 МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ   
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

ФАКУЛЬТЕТ БІОМЕДИЧНОЇ ІНЖЕНЕРІЇ

КАФЕДРА БІОМЕДИЧНОЇ КІБЕРНЕТИКИ

**Компьютерный практикум №1**

з дисципліни «Об'єктно-орієнтоване програмування»

# на тему: «Реалізація класів, конструктори та деструктори»

Варіант №7

**Виконав:**

студент гр. БС-03

Затуловский Г. А.

**Перевірив:**

ас. каф. БМК Бабенко В.О.

Зараховано від \_\_\_.\_\_\_.\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

                      (підпис викладача)

Київ-2022

**Завдання:**

1. Вивчити теоретичні програмування С++ відомості щодо створення класiв

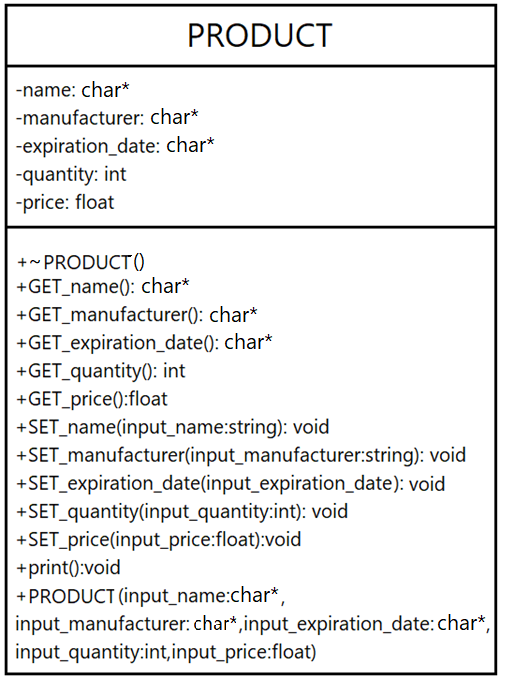
та об'єктів мовою програмування С++.

2. Вiдповiдно до свого варіанту розробити клас та відповідну діаграму класів в нотац UML для описаних в завданнi характерних даних, визначити конструктор користувача, за допомогою якого можна iнiцiалiзувати всі поля визначеного класу, а також методи для змiни значень кожного із полів:

3. Розробити програмний застосунок, в якому необхідно створити об'ект, що є екземпляром визначеного в заданого класу, та ініціалізувати його поля за допомогою створеного конструктора користувача, вивести інформацію про створений об'скт на екран, змінити значення полiв об'екта за допомогою викликiв відповідних методів класу, після чого знову вивести інформацію про модифікований об’єкт на екран, видалити створений об'єкт

4. Скласти і захистити звіт по роботі.

**UML діаграма:**



**Код програми:**

#include <iostream>

using namespace std;

class PRODUCT

{

private:

char\* name;

char\* manufacturer;

char\* expiration\_date;

int quantity;

float price;

public:

PRODUCT(char\* input\_name, char\* input\_manufacturer, char\* input\_expiration\_date, int input\_quantily, float input\_price) : name(input\_name), manufacturer(input\_manufacturer), expiration\_date(input\_expiration\_date), quantity(input\_quantily), price(input\_price)

{

name = input\_name;

manufacturer = input\_manufacturer;

expiration\_date = input\_expiration\_date;

quantity = input\_quantily;

price = input\_price;

}

~PRODUCT()

{

cout << " Destructor called" << endl;

delete[] name;

delete[] manufacturer;

delete[] expiration\_date;

}

char\* GET\_name()

{

return name;

}

char\* GET\_manufacturer()

{

return manufacturer;

}

char\* GET\_expiration\_date()

{

return expiration\_date;

}

int GET\_quantity()

{

return quantity;

}

float GET\_price()

{

return price;

}

void SET\_name(char\* input\_name)

{

name = input\_name;

}

void SET\_manufacturer(char\* input\_manufacturer)

{

manufacturer = input\_manufacturer;

}

void SET\_expiration\_date(char\* input\_expiration\_date)

{

expiration\_date = input\_expiration\_date;

}

void SET\_quantity(int input\_quantily)

{

quantity = input\_quantily;

}

void SET\_price(float input\_price)

{

price = input\_price;

}

void print()

{

cout << "Name: " << GET\_name()

<< "\nManufacturer: " << GET\_manufacturer()

<< "\nPrice: " << GET\_price()

<< "\nExpiration\_date: " << GET\_expiration\_date()

<< "\nQuantity: " << GET\_quantity() << endl << endl;

}

};

int main()

{

float price = 0;

int quantity = 0;

int choice = 0;

int max = 40;

char\* name = new char[max];

char\* manufacturer = new char[max];

char\* expiration\_date = new char[max];

PRODUCT roduct1(name, manufacturer, expiration\_date, quantity, price);

cout << "Name: ";

cin >> name;

roduct1.SET\_name(name);

cout << "Manufacturer: ";

cin >> manufacturer;

roduct1.SET\_manufacturer(manufacturer);

cout << "Price: ";

cin >> price;

roduct1.SET\_price(price);

cout << "Expiration\_date: ";

cin >> expiration\_date;

roduct1.SET\_expiration\_date(expiration\_date);

cout << "Quantity: ";

cin >> quantity;

roduct1.SET\_quantity(quantity);

while (choice != 8)

{

cout << "\nInput name - 1\nInput manufacturer - 2\nInput price - 3\nInput expiration date - 4\nInput quantity - 5\nInput all information about product - 6\nPrint information about product - 7\nExit - 8\n" << endl;

cin >> choice;

if (choice == 1)

{

cout << "Name: ";

cin >> name;

roduct1.SET\_name(name);

}

else if (choice == 2)

{

cout << "Manufacturer: ";

cin >> manufacturer;

roduct1.SET\_manufacturer(manufacturer);

}

else if (choice == 3)

{

cout << "Price: ";

cin >> price;

roduct1.SET\_price(price);

}

else if (choice == 4)

{

cout << "Expiration\_date: ";

cin >> expiration\_date;

roduct1.SET\_expiration\_date(expiration\_date);

}

else if (choice == 5)

{

cout << "Quantity: ";

cin >> quantity;

roduct1.SET\_quantity(quantity);

}

else if (choice == 6)

{

cout << "Name: ";

cin >> name;

roduct1.SET\_name(name);

cout << "Manufacturer: ";

cin >> manufacturer;

roduct1.SET\_manufacturer(manufacturer);

cout << "Price: ";

cin >> price;

roduct1.SET\_price(price);

cout << "Expiration\_date: ";

cin >> expiration\_date;

roduct1.SET\_expiration\_date(expiration\_date);

cout << "Quantity: ";

cin >> quantity;

roduct1.SET\_quantity(quantity);

}

else if (choice == 7)

{

roduct1.print();

}

else if (choice == 8)

{

cout << "Exit from the program " << endl;

}

else

{

cout << "Incorrectly selected command.The number must be between 1 and 8" << endl;

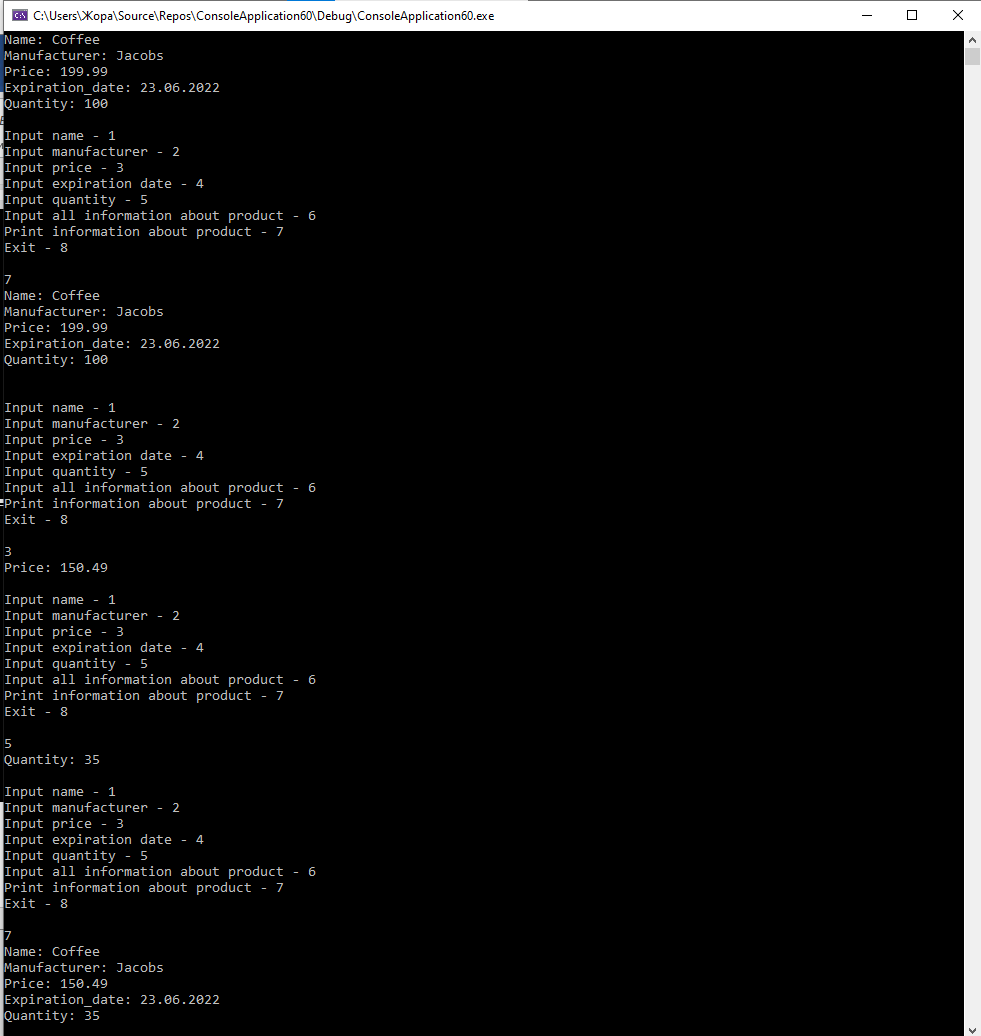
}

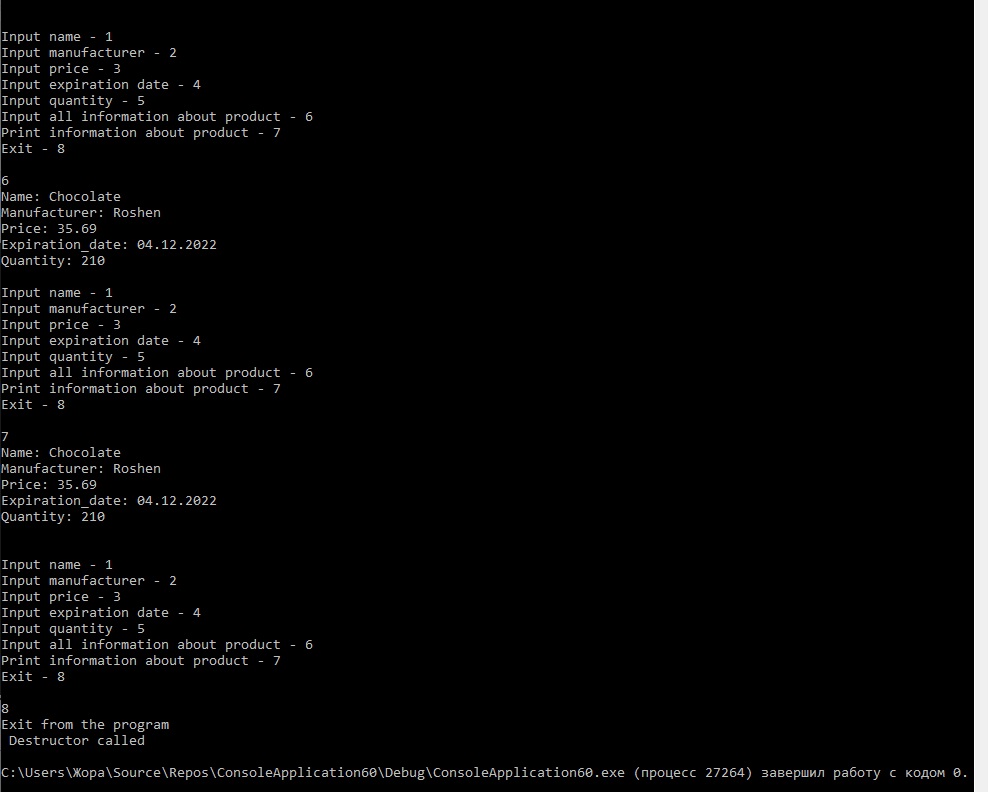
}

return 0;

}

**Результати:**

****

****

**Контрольні запитання:**

**1. Що таке клас, навіщо він потрібен?**

Клас — це спеціальна конструкція, яка використовується для групування пов'язаних змінних та функцій.

**2. Що таке об'єкт?**

Об'єкт - це програмний компонент, який зберігає дані та функції для роботи з цими даними. Об'єкти можуть відповідати об'єктам реального світу (товари у інтернет-магазині) або бути віртуальними об'єктами (для збереження тексту створюється об'єкт). Об'єктами можуть бути масиви, числа, логічні значення.

**3. Як пов'язані між собою класи і об'єкти в програмi?**

Класи й об'єкти розглядаються як додаткова мовна конструкція, що доповнює процедурне програмування. За допомогою об’єктно-орієнтованого програмування є можливість описати різні об’єкти, їх характеристики та властивості. Об’єкти описуються за допомогою класів. Об’єкт це сутність зі своїми властивостями та характеристиками, яка може реагувати на повідомлення за допомогою своїх існуючих даних та з’являється при створення екземпляра класу або прототипу. Клас – це список характеристик та можливостей об’єкта.

**4. Що таке інкапсуляція?**

Інкапсуляція — один з трьох основних механізмів об'єктно-орієнтованого програмування. Об’єкт вміщує не тільки дані, але і правила їх обробки, оформлені в вигляді виконуваних фрагментів (методів). А також про те, що доступ до стану об'єкта напряму заборонено, і ззовні з ним можна взаємодіяти виключно через заданий інтерфейс (відкриті поля та методи), що дозволяє знизити зв'язність.

**5. За рахунок чого реалізується захист від несанкціонованого доступу до даних?**

Захист від несанкціонованого доступу до даних реалізується через поля класу з назвою protected, privat

**6. Чим вiдрізняються поля від змiнних?**

Статичні поля семантично не відрізняються від звичайних глобальних змінних, але вони доступні тільки по кваліфікованому імені (тобто із зазначенням імені класу), і тому, на відміну від глобальних змінних, не захаращують простір глобальних імен

**7. Що визначають методи класу?**

Метод класу - це іменований блок виконуваного коду (набір операторів), який може бути викликаний на виконання з різних частин програми. Розрізняють прості методи і статичні методи (методи класу): прості методи мають доступ до даних об'єкта (конкретного екземпляра даного класу)

**8. Перелічити, чим можуть бути поля класу.**

Поля класу (змінна-член, data member, class field, instance variable) — змінна, зв'язана з класом або об'єктом. Всі дані об'єкта зберігаються в його полях. Доступ до полів здійснюється по імені.

**9. Що таке конструктор, павіщо він потрібен?**

Конструктор — пеціальний метод класу, який автоматично викликається при створенні об'єкта. та визначити інваріант класу. ... Деякі мови розрізняють кілька особливих типів конструкторів: Конструктор за умовчанням — конструктор, що не приймає аргументів, або всі аргументи якого мають значення за умовчанням.

**10. Що таке деструктор, нащо він потрібен?**

Деструктор-це спеціальний тип методу класу, який виконується при видаленні об'єкта класу. У той час як конструктори призначені для ініціалізації класу, деструктори призначені для очищення пам'яті після нього.