ий варіант то спрацьовує наступна конструкція switch, за допомогою якої можна обрати наступні дії:

А) побачити повний список персоналу

Б) побачити список прибиральників

В) побачити список охорони

Г) побачити список касирiв

Д) побачити список механiки

Е) побачити список адмiнiстраторiв

Є) переходу в головне меню

Варіант А виводить весь список персоналу, а варіанти від Б до Е

Виводить працівників певної професії. Останій варіант повертає користувача назад в головне меню.

Обравши 4-ій варіант користувачу виводиться вся інформація про АЗС.

2.2. Аналіз особливостей алгоритму.

У кожній конструкції switch викликаються методи відповідних класів.

1. У першому випадку відкриваються ті розділи товарів, які обрав користувач.( CCout\_fuil(), CCout\_avtohim() i CCout\_drinks())
2. У другому випадку використовуються функція check()
3. У третьому випадку використовуються функція
4. У четвертій конструкції використовуються функції: station\_name() і Full\_information().

2.3. Блок-схема алгоритму

Кінець

Full\_information().

station\_name()

staff\_inf()

check()

CCout\_drinks()

CCout\_avtohim()

CCout\_fuil()

Switch

(choice2))

Ввід choice2

Switch

(choice1))

Ввід choice1

Початок

3.ПРОГРАМНА РЕАЛІЗАЦІЯ АЛГОРИТМУ ЗГІДНО ІНДИВІДУАЛЬНОГО ЗАВДАННЯ

3.1. Опис програмної реалізації алгоритму.

3.1.1. Опис структур даних програми.

В програмі наявні наступні типи даних:

А)Змінні цілого типу для виконання конструкції switch.

Б)Змінні типу double в яких містилася ціна купленого товару.

В)Змінні символьного типу.

Г) Об’єкти класів для викликання методів класів.

3.1.2 Порядок викликів функцій

А) Отримання першого вибору

Б) Перехід до наступного вибору в залежності від першого вибору

Можливі виконання функції:

А) Вивід на екран розділів товарів

Б) Вивід на екран списку куплених товарів і загальної ціни покупки

В) Ввід необхідних даних про працівників

Г) Вивід на екран певної інформації

Усі функції знаходяться у класах які побудовані наступним чином: існує незалежний клас WOG, який має функції привітання та показу інформації про АЗС. Клас Staff також не залежить від жодного іншого класу. Він містить в собі функції показу списку працівників АЗС. Базовий клас Goods і 3 похідних класи СFuil, Drinks i Product.

Діаграма класів наведена нижче

Product

Drinks

СFuil

Goodss

Staff

WOG

3.1.3. Опис реалізаціх основних функцій/методів

Кожен клас реалізований за допомого файла header , де зберігає в собі оголошення змінних класу та методів і де вони реалізуються.

1)Вивід розділів товарів

A)Паливо

double sumfuil;

double CCout\_fuil() {

double neededfuil;

string fuilname;

int choice4,choice3, volumeoffuel;

cout << "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" << endl;

cout << "|No.|Паливо |Доступний об'єм |Цiна |\n";

CFuil first;

first = CFuil("ДП MUSTANG", 3000, 26.99);

cout << "| 1 |" << first.GetName()<<setw(3) << "|" << first.GetVolumeAvaible()<<setw(13) << "|" << first.Getcost() << "|\n";

CFuil second;

second = CFuil("ДП MUSTANG+", 2500, 27.99);

cout << "| 2 |" << second.GetName() << setw(2) << "|" << second.GetVolumeAvaible() << setw(13) << "|" << second.Getcost() << "|\n";

CFuil third;

third = CFuil("92 MUSTANG", 1200, 26.99);

cout << "| 3 |" << third.GetName() << setw(3) << "|" << third.GetVolumeAvaible() << setw(13) << "|" << third.Getcost() << "|\n";

CFuil fourth;

fourth = CFuil("95 MUSTANG", 5000, 26.99);

cout << "| 4 |" <<fourth.GetName() << setw(3) << "|" <<fourth.GetVolumeAvaible() << setw(13) << "|" <<fourth.Getcost() << "|\n";

CFuil fifth;

fifth = CFuil("100 MUSTANG", 600, 26.99);

cout << "| 5 |" << fifth.GetName() << setw(2) << "|" << fifth.GetVolumeAvaible() << setw(14) << "|" << fifth.Getcost() << "|\n";

CFuil sixth;

sixth = CFuil("ГАЗ", 2000, 26.99);

cout << "| 6 |" << sixth.GetName() << setw(10) << "|" << sixth.GetVolumeAvaible() << setw(13) << "|" << sixth.Getcost() << "|\n";

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

cout << "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" << endl;

cout << "press 1 - для cтворення покупки" << endl;

cout << "press 2 - для виходу до меню " << endl;

cout << "Ваш вибiр:";

cin >> choice3;

if (choice3 == 1) {

cout << "Оберiть паливо за номером: ";

cin >> choice4;

switch (choice4) {

case 1:

neededfuil = first.Getcost();

fuilname = first.GetName();

break;

case 2:

neededfuil = second.Getcost();

fuilname = second.GetName();

break;

case 3:

neededfuil = third.Getcost();

fuilname = third.GetName();

break;

case 4:

neededfuil = fourth.Getcost();

fuilname = fourth.GetName();

break;

case 5:

neededfuil = fifth.Getcost();

fuilname = fifth.GetName();

break;

case 6:

neededfuil = sixth.Getcost();

fuilname = sixth.GetName();

break;

default: cout << "Вибран не iснуючий пункт меню" << endl;

system("pause");

}

cout << "Введiть необхiдний об'єм палива:";

cin >> volumeoffuel;

sumfuil = volumeoffuel \* neededfuil;

cout << "Цiна покупки:";

cout << sumfuil << endl;

ofstream outshoppingfile;

outshoppingfile.open("MyFile.txt", ios::app);

if (!outshoppingfile.is\_open()) {

cout << "ERROR";

}

outshoppingfile << "Тип покупки:палне;\n" << "Назва пального:"<<fuilname<<";"<<endl

<<"Об'єм купленого пального:"<< volumeoffuel << ";" << endl

<<"Цiна за пальне:"<<sumfuil << ";" << endl;

outshoppingfile.close();

system("pause");

}

return sumfuil;

}

Б)Напої

Double sum\_all\_drinks=0;

double CCout\_drinks() {

cout << "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" << endl;

cout << "|No.|Напої |Розмiр|Цiна|\n";

Drinks first;

first = Drinks("Flat Black", "L", 34.00);

cout << "| 1 |" << first.GetName() << setw(9) << "|" << setw(4) << first.GetSize() << setw(3) << "|" << setw(3) << first.Getcost() << " |\n";

Drinks second;

second = Drinks("Flat White", "L", 39.00);

cout << "| 2 |" << second.GetName() << setw(9) << "|" << setw(4) << second.GetSize() << setw(3) << "|" << setw(3) << second.Getcost() << " |\n";

Drinks third;

third = Drinks("Aмерикано", "M", 28.00);

cout << "| 3 |" << third.GetName() << setw(10) << "|" << setw(4) << third.GetSize() << setw(3) << "|" << setw(3) << third.Getcost() << " |\n";

Drinks fourth;

fourth = Drinks("Гарячий шоколад", "L", 39.00);

cout << "| 4 |" << fourth.GetName() << setw(4) << "|" << setw(4) << fourth.GetSize() << setw(3) << "|" << setw(3) << fourth.Getcost() << " |\n";

Drinks fifth;

fifth = Drinks("Гарячий шоколад", "M", 29.00);

cout << "| 5 |" << fifth.GetName() << setw(4) << "|" << setw(4) << fifth.GetSize() << setw(3) << "|" << setw(3) << fifth.Getcost() << " |\n";

Drinks sixth;

sixth = Drinks("Фiрмовий чай", "XXL", 35.00);

cout << "| 6 |" << sixth.GetName() << setw(7) << "|" << setw(4) << sixth.GetSize() << setw(3) << "|" << setw(3) << sixth.Getcost() << " |\n";

Drinks seventh;

seventh = Drinks("Експресо","S",27.00);

cout << "| 7 |" << seventh.GetName() << setw(11) << "|" << setw(4) << seventh.GetSize() << setw(3) << "|" << setw(3) << seventh.Getcost() << " |\n";

Drinks eighth;

eighth = Drinks("Капучино", "L", 34.00);

cout << "| 8 |" << eighth.GetName() << setw(11) << "|" << setw(4) << eighth.GetSize() << setw(3) << "|" << setw(3) << eighth.Getcost() << " |\n";

Drinks ninth;

ninth = Drinks("Капучино", "XXL", 39.00);

cout << "| 9 |" << ninth.GetName() << setw(11) << "|" << setw(4) << ninth.GetSize() << setw(3) << "|" << setw(3) << ninth.Getcost() << " |\n";

cout << "------------------------------------\n";

cout << "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" << endl;

cout << "press 1 - для cтворення покупки" << endl;

cout << "press 2 - для виходу до меню " << endl;

cout << "Ваш вибiр:";

int choice,choice2,choice3,quantity;

double needed\_drink=0,sum\_drinks;

string drink\_name, drink\_size;

cin >> choice;

if (choice == 1) {

ofstream outshoppingfile;

outshoppingfile.open("MyFile.txt", ios::app);

if (!outshoppingfile.is\_open()) {

cout << "ERROR";

}

do {

cout << "Оберiть напiй за номером: ";

cin >> choice2;

switch (choice2)

{

case 1:

needed\_drink = first.Getcost();

drink\_name = first.GetName();

drink\_size = first.GetSize();

break;

case 2:

needed\_drink = second.Getcost();

drink\_name = second.GetName();

drink\_size = second.GetSize();

break;

case 3:

needed\_drink = third.Getcost();

drink\_name = third.GetName();

drink\_size = third.GetSize();

break;

case 4:

needed\_drink = fourth.Getcost();

drink\_name = fourth.GetName();

drink\_size = fourth.GetSize();

break;

case 5:

needed\_drink = fifth.Getcost();

drink\_name = fifth.GetName();

drink\_size = fifth.GetSize();

break;

case 6:

needed\_drink = sixth.Getcost();

drink\_name = sixth.GetName();

drink\_size = sixth.GetSize();

break;

case 7:

needed\_drink = seventh.Getcost();

drink\_name = seventh.GetName();

drink\_size = seventh.GetSize();

break;

case 8:

needed\_drink = eighth.Getcost();

drink\_name = eighth.GetName();

drink\_size = eighth.GetSize();

break;

case 9:

needed\_drink = ninth.Getcost();

drink\_name = ninth.GetName();

drink\_size = ninth.GetSize();

break;

default:

cout << "Вибран не iснуючий пункт меню" << endl;

system("pause");

break;

}

cout << "Скiльки бажаєте таких напоїв:";

cin >> quantity;

sum\_drinks = quantity \* needed\_drink;

cout << "Цiна покупки:" << sum\_drinks<<endl;

sum\_all\_drinks += sum\_drinks;

outshoppingfile << "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n";

outshoppingfile << "Тип покупки:напiй;\n" << "Назва напою:" << drink\_name << ";" << endl;

outshoppingfile << "Розмiр напою:" << drink\_size << ";\n";

outshoppingfile << "Кiлькiсть:" << quantity << ";\n";

outshoppingfile << "Цiна :" << sum\_drinks << ";" << endl;

cout << "Чи бажаєте здiйснити ще одну покупку?\n";

cout << "press 1 - для cтворення покупки" << endl;

cout << "press 2 - для виходу до меню " << endl;

cin >> choice3;

} while (choice3 == 1);

outshoppingfile.close();

}

return sum\_all\_drinks;

}

В)Автохімія

double sum\_avto = 0

double CCout\_avtohim() {

double needed\_avto;

string avto\_name;

int choice3,choice5,avto\_volume;

cout << "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" << endl;

cout << "|No.| Автохiмiя |Об'єм|Цiна |\n";

Product first;

first = Product("Омивач скла WOG Winter", 1, 49.95);

cout << "| 1 |" << first.GetName() << setw(9) << "|" << setw(3) << first.GetVolume1() << "л." << "| " << first.Getcost() << "|\n";

Product second;

second = Product("Омивач скла WOG Winter", 4, 129.99);

cout << "| 2 |" << second.GetName() << setw(9) << "|" << setw(3) << second.GetVolume1() << "л." << "|" << second.Getcost() << "|\n";

Product third;

third = Product("Всесезонний омивач скла WOG", 2, 43.55);

cout << "| 3 |" << third.GetName() << setw(4) << "|" << setw(3) << third.GetVolume1() << "л." << "| " << third.Getcost() << "|\n";

Product fourth;

fourth = Product("Омивач скла WOG Winter", 1, 49.95);

cout << "| 4 |" << fourth.GetName() << setw(9) << "|" << setw(3) << fourth.GetVolume1() << "л." << "| " << fourth.Getcost() << "|\n";

Product fifth;

fifth = Product("WOG Антифриз", 4, 349.45);

cout << "| 5 |" << fifth.GetName() << setw(19) << "|" << setw(3) << fifth.GetVolume1() << "л." << "|" << fifth.Getcost() << "|\n";

Product sixth;

sixth = Product("WOG гальмiвна рiдина", 1, 99.99);

cout << "| 6 |" << sixth.GetName() << setw(11) << "|" << setw(2) << sixth.GetVolume1() << " л." << "| " << sixth.Getcost() << "|\n";

cout << "-------------------------------------------------\n";

cout << "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" << endl;

cout << "press 1 - для cтворення покупки" << endl;

cout << "press 2 - для виходу до меню " << endl;

cout << "Ваш вибiр:";

cin >> choice3;

if (choice3 == 1) {

ofstream outshoppingfile;

outshoppingfile.open("MyFile.txt", ios::app);

if (!outshoppingfile.is\_open()) {

cout << "ERROR";

}

do {

cout << "Оберiть автохiмiю за номером: ";

int choice4;

cin >> choice4;

switch (choice4) {

case 1:

needed\_avto = first.Getcost();

avto\_name = first.GetName();

avto\_volume=first.GetVolume1();

break;

case 2:

needed\_avto = second.Getcost();

avto\_name = second.GetName();

avto\_volume = second.GetVolume1();

break;

case 3:

needed\_avto = third.Getcost();

avto\_name = third.GetName();

avto\_volume = third.GetVolume1();

break;

case 4:

needed\_avto = fourth.Getcost();

avto\_name = fourth.GetName();

avto\_volume = fourth.GetVolume1();

break;

case 5:

needed\_avto = fifth.Getcost();

avto\_name = fifth.GetName();

avto\_volume = fifth.GetVolume1();

break;

case 6:

needed\_avto = sixth.Getcost();

avto\_name = sixth.GetName();

avto\_volume = sixth.GetVolume1();

break;

default:

cout << "Вибран не iснуючий пункт меню" << endl;

system("pause");

break;

}

cout << "Цiна покупки:";

sum\_avto += needed\_avto;

cout << needed\_avto <<endl;

cout << "Чи бажаєте здiйснити ще одну покупку?\n";

cout << "press 1 - для cтворення покупки" << endl;

cout << "press 2 - для виходу до меню " << endl;

cin >> choice5;

outshoppingfile << "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n";

outshoppingfile << "Тип покупки:автохiмiя;\n" << "Назва автохiмiї:" << avto\_name << ";" << endl;

outshoppingfile << "Об'єм автохiмiї:" << avto\_volume << ";\n" ;

outshoppingfile << "Цiна за автохiмiю:" << needed\_avto << ";" << endl;

} while (choice5 == 1);

outshoppingfile << "Загальна цiна за автохiмiю:" << needed\_avto << ";\n";

outshoppingfile.close();

}

return sum\_avto;

}

2)Вивід записаних даних з файлу і вивід загальної суми покупки

void check() {

ofstream outshoppingfile;

outshoppingfile.open("MyFile.txt", ios::app);

if (!outshoppingfile.is\_open()) {

cout << "ERROR";

}

outshoppingfile << "=========================================\n";

outshoppingfile << "Загальна сума до сплати:" << sumfuil + sum\_avto + sum\_all\_drinks << ";\n";

outshoppingfile.close();

int cash;

string line;

system("cls");

ifstream inshoppingfile;

inshoppingfile.open("MyFile.txt");

if (!inshoppingfile.is\_open())

{

cout << "EROR";

}

while (!inshoppingfile.eof()) {

line = "";

getline(inshoppingfile, line);

cout << line << endl;

}

inshoppingfile.close();

cout << "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" << endl;

cout << "press 1 - для очищення корзини" << endl;

cout << "press 2 - для виходу до меню " << endl;

cin >> cash;

if (cash == 1) {

ofstream ofs;

ofs.open("MyFile.txt", ofstream::out | ofstream::trunc);

}

}

3) Перегляду персоналу

void staff\_inf() {

Staff f;

int choice;

f.staff\_list();

Staff:

system("cls");

cout << "press 1 - щоб побачити повний список персоналу" << endl;

cout << "press 2 - щоб побачити список прибиральникiв" << endl;

cout << "press 3 - щоб побачити список охорони" << endl;

cout << "press 4 - щоб побачити список касирiв" << endl;

cout << "press 5 - щоб побачити список механiки" << endl;

cout << "press 6 - щоб побачити список адмiнiстраторiв" << endl;

cout << "press 7 - для переходу в головне меню" << endl;

cin >> choice;

switch (choice)

{

case 1:

{

system("cls");

f.cleaners();

f.securyty();

f.cashiers();

f.mechanics();

f.administrator();

system("pause");

break;

}

case 2:

{

system("cls");

f.cleaners();

system("pause");

goto Staff;

break;

}

case 3:

{

system("cls");

f.securyty();

system("pause");

goto Staff;

break;

}

case 4:

{

system("cls");

f.cashiers();

system("pause");

goto Staff;

break;

}

case 5:

{

system("cls");

f.mechanics();

system("pause");

goto Staff;

break;

}

case 6:

{

system("cls");

f.administrator();

system("pause");

goto Staff;

break;

}

case 7:

{

break;

}

default:

{

system("cls");

cout << "Invalid input" << endl;

goto Staff;

break;

}

}

}

1. перегляду iнформацiї про запрaвку

void station\_name()

{

cout << "Ласкаво просимо до " << name << endl;

}

void full\_information()

{

cout << "АЗС знаходиться на " << streat << location << ". АЗС засновано " << date << ".\nВласник " << owner << "." << endl << "Графiк роботи:цiлодобовий.\n";

}

4. ВІДЛАГОЖДЕННЯ ТА ТЕСТУВАННЯ ПРОГРАМИ ЗГІДНО З ІНДИВІДУАЛЬНИМ ЗАВДАННЯМ

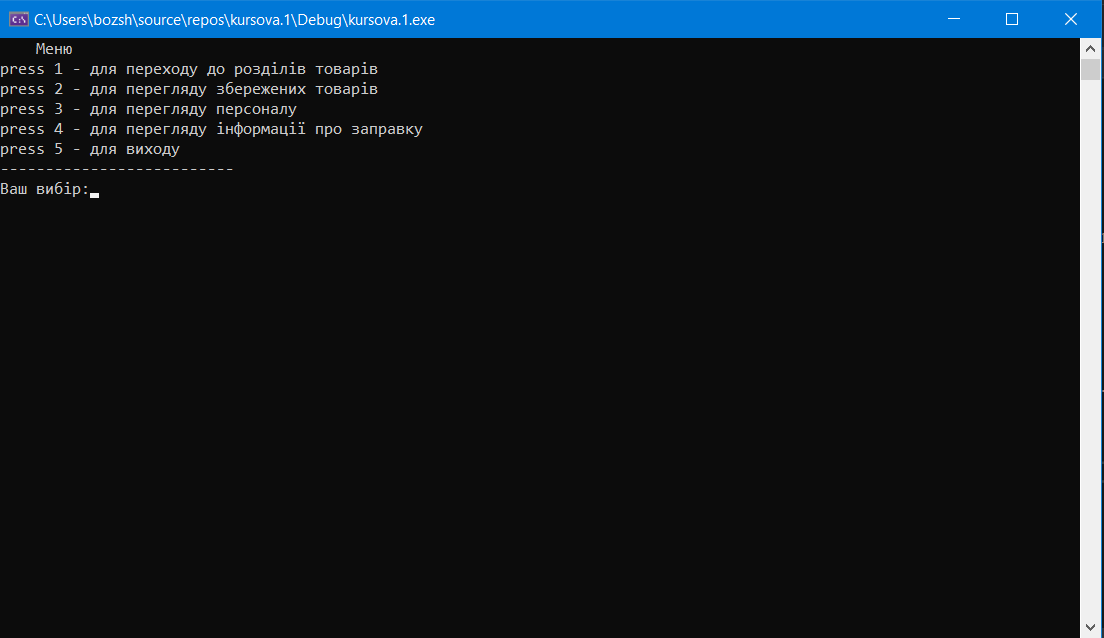
4.1.Відлагодження та тестування певних частин програмою.

Під час тестування окремих частин програм з’являлись певні недоліки та помилки. Щоб все правильно працювало потрібно було не дати можливості користувачу змушувати програму некоректно працювати. Тож всі можливі неполадки були ізольовані і опрацьовані так, щоб з’являлось певне повідомлення з певною інформацією, звідки користувач дізнавався як слід користуватися програмою, щоб уникати таких невизначеностей.

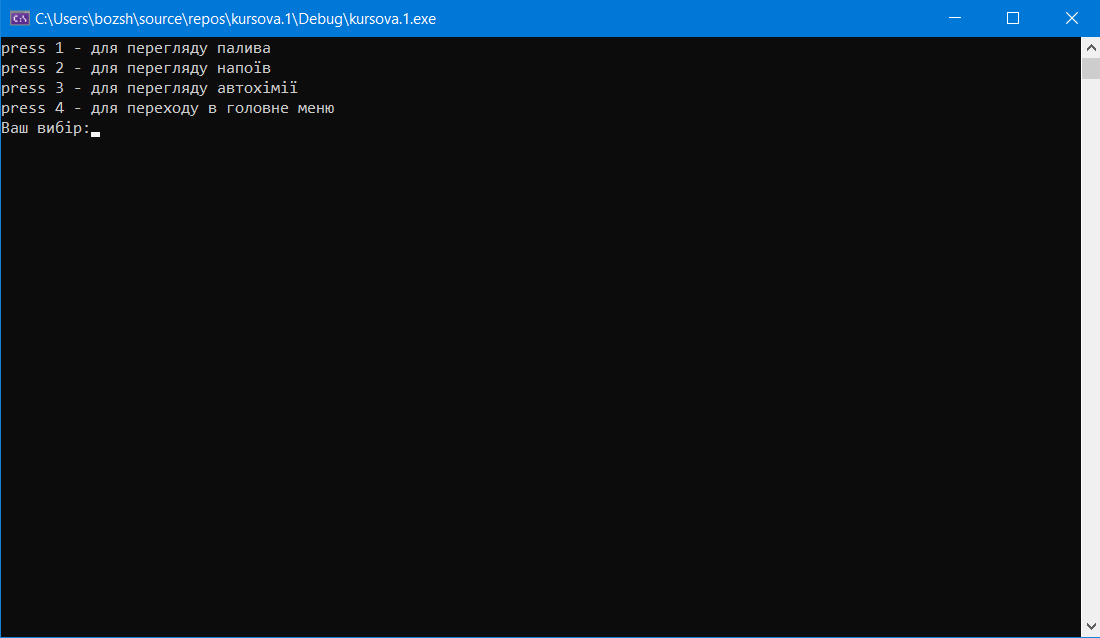
4.2. Відлагождення та тестування програми вцілому.

Подібні кроки були виконані і з перевіркою всієї програми. Нижче наведені скріншоти, які запевнюють в коректній роботі програми:

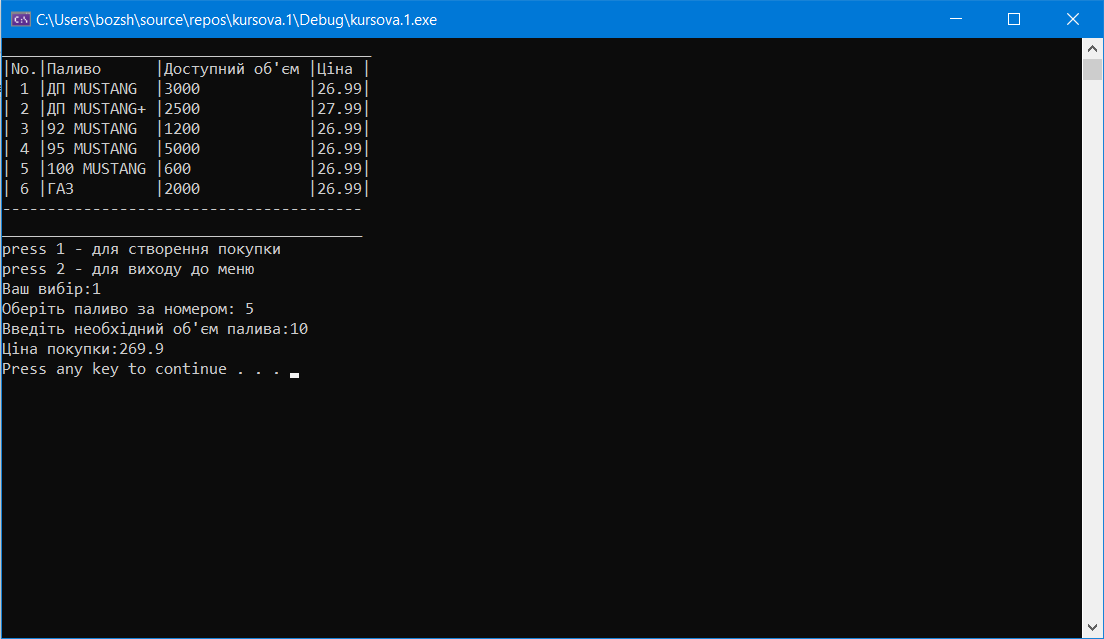
1)Головне меню



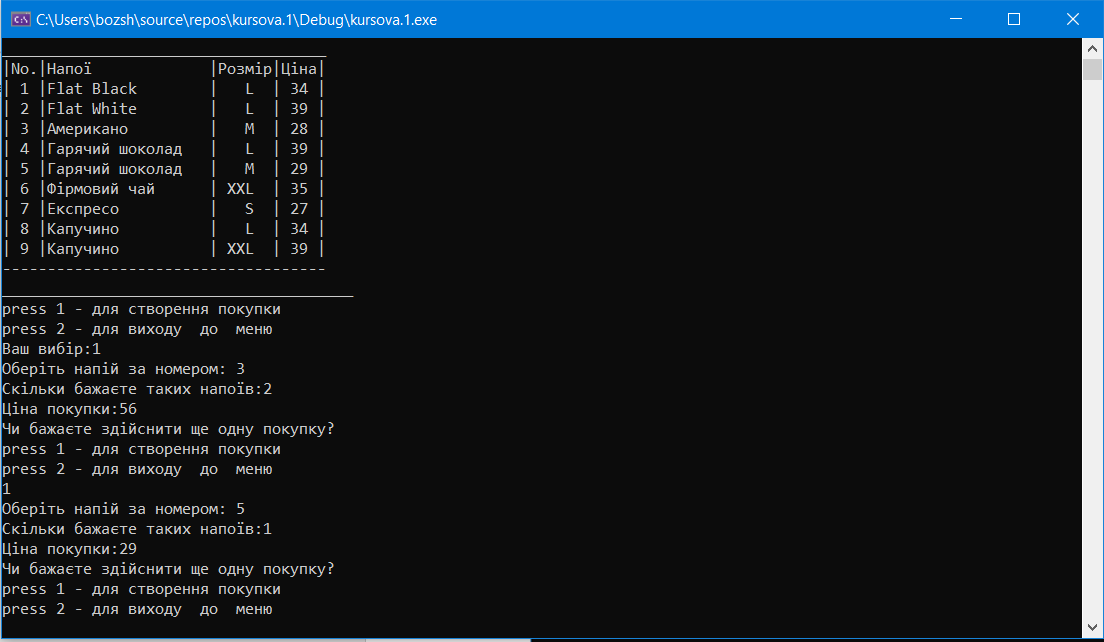
2)Перехід до розділів товарів



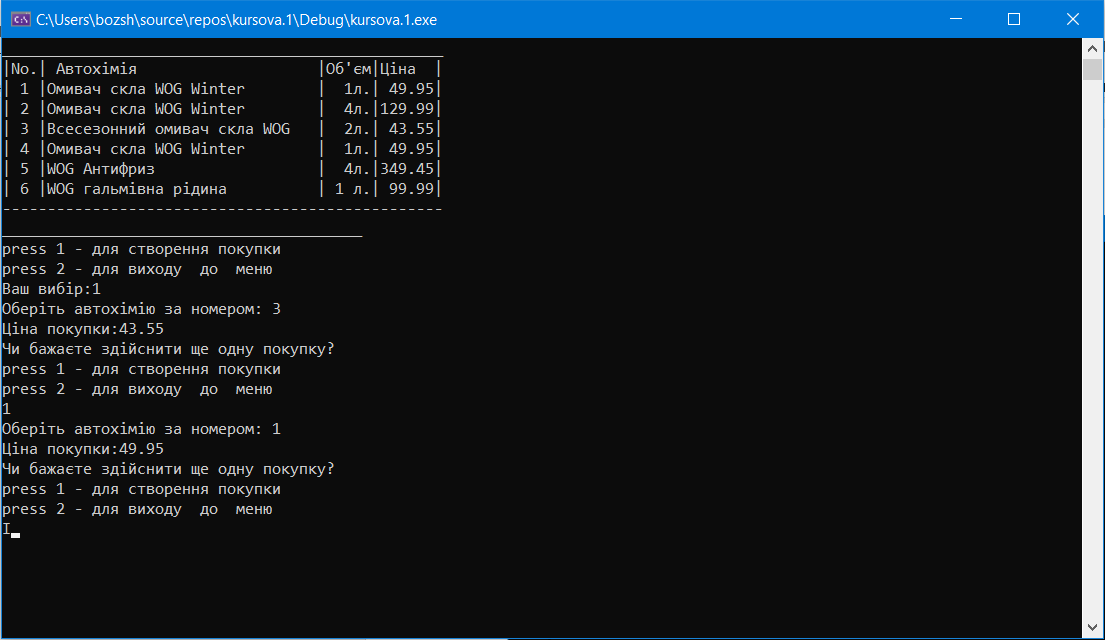
3)Перегляд і купівля палива



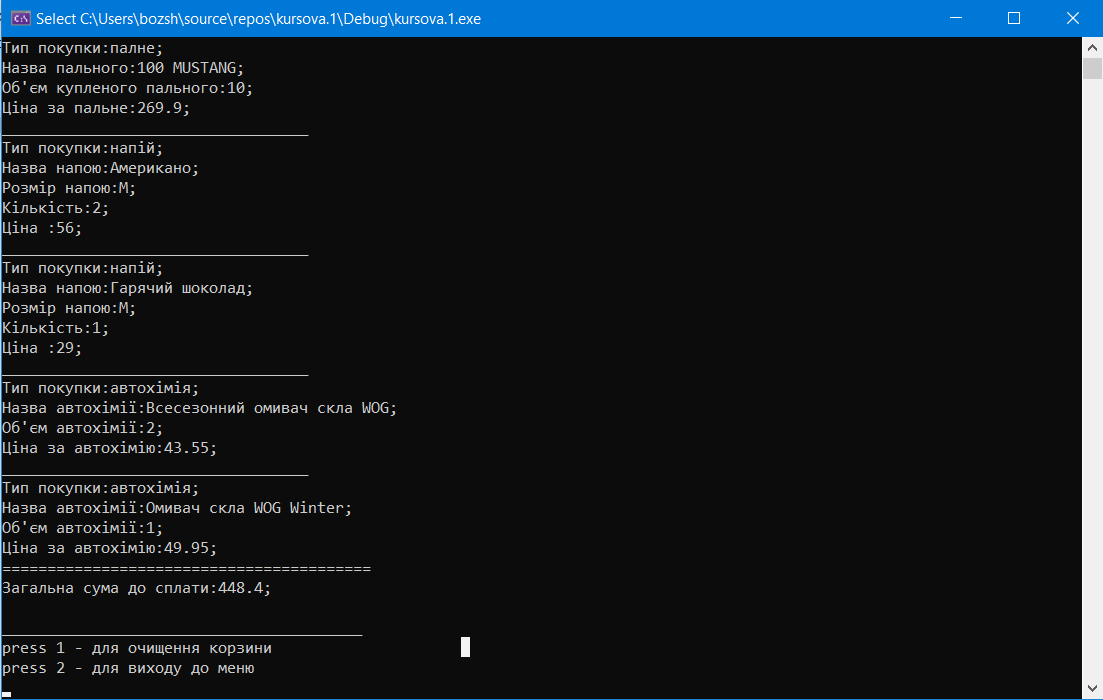
4) Перегляд і купівля напоїв



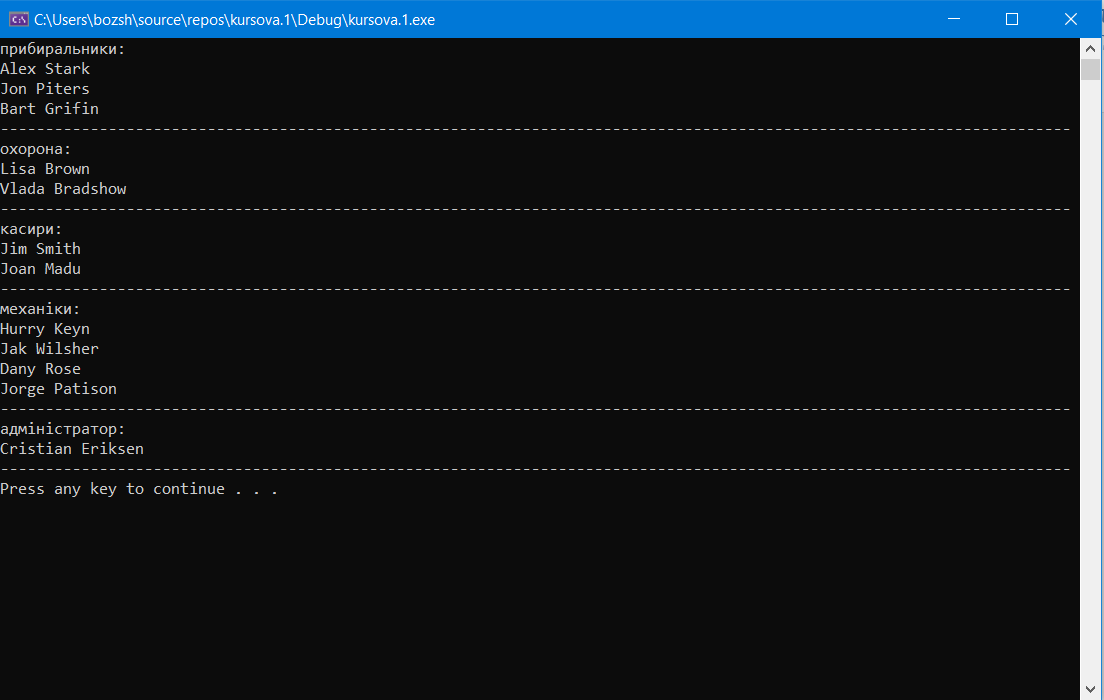
1. Перегляд і купівля автохімії



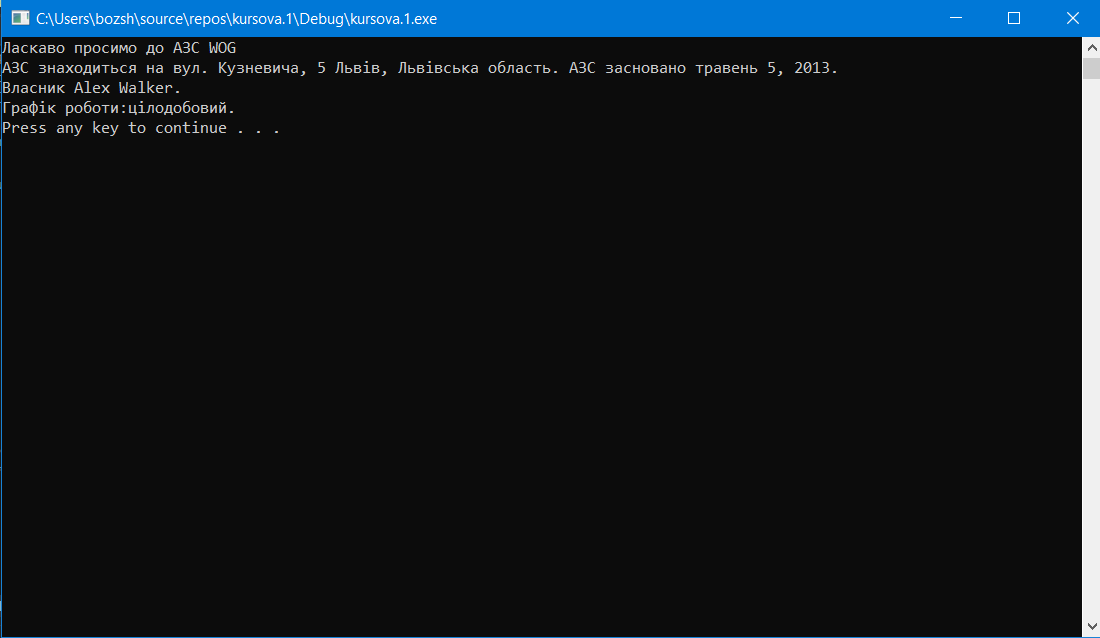
7)Перегляд куплених товарів і загальної суми покупки



8) перегляд персоналу



9) перегляд iнформацiї про запрaвку



5. АНАЛІЗ ПРОДУКТИВНОСТІ(ЧАСУ РОБОТИ) ТА СТРУКТУРНОЇ СКЛАДНОСТІ РЕАЛІЗОВАНОЇ ПРОГРАМИ ЗГІДНО ІНДИВІДУАЛЬНОГО ЗАВДАННЯ

5.1. Аналіз структурної складності виконання програми.

Оцінюючи структуру програми, потрібно наголосити, що наявні конструкції switch, полегшують не тільки користування програмою в ролі користувача, але як на мене, полегшують власне кодування. Ці конструкції начебто уявно відділяють блок коду від усієї програми, що покращує сприйняття й розуміння виконаних та запланованих цілей (функцій, операцій). Як висновок, можу сказати що, виконання програми на структурному рівні не є складним.

Всі операції всередині програми виконуються миттєво, немає затримки між переходами по меню та виведенням інформації на консоль.

ВИСНОВОК

При підготовці куросвої роботи я навчився багатьом новим речам. Навчився планувати майбутні дії, маючи якесь певне завдання та створювати блок-схему, яка допомфгає при написанні коду. Навчився, за допомогою конструкції switch робити головне меню програм та зрозумів її корисність у програмувані. Також оволодів базовими елементами об’єктно-орієнтованого програмування.