**Заклад вищої освіти**

**«МІЖНАРОДНИЙ НАУКОВО-ТЕХНІЧНИЙ УНІВЕРСИТЕТ ІМЕНІ АКАДЕМІКА Ю.БУГАЯ»**

Кафедра Комп’ютерних наук та інженерії програмного забезпечення

**Курсова робота з дисципліни**

**"Об’єктно-орієнтоване програмування"**

**на тему: Інтернет-магазин комп’ютерної техніки**

Виконав студент групи І21

Грабовик Катерина Вячеславівна

Перевірив:

Карпенко Сергій Грігорович

**2024 р**

**Зміст**

[**ВСТУП** 3](#_Toc165722400)

[**Розділ 1. Лямбда-вирази в C# і їх застосування** 4](#_Toc165722401)

[**Розділ 2. Практична частина.** **Інтернет-магазин комп’ютерної техніки** 7](#_Toc165722402)

[**2.1. Постановка задачі** 7](#_Toc165722403)

[**2.2. Розробка ієрархії класів** 8](#_Toc165722404)

[**2.3. Розробка алгоритму рішення задачі** 8](#_Toc165722405)

[**2.3.1 Зберігання інформації** 8](#_Toc165722406)

[**2.3.2 Перевизначені методи** 8](#_Toc165722407)

[**2.3.3 Реалізація функцій** 8](#_Toc165722408)

[**2.4. Розробка програми рішення задачі** 9](#_Toc165722409)

[**2.4.1 База даних** 9](#_Toc165722410)

[**2.4.2 Базовий клас** 10](#_Toc165722411)

[**2.4.3 Похідні класи** 10](#_Toc165722412)

[**2.4.4 Створення таблиць** 11](#_Toc165722413)

[**2.4.5 Основний код** 12](#_Toc165722414)

[**2.4.6 Windows Forms** 22](#_Toc165722415)

[**ВИСНОВКИ** 33](#_Toc165722416)

[**РЕСУРСИ МЕРЕЖІ ІNTЕRNЕT** 34](#_Toc165722417)

# **ВСТУП**

Відповідно до навчального плану вивчення дисципліни «Об‘єктно-

орієнтоване програмування» у 4 семестрі передбачається виконання

студентами курсової роботи.

У розробленій курсовій роботі студент повинен показати знання:

- основних елементів предметної області відповідно до постановки завдання;

- основних концепцій об‘єктно-орієнтованого програмування;

- сучасних інструментальних засобів, призначених для розроблення

об‘єктно-орієнтованих програм.

Студент повинен показати вміння з:

- аналізу постановки завдання;

- декомпозиції програмної системи на складові;

- розроблення загальної архітектури програмної системи;

- деталізації загальної архітектури програмної системи у термінах об‘єктно-орієнтованого програмування;

- програмної реалізації системи, що розробляється, на мові програмування.

# **Розділ 1. Лямбда-вирази в C# і їх застосування**

Лямбда-вирази є компактним способом створення анонімних методів. Вони покращують читаність коду, зменшують обсяг написаного коду та дозволяють передавати функціональність у більш лаконічній формі. Лямбда-вирази використовуються у різних сценаріях, включаючи LINQ-запити, обробники подій, а також функціональне програмування загалом.

В усіх лямбда-виразах використовується лямбда-оператор ***=>***, який читається як «переходить в». Ліва частина лямбда-оператора визначає параметри введення (якщо такі є), а права частина містить вираз або блок оператора. Приклад:

**namespace** **Lab9\_1**

{

**internal** **class** **Program**

{

**delegate** **double** **M**(**double** x);

**static** **void** **Main**(**string**[] args)

{

Console.OutputEncoding = Encoding.Unicode;

Console.Write("Введіть число: ");

**double** x=Convert.ToDouble(Console.ReadLine());

**if** (x > **0**) { M r = x => Math.Pow(Math.Sin(x), **2**) + **1**; Console.WriteLine("F(x)= sin^2(x)+1 = " + r(x)); }

**if** (x <= **0**) { M r = x => **1** - **2** \* Math.Sin(x); Console.WriteLine("F(x)= 1-2\*sin(x) = " + r(x)); }

}

}

}

Спочатку в класі Program створюємо делегат, що приймає числовий параметр. В функції Main створюємо екземпляр делегату, і спочатку знаком = призначаємо значення, а далі лямбда-оператором використовуємо це значення у виразі.

В .NET існують вбудовані універсальні делегати, наприклад **Predicate**, який використовується лише для порівняння і приймає один параметр. В результаті повертає значення true або false. Приклад:

**namespace** **Lab9\_3**

{

**internal** **class** **Program**

{

**static** **void** **Main**(**string**[] args)

{

Console.OutputEncoding = Encoding.Unicode;

Console.Write("Визначення рівності 2х чисел.True - рівні,False - різні.\nx=");

**double** x=Convert.ToDouble(Console.ReadLine());

Console.Write("y=");

**double** y = Convert.ToDouble(Console.ReadLine());

Predicate<**double**> IsEqual = x => x == y;

**bool** res = IsEqual(x);

Console.WriteLine("x та y "+res);

}

}

}

Делегат **Func** може приймати до 16 параметрів, та повертає результат будь якої дії:

**namespace** **Lab9\_2**

{

**internal** **class** **Program**

{

**static** **void** **Main**(**string**[] args)

{

Func<**double**,**double**,**double**,**double**> res = (x,x2,x3) => x\*x2\*x3;

Console.OutputEncoding = Encoding.Unicode;

Console.Write("Введіть x1: ");

**double** x=Convert.ToDouble(Console.ReadLine());

Console.Write("Введіть x2: ");

**double** x2 = Convert.ToDouble(Console.ReadLine());

Console.Write("Введіть x3: ");

**double** x3 = Convert.ToDouble(Console.ReadLine());

Console.WriteLine("{0}\*{1}\*{2} = {3}",x,x2,x3,res(x,x2,x3));

}

}

}

Делегат приймає значення 3х чисел та повертає їх добуток.

Також **Func** можна використовувати з блоковими лямбда-виразами:

**namespace** **Lab9\_4**

{

**internal** **class** **Program**

{

**static** **void** **Main**(**string**[] args)

{

Console.OutputEncoding = Encoding.Unicode;

**int** count;

Console.Write("Максимальна к-ть чисел в масиві: ");

**int** x=Convert.ToInt32(Console.ReadLine());

Console.Write("Число що треба знайти: ");

**int** y = Convert.ToInt32(Console.ReadLine());

**int**[] array =**new** **int**[x];

Random random= **new** Random();

**for**(**int** i=**0**;i<x;i++)

{

array[i] = random.Next(**1**,x);

}

Console.Write("Array=[ ");

**for** (**int** i = **0**; i < x; i++)

{

Console.Write(array[i]+",");

}

Console.WriteLine("];");

Func<**int**,**int**> Find = y =>

{

count = **0**;

**for**(**int** i=**0**;i< array.Length;i++)

{

**if** (y == array[i]) count =count+ **1**;

}

**return** count;

};

**int** res = Find(y);

Console.WriteLine("Всього в масиві така к-ть числа {0}: {1}.",y,res);

}

}

}

# **Розділ 2. Практична частина.** **Інтернет-магазин комп’ютерної техніки**

## **2.1. Постановка задачі**

Мета: Розробити програму для вибору комп’ютерної техніки. В магазині є

техніка лише двох типів: ПК і ноутбуки. Реалізувати 2 версії програми (консольну і з інтерфейсом WindowsForms).

Предметна область: ком’ютерна техніка. Дано 2 типи : ПК та ноутбуки.

Для опису ПК у файлі потрібно зберігати таку інформацію:

- ID ПК;

- модель;

- виробник;

- корпус;

- об’єм RAM;

- жорсткий диск;

- кількість ядер процесора;

- вартість;

- наявність на складі.

Для опису ноутбуків у файлі потрібно зберігати таку інформацію:

- ID ноутбука;

- виробник;

- діагональ екрану;

- об’єм RAM;

- жорсткий диск;

- кількість ядер процесора;

- динаміки;

- вартість;

- наявність на складі.

## **2.2. Розробка ієрархії класів**

Створюємо базовий клас DataClasses, В якому зберігається спільна інформація про обидва види техніки, а також 3 методи, що будуть перевизначені в похідних класах PC і LapTop. Ці класи наслідують всі елементи базового, а також мають 2 свої власні.

## **2.3. Розробка алгоритму рішення задачі**

### **2.3.1 Зберігання інформації**

Для зберігання даних я скористалася таблицями SQL Server, вбудованих в VS.

Зберігання відбувається за такою структурою:

ID, Модель(string)/Діагональ(double), Корпус(string)/Динаміки(string), Виробник(string), RAM(int), жорсткий диск(int), кількість ядер(int), ціна(double), наявність на складі(bool).

Я створювала 2 окремі таблиці даних.

### **2.3.2 Перевизначені методи**

Перший метод використовується в обидвох похідних класах. Він використовує для виводу елементів в консольному додатку.

Другий метод також використовується лише в консольному додатку, та повертає речення, якщо ціна ПК більша 40 000 грн.

Третій розраховує ціну ноутбука зі знижкою. Користувач вписує % знижки. Перед викликом методу, це число ділиться на 100, метод приймає це значення і віднімає від ціни добуток ціни і вхідного параметра.

### **2.3.3 Реалізація функцій**

-додавання, видалення техніки;

- редагування даних про ціну;

- перегляд моделей ПК, впорядкованих за ціною в порядку зростання;

- відбір тільки ноутбуків;

- вибір тільки ноутбуків вказаної фірми (наприклад, ASUS);

- пошук ПК за об’ємом диску;

- пошук ноутбуків, які є на складі;

- обчислення вартості ноутбуку зі знижкою (перевизначений метод);

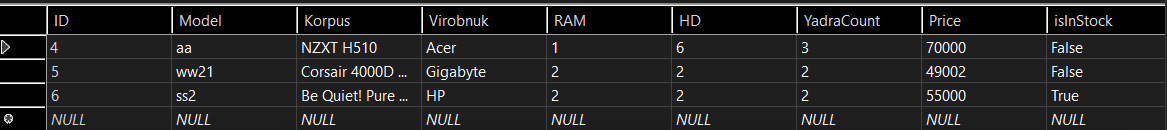
- якщо вартість ПК більше 40 тис. гривень, мишка у подарунок (перевизначений метод).

Багато функцій потребують взаємодію з користувачем, тож я створила 5 методів:

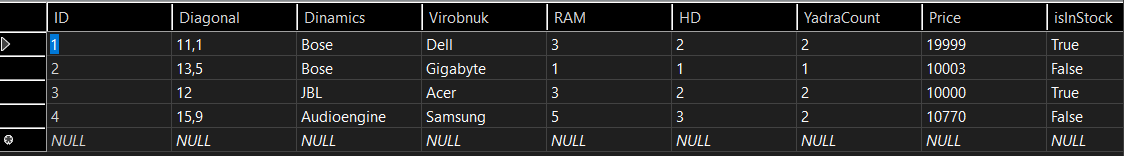
1. М() – виконує роль головного меню, що має посилання на інші 4 методи.
2. Dodati() – виконує роль додавання елементів до бази даних.
3. Vidal() – видалення елементів за ID.
4. Redact() – редагування ціни.
5. Show() – найбільший за функціоналом метод. Він показує усю техніку, окремо ПК та ноутбуки з власними функціями відбору.

## **2.4. Розробка програми рішення задачі**

### **2.4.1 База даних**



*Рис1.Таблиця ПК*



*Рис2. Таблиця ноутбуків*

### **2.4.2 Базовий клас**

**public** **class** **DataClasses**

{

**public** **int** ID { **get**; **set**; }// айді

**public** **string** Virobnuk { **get**; **set**; }//Виробник

**public** **int** RAM { **get**; **set**; }//RAM

**public** **int** HD { **get**; **set**; }//жорсткий диск

**public** **int** YadraCount { **get**; **set**; }//кількість ядер

**public** **double** Price { **get**; **set**; }//ціня

**public** **bool** isInStock { **get**; **set**; }//статус

**public** **DataClasses**(**string** virobnuk, **int** rAM, **int** hD, **int** yadraCount, **double** price, **bool** isInStock)

{

Virobnuk = virobnuk;

RAM = rAM;

HD = hD;

YadraCount = yadraCount;

Price = price;

**this**.isInStock = isInStock;

}

**public** **virtual** **double** **Znijka**(**double** zn) => -**1**;//знижка

**public** **virtual** **string** **Podarok**() => "";//подарунок

**public** **override** **string** **ToString**() => $"ID:{ID}; Виробник:{Virobnuk};Об'єм RAM:{RAM};Об'єм жорского диску:{HD};Кількість ядер:{YadraCount};Ціна:[{Price}]грн;";//для виводу інформацї

}

### **2.4.3 Похідні класи**

**public** **class** **PC** : DataClasses

{ **public** **string** Model { **get**; **set**; }//модель

**public** **string** Korpus { **get**; **set**; }//корпус

**public** **PC**(**string** virobnuk, **int** rAM, **int** hD, **int** yadraCount, **double** price, **bool** isInStock, **string** model, **string** korpus)

: **base**(virobnuk, rAM, hD, yadraCount, price, isInStock)

{Model = model;

Korpus = korpus; }

**public** **override** **string** **Podarok**() => "\*\*\*}Ціна ПК перевищує 40000. Вам надано подарунок у вигляді мишки.{\*\*\*";

**public** **override** **string** **ToString**() => **base**.ToString() + $"Корпус:{Korpus};Модель:{Model}.";

}

**public** **class** **LapTop** : DataClasses

{

**public** **double** Diagonal { **get**; **set**; }//діагональ

**public** **string** Dinamics { **get**; **set**; }//динаміки

**public** **LapTop**(**string** virobnuk, **int** rAM, **int** hD, **int** yadraCount, **double** price, **bool** isInStock, **double** diagonal, **string** dinamics)

: **base**(virobnuk, rAM, hD, yadraCount, price, isInStock)

{

Diagonal = diagonal;

Dinamics = dinamics;

}

**public** **override** **double** **Znijka**(**double** zn) => Price -= Price \* zn;

**public** **override** **string** **ToString**() => **base**.ToString() + $"Діагональ:{Diagonal};Динаміки:{Dinamics}.";

}

### **2.4.4 Створення таблиць**

**namespace** **KursovaConsole.DB**

{

**public** **class** **DDBContext**:DbContext

{

**public** DbSet<PC> PCs { **get**; **set**; }

**protected** **override** **void** **OnConfiguring**(DbContextOptionsBuilder optionsBuilder)

{

optionsBuilder.UseSqlServer(@"Server=(localdb)\mssqllocaldb;Database=PCdataConsole;Trusted\_Connection=True;");

}

}

**public** **class** **DDBContext2** : DbContext

{

**public** DbSet<LapTop> LTs { **get**; **set**; }

**protected** **override** **void** **OnConfiguring**(DbContextOptionsBuilder optionsBuilder)

{

optionsBuilder.UseSqlServer(@"Server=(localdb)\mssqllocaldb;Database=LapTopdataConsole;Trusted\_Connection=True;");

}

}

**public** **class** **DDBContext3** : DbContext

{

**public** DbSet<PC> PCs { **get**; **set**; }

**protected** **override** **void** **OnConfiguring**(DbContextOptionsBuilder optionsBuilder)

{

optionsBuilder.UseSqlServer(@"Server=(localdb)\mssqllocaldb;Database=PCdata;Trusted\_Connection=True;");

}

}

**public** **class** **DDBContext4** : DbContext

{

**public** DbSet<LapTop> LTs { **get**; **set**; }

**protected** **override** **void** **OnConfiguring**(DbContextOptionsBuilder optionsBuilder)

{

optionsBuilder.UseSqlServer(@"Server=(localdb)\mssqllocaldb;Database=LapTopdata;Trusted\_Connection=True;");

}

}

}

### **2.4.5 Основний код**

**namespace** **KursovaConsole**

{

**internal** **class** **Program**

{

**static** List<PC> pcs = **new** List<PC>();

**static** List<LapTop> lapTops = **new** List<LapTop>();

**static** **void** **Main**(**string**[] args)

{

Console.OutputEncoding = Encoding.UTF8;

Console.InputEncoding = Encoding.UTF8;

Console.WriteLine("При першому включенні програми дії завантажуються повільно. Почекайте, доки все завантжиться після натиску.");

M();

}

//Головне меню

**static** **void** **M**()

{

Console.WriteLine("Для виконання дії введіть відповідну цифру:\n1-додати,\n2-видалити,\n3-показати,\n4-редагувати.\n5-вийти.\nІнакша відповідь буде вважатися помилкою.");

**string** vidp1 = Console.ReadLine();

**switch** (vidp1)

{

**case** "1": Dodati(); **break**;

**case** "2": Vidal(); **break**;

**case** "3":

Console.Clear();

Show();

**break**;

**case** "4": Redact(); **break**;

**case** "5": Console.WriteLine("---/Закриття програми\\---"); **break**;

**default**:

Console.WriteLine("Нічого не обрано/Некоректний вибір. Повернення на головне меню - будь яка кнопка"); Console.ReadLine();

Console.Clear();

M(); **break**;

}

}

//Показ

**static** **void** **Show**()

{

**try**

{

Console.WriteLine("Опції:\n1-всі елементи,\n2-ПК,\n3-ноутбуки,\n4-На головний екран.");

**string** res = Console.ReadLine();

pcs.Clear();

lapTops.Clear();

**var** db1 = **new** DDBContext();

db1.Database.EnsureCreated();

db1.PCs.Load();

**foreach** (**var** pl **in** db1.PCs.Local.ToList()) { pcs.Add(pl); }

**var** db2 = **new** DDBContext2();

db2.Database.EnsureCreated();

db2.LTs.Load();

**foreach** (**var** pl **in** db2.LTs.Local.ToList()) { lapTops.Add(pl); }

**switch** (res)

{

**case** "1":

Console.Clear();

Console.WriteLine("-------------------------|Вся електротехніка|-------------------------");

**foreach** (**var** l **in** pcs)

{

Console.WriteLine(l);

Console.WriteLine("-------------------------");

}

**foreach** (**var** l **in** lapTops)

{

Console.WriteLine(l);

Console.WriteLine("-------------------------");

}

Show();

**break**;//Все

**case** "2":

**if** (pcs.IsNullOrEmpty())

{

Console.Write("Список порожній,тому неможливо виконати додаткові дії.Тисніть будь що для повернення на головне меню."); Console.ReadLine();

Console.Clear();

M();

}

**else**

{

Console.Clear();

Console.WriteLine("-------------------------|Персональні комп'ютери|-------------------------");

**foreach** (**var** l **in** pcs)

{

Console.WriteLine(l);

**if** (l.Price > **40000**) { Console.WriteLine(l.Podarok()); }

Console.WriteLine("-------------------------");

}

Console.WriteLine("Опції:\n1-Пошук за об'ємом на диску,\n2-Сортування за ціною,\n3-Меню вибору,\n4-На головний екран.");

**string** PCOnly = Console.ReadLine();

**switch** (PCOnly)

{

**case** "1":

Console.Write("Вкажіть об'єм диску: ");

**int** V = **int**.Parse(Console.ReadLine());

Console.Clear();

Console.WriteLine($"-------------------------|Персональні комп'ютери HD = {V}|-------------------------");

**foreach** (**var** l **in** pcs)

{

**if** (l.HD == V)

{

Console.WriteLine(l);

Console.WriteLine("-------------------------");

}

}

Show();

**break**;

**case** "2":

Console.Clear();

Console.WriteLine($"-------------------------|Персональні комп'ютери відсортовані по ціні в порядку зростання|-------------------------");

IEnumerable<PC> pcc = pcs.OrderBy(pc => pc.Price);

**foreach** (**var** l **in** pcc) { Console.WriteLine(l); Console.WriteLine("-------------------------"); }

Show();

**break**;

**case** "3":

Console.Clear();

Show();

**break**;//Меню вибору

**case** "4":

Console.Clear();

M();

**break**;//Головне меню

**default**:

Console.WriteLine("Нічого не обрано/Некоректний вибір. Повернення на головне меню - будь яка кнопка"); Console.ReadLine(); Console.Clear();

M(); **break**;

}

}

**break**;//Тільки ПК

**case** "3":

**if** (lapTops.IsNullOrEmpty())

{

Console.Write("Список порожній,тому неможливо виконати додаткові дії.Тисніть будь що для повернення на головне меню."); Console.ReadLine();

Console.Clear();

M();

}

**else**

{

Console.Clear();

Console.WriteLine("-------------------------|Ноутбуки|-------------------------");

**foreach** (**var** l **in** lapTops)

{

Console.WriteLine(l);

Console.WriteLine("-------------------------");

}

Console.WriteLine("Опції:\n1-Використати промокод,\n2-Показати за певними критеріями,\n3-Меню вибору,\n4-На головний екран.");

**string** LapOnly = Console.ReadLine();

**switch** (LapOnly)

{

**case** "1":

Console.Write("Оберіть елемент, ввівши його ID: ");

**int** id = **int**.Parse(Console.ReadLine());

**foreach** (**var** l **in** lapTops)

{

**if** (l.ID == id)

{

Console.Write("Введіть знижку (у відсодках %): ");

**double** ProCode = **double**.Parse(Console.ReadLine());

Console.Clear();

Console.WriteLine($"-------------------------|Ноутбук {l.ID} Зі знижкою {ProCode}%|-------------------------");

ProCode = ProCode / **100**;

l.Znijka(ProCode);

Console.WriteLine(l);

}

**else**

{

Console.Write("Такого елемента не знайдено. Повернення на Меню вибору - будь яка кнопка"); Console.ReadLine();

Console.Clear();

Show();

}

}

Show();

**break**;//Промокод

**case** "2":

Console.WriteLine("Опції:\n1-Вибір за виробником,\n2-Наявність на складі.");

**string** optionsLap = Console.ReadLine();

**switch** (optionsLap)

{

**case** "1":

Console.Write("Впишіть назву виробника: ");

**string** virLap = Console.ReadLine();

Console.Clear();

Console.WriteLine($"-------------------------|Ноутбуки фірми {virLap}|-------------------------");

**foreach** (**var** l **in** lapTops)

{

**if** (l.Virobnuk == virLap)

{

Console.WriteLine(l);

}

}

Show();

**break**;//Виробник

**case** "2":

Console.Clear();

Console.WriteLine($"-------------------------|Ноутбуки в наявності|-------------------------");

**foreach** (**var** l **in** lapTops)

{

**if** (l.isInStock == **true**) { Console.WriteLine(l); } **else** { Console.WriteLine($"ID={l.ID} ;відсутній."); }

Console.WriteLine("-------------------------");

}

Show();

**break**;//В наявності

**default**:

Console.WriteLine("Нічого не обрано/Некоректний вибір. Повернення на головне меню - будь яка кнопка"); Console.ReadLine();

M(); **break**;

}

**break**;//Критерії

**case** "3":

Console.Clear();

Show();

**break**;//Меню вибору

**case** "4":

Console.Clear();

M();

**break**;//Головне меню

**default**:

Console.WriteLine("Нічого не обрано/Некоректний вибір. Повернення на головне меню - будь яка кнопка"); Console.ReadLine();

Console.Clear();

M(); **break**;

}

}

**break**;//Тільки ноутбуки

**case** "4":

Console.Clear();

M();

**break**;//Головне меню

**default**:

Console.WriteLine("Нічого не обрано/Некоректний вибір. Повернення на головне меню - будь яка кнопка"); Console.ReadLine();

Console.Clear();

M(); **break**;

}

}

**catch** (FormatException)

{

Console.Write("Введено неправильний формат.Тисніть будь що, щоб повернутись в меню.");

Console.ReadLine();

Console.Clear();

Show();

}

}

//Додавання

**static** **void** **Dodati**()

{

**try**

{

Console.Clear();

Console.WriteLine("-------------------------|Додавання|-------------------------");

Console.Write("Виробник: ");

**string** Vir = Console.ReadLine();

Console.Write("об'єм RAM: ");

**int** RAM = **int**.Parse(Console.ReadLine());

Console.Write("об'єм HD: ");

**int** HD = **int**.Parse(Console.ReadLine());

Console.Write("Кількість ядер: ");

**int** YadCount = **int**.Parse(Console.ReadLine());

Console.Write("Ціна: ");

**double** Price = **double**.Parse(Console.ReadLine());

Console.Write("Якщо в наявності, пишіть 1,якщо немає - тисніть будь яку кнопку: ");

**bool** isAwaible;

**if** (Console.ReadLine() == "1")

{

isAwaible = **true**;

}

**else**

{

isAwaible = **false**;

}

Console.WriteLine("Вкажіть тип.\n1 - ПК,\n2 - ноутбук");

**string** res = Console.ReadLine();

**switch** (res)

{

**case** "1":

Console.Write("Модель: ");

**string** Model = Console.ReadLine();

Console.Write("Корпус: ");

**string** Korpus = Console.ReadLine();

**var** db1 = **new** DDBContext();

db1.Database.EnsureCreated();

db1.PCs.Load();

db1.Add(**new** PC(Vir, RAM, HD, YadCount, Price, isAwaible, Model, Korpus));

db1.SaveChanges();

**foreach** (**var** pl **in** db1.PCs.Local.ToList())

{

pcs.Add(pl);

}

**break**;

**case** "2":

Console.Write("Діагональ: ");

**double** Diagonal = **double**.Parse(Console.ReadLine());

Console.Write("Динаміки: ");

**string** Dinamics = Console.ReadLine();

**var** db2 = **new** DDBContext2();

db2.Database.EnsureCreated();

db2.LTs.Load();

db2.Add(**new** LapTop(Vir, RAM, HD, YadCount, Price, isAwaible, Diagonal, Dinamics));

db2.SaveChanges();

**foreach** (**var** pl **in** db2.LTs.Local.ToList())

{

lapTops.Add(pl);

}

**break**;

**default**:

Console.WriteLine("Нічого не обрано/Некоректний вибір. Повернення на головне меню - будь яка кнопка"); Console.ReadLine();

Console.Clear();

M(); **break**;

}

Console.Clear();

Console.WriteLine("Об'єкт додано");

M();

}

**catch** (FormatException)

{

Console.Write("Введено неправильний формат.Тисніть будь що, щоб почати заново.");

Console.ReadLine();

Console.Clear();

Dodati();

}

}

//Видалення

**static** **void** **Vidal**()

{

**try**

{

Console.Clear();

pcs.Clear();

lapTops.Clear();

**var** db1 = **new** DDBContext();

db1.Database.EnsureCreated();

db1.PCs.Load();

**foreach** (**var** pl **in** db1.PCs.Local.ToList()) { pcs.Add(pl); }

**var** db2 = **new** DDBContext2();

db2.Database.EnsureCreated();

db2.LTs.Load();

**foreach** (**var** pl **in** db2.LTs.Local.ToList()) { lapTops.Add(pl); }

Console.WriteLine("Оберіть тип техніки.\n1-ПК,\n2-ноутбуки.");

**string** res = Console.ReadLine();

**switch** (res)

{

**case** "1":

**if** (pcs.IsNullOrEmpty())

{

Console.Write("Список порожній.Дію виконати неможливо.Тисніть будь що для повернення на головне меню."); Console.ReadLine();

Console.Clear();

M();

}

**else**

{

**foreach** (**var** l **in** pcs)

{

Console.WriteLine(l);

Console.WriteLine("-------------------------");

}

Console.Write("Введіть ID для видалення: ");

**int** id = **int**.Parse(Console.ReadLine());

**foreach** (**var** pl **in** db1.PCs.Local.ToList())

{

**if** (pl.ID == id)

{

db1.Remove(pl);

db1.SaveChanges();

**break**;

}

**else**

{

Console.Write("Такого елемента не знайдено. Повернення на головне меню - будь яка кнопка"); Console.ReadLine();

Console.Clear();

M();

}

}

}

**break**;

**case** "2":

**if** (lapTops.IsNullOrEmpty())

{

Console.Write("Список порожній.Дію виконати неможливо.Тисніть будь що для повернення на головне меню."); Console.ReadLine();

Console.Clear();

M();

}

**else**

{

**foreach** (**var** l **in** lapTops)

{

Console.WriteLine(l);

Console.WriteLine("-------------------------");

}

Console.Write("Введіть ID для видалення: ");

**int** idi = **int**.Parse(Console.ReadLine());

**foreach** (**var** pl **in** db2.LTs.Local.ToList())

{

**if** (pl.ID == idi)

{

db2.Remove(pl);

db2.SaveChanges();

**break**;

}

**else**

{

Console.Write("Такого елемента не знайдено. Повернення на головне меню - будь яка кнопка"); Console.ReadLine();

Console.Clear();

M();

}

}

}

**break**;

**default**:

Console.Write("Нічого не обрано/Некоректний вибір. Повернення на головне меню - будь яка кнопка"); Console.ReadLine();

Console.Clear();

M(); **break**;

}

Console.Clear();

Console.WriteLine("Об'єкт Видалено");

M();

}

**catch** (FormatException)

{

Console.Write("Введено неправильний формат.Тисніть будь що, щоб Повернутися в головне меню.");

Console.ReadLine();

Console.Clear();

M();

}

}

//Редагування

**static** **void** **Redact**()

{

Console.Clear();

pcs.Clear();

lapTops.Clear();

**var** db1 = **new** DDBContext();

db1.Database.EnsureCreated();

db1.PCs.Load();

**foreach** (**var** pl **in** db1.PCs.Local.ToList()) { pcs.Add(pl); }

**var** db2 = **new** DDBContext2();

db2.Database.EnsureCreated();

db2.LTs.Load();

**foreach** (**var** pl **in** db2.LTs.Local.ToList()) { lapTops.Add(pl); }

Console.WriteLine("Оберіть тип техніки.\n1-ПК,\n2-ноутбуки.");

**string** res = Console.ReadLine();

**switch** (res)

{

**case** "1":

**if** (pcs.IsNullOrEmpty())

{

Console.Write("Список порожній.Дію виконати неможливо.Тисніть будь що для повернення на головне меню."); Console.ReadLine();

Console.Clear();

M();

}

**else**

{

**foreach** (**var** l **in** pcs)

{

Console.WriteLine(l);

Console.WriteLine("-------------------------");

}

Console.Write("Введіть ID для редагування ціни: ");

**int** id = **int**.Parse(Console.ReadLine());

**foreach** (**var** pli **in** db1.PCs.Local.ToList())

{

**if** (pli.ID == id)

{

Console.WriteLine("Стара ціна: " + pli.Price);

**break**;

}

**else**

{

Console.Write("Такого елемента не знайдено. Повернення на головне меню - будь яка кнопка"); Console.ReadLine();

Console.Clear();

M();

}

}

Console.Write("Введіть нову ціну: ");

**double** nPPrice = **double**.Parse(Console.ReadLine());

**foreach** (**var** pli **in** db1.PCs.Local.ToList())

{

**if** (pli.ID == id)

{

pli.Price = nPPrice;

db1.SaveChanges();

}

}

}

**break**;

**case** "2":

**if** (lapTops.IsNullOrEmpty())

{

Console.Write("Список порожній.Дію виконати неможливо.Тисніть будь що для повернення на головне меню."); Console.ReadLine();

Console.Clear();

M();

}

**else**

{

**foreach** (**var** l **in** lapTops)

{

Console.WriteLine(l);

Console.WriteLine("-------------------------");

}

Console.Write("Введіть ID для редагування ціни: ");

**int** idi = **int**.Parse(Console.ReadLine());

**foreach** (**var** pli **in** db2.LTs.Local.ToList())

{

**if** (pli.ID == idi)

{

Console.WriteLine("Стара ціна: " + pli.Price);

**break**;

}

**else**

{

Console.Write("Такого елемента не знайдено. Повернення на головне меню - будь яка кнопка"); Console.ReadLine();

Console.Clear();

M();

}

}

Console.Write("Введіть нову ціну: ");

**double** nPrice = **double**.Parse(Console.ReadLine());

**foreach** (**var** pli **in** db2.LTs.Local.ToList())

{

**if** (pli.ID == idi)

{

pli.Price = nPrice;

db2.SaveChanges();

}

}

}

**break**;

**default**:

Console.WriteLine("Нічого не обрано/Некоректний вибір. Повернення на головне меню - будь яка кнопка"); Console.ReadLine();

Console.Clear();

M(); **break**;

}

Console.Clear();

Console.WriteLine("Ціну змінено");

M();

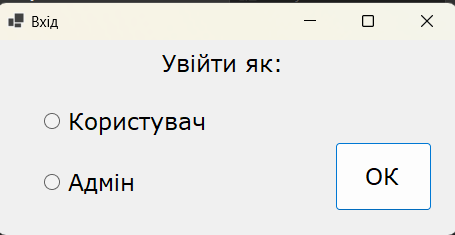
}

}

}

### **2.4.6 Windows Forms**

Форма входу:



Код:

**public** **partial** **class** **Form1** : Form

{

**public** **Form1**()

{

InitializeComponent();

}

**private** **void** **button1\_Click**(**object** sender, EventArgs e)

{

**if**(radioButton1.Checked==**true**){ User us = **new** User();us.ShowDialog(); }

**if** (radioButton2.Checked == **true**)

{

/\* if (textBox1.Text == "") { MessageBox.Show("Не введено пароль."); }

else if (textBox1.Text != "123") { MessageBox.Show("Неправильний пароль.");textBox1.Text = ""; }

else { }\*/

Admin ad = **new** Admin(); ad.ShowDialog();

}

}

**private** **void** **radioButton1\_CheckedChanged**(**object** sender, EventArgs e)

{

textBox1.Visible = **false**;

}

**private** **void** **radioButton2\_CheckedChanged**(**object** sender, EventArgs e)

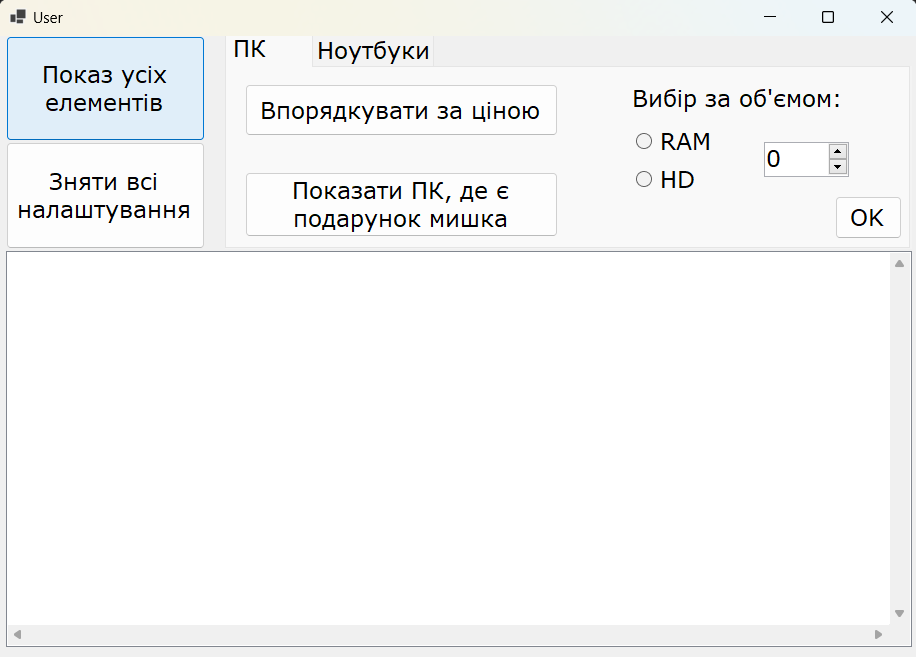
{

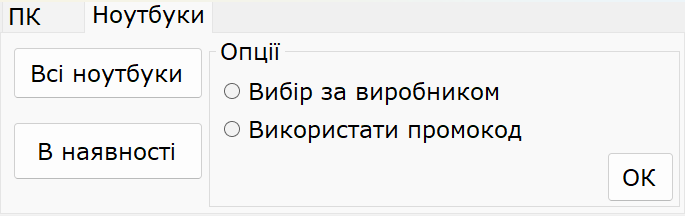
textBox1.Visible=**true**;

}

}

Форма Користувача:





Код:

**namespace** **KursovaWinForms**

{

**public** **partial** **class** **User** : Form

{

DDBContext3 dbPC;

DDBContext4 dbLT;

**public** **User**()

{

InitializeComponent();

dbPC = **new** DDBContext3();

dbLT = **new** DDBContext4();

dbPC.Database.EnsureCreated();

dbLT.Database.EnsureCreated();

dbPC.PCs.Load();

dbLT.LTs.Load();

}

//показ

**private** **void** **button1\_Click**(**object** sender, EventArgs e)

{

listBox1.Items.Clear();

**foreach** (**var** item **in** dbPC.PCs.Local.ToList())

{ listBox1.Items.Add($"Ціна:[{item.Price}]грн,Виробник:{item.Virobnuk};Об'єм RAM:{item.RAM};Об'єм жорского диску:{item.HD};Кількість ядер:{item.YadraCount},Корпус:{item.Korpus},Модель:{item.Model}."); }

**foreach** (**var** item **in** dbLT.LTs.Local.ToList())

{ listBox1.Items.Add($"Ціна:[{item.Price}]грн,Виробник:{item.Virobnuk};Об'єм RAM:{item.RAM};Об'єм жорского диску:{item.HD};Кількість ядер:{item.YadraCount},Діагональ:{item.Diagonal},Динаміки:{item.Dinamics}."); }

}

//Очистити

**private** **void** **button2\_Click**(**object** sender, EventArgs e)

{

button8.Enabled = **true**;

numericUpDown1.Value = **0**;

radioButton1.Checked = **false**;

radioButton2.Checked = **false**;

radioButton3.Checked = **false**;

radioButton4.Checked = **false**;

numericUpDown2.Visible = **false**;

label2.Visible = **false**;

label3.Visible = **false**;

numericUpDown3.Visible = **false**;

numericUpDown3.Value = **0**;

comboBox1.Visible = **false**;

listBox1.Items.Clear();

}

//---------------ПК----------------

//За ціною

**private** **void** **button3\_Click**(**object** sender, EventArgs e)

{

radioButton1.Checked = **false**;

radioButton2.Checked = **false**;

listBox1.Items.Clear();

**foreach** (**var** item **in** dbPC.PCs.Local.ToList())

{ listBox1.Items.Add($"Ціна:[{item.Price}]грн,Виробник:{item.Virobnuk};Об'єм RAM:{item.RAM};Об'єм жорского диску:{item.HD};Кількість ядер:{item.YadraCount},Корпус:{item.Korpus},Модель:{item.Model}."); }

listBox1.Sorted = **true**;

}

//Мишка

**private** **void** **button4\_Click**(**object** sender, EventArgs e)

{

radioButton1.Checked = **false**;

radioButton2.Checked = **false**;

listBox1.Sorted = **false**;

listBox1.Items.Clear();

**foreach** (**var** item **in** dbPC.PCs.Local.ToList())

{

**if** (item.Price > **40000**)

{

listBox1.Items.Add($"Ціна:[{item.Price}]грн,Виробник:{item.Virobnuk};Об'єм RAM:{item.RAM};Об'єм жорского диску:{item.HD};Кількість ядер:{item.YadraCount},Корпус:{item.Korpus},Модель:{item.Model}.");

}

}

}

//За об'ємом

**private** **void** **button5\_Click**(**object** sender, EventArgs e)

{

listBox1.Sorted = **false**;

listBox1.Items.Clear();

**if** (radioButton1.Checked == **true**)

{

**foreach** (**var** item **in** dbPC.PCs.Local.ToList())

{

**if** (item.RAM == numericUpDown1.Value)

{

listBox1.Items.Add($"Об'єм RAM:{item.RAM};Ціна:[{item.Price}]грн,Виробник:{item.Virobnuk};Об'єм жорского диску:{item.HD};Кількість ядер:{item.YadraCount},Корпус:{item.Korpus},Модель:{item.Model}.");

}

}

}

**if** (radioButton2.Checked == **true**)

{

**foreach** (**var** item **in** dbPC.PCs.Local.ToList())

{

**if** (item.HD == numericUpDown1.Value)

{

{ listBox1.Items.Add($"Об'єм жорского диску:{item.HD};Ціна:[{item.Price}]грн,Виробник:{item.Virobnuk};Об'єм RAM:{item.RAM};Кількість ядер:{item.YadraCount},Корпус:{item.Korpus},Модель:{item.Model}."); }

}

}

}

}

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

//-----------Ноутбуки---------------

//Всі

**private** **void** **button6\_Click**(**object** sender, EventArgs e)

{

radioButton3.Checked = **false**;

radioButton4.Checked = **false**;

numericUpDown2.Visible = **false**;

label2.Visible = **false**;

label3.Visible = **false**;

numericUpDown3.Visible = **false**;

numericUpDown3.Value = **0**;

comboBox1.Visible = **false**;

comboBox1.SelectedItem = **null**;

listBox1.Items.Clear();

**foreach** (**var** item **in** dbLT.LTs.Local.ToList())

{ listBox1.Items.Add($"Ціна:[{item.Price}]грн,Виробник:{item.Virobnuk};Об'єм RAM:{item.RAM};Об'єм жорского диску:{item.HD};Кількість ядер:{item.YadraCount},Діагональ:{item.Diagonal},Динаміки:{item.Dinamics}."); }

}

//в наявності

**private** **void** **button7\_Click**(**object** sender, EventArgs e)

{

radioButton3.Checked = **false**;

radioButton4.Checked = **false**;

numericUpDown2.Visible = **false**;

label2.Visible = **false**;

label3.Visible = **false**;

numericUpDown3.Visible = **false**;

numericUpDown3.Value = **0**;

comboBox1.Visible = **false**;

comboBox1.SelectedItem = **null**;

listBox1.Items.Clear();

**foreach** (**var** item **in** dbLT.LTs.Local.ToList())

{

**if** (item.isInStock == **true**)

{

listBox1.Items.Add($"Ціна:[{item.Price}]грн,Виробник:{item.Virobnuk};Об'єм RAM:{item.RAM};Об'єм жорского диску:{item.HD};Кількість ядер:{item.YadraCount},Діагональ:{item.Diagonal},Динаміки:{item.Dinamics}.");

}

}

}

// За промокодом/виробником

**private** **void** **button8\_Click**(**object** sender, EventArgs e)

{

listBox1.Items.Clear();

**if** (radioButton3.Checked == **true**)

{

**foreach** (**var** item **in** dbLT.LTs.Local.ToList())

{

**if** (item.Virobnuk == (**string**)comboBox1.SelectedItem)

{

listBox1.Items.Add($"Виробник:{item.Virobnuk};Ціна:[{item.Price}]грн;Об'єм RAM:{item.RAM};Об'єм жорского диску:{item.HD};Кількість ядер:{item.YadraCount},Діагональ:{item.Diagonal},Динаміки:{item.Dinamics}.");

}

}

}

**if** (radioButton4.Checked == **true**)

{

**if** (numericUpDown3.Value == **0**)

{

MessageBox.Show("Не обрано позицію");

}

**else**

{

**foreach** (**var** item **in** dbLT.LTs.Local.ToList())

{

**if** (item.ID == numericUpDown3.Value)

{

listBox1.Items.Clear();

**double** zn = (**double**)numericUpDown2.Value / **100**;

item.Znijka(zn);

listBox1.Items.Add($"Ціна:[{item.Price}]грн,Виробник:{item.Virobnuk};Об'єм RAM:{item.RAM};Об'єм жорского диску:{item.HD};Кількість ядер:{item.YadraCount},Діагональ:{item.Diagonal},Динаміки:{item.Dinamics}.");

button8.Enabled = **false**;

}

}

}

}

}

**private** **void** **radioButton3\_CheckedChanged**(**object** sender, EventArgs e)

{

listBox1.Items.Clear();

button8.Enabled = **true**;

comboBox1.Visible = **true**;

numericUpDown2.Visible = **false**;

label2.Visible = **false**;

label3.Visible = **false**;

numericUpDown3.Visible = **false**;

**foreach** (**var** item **in** dbLT.LTs.Local.ToList())

{ listBox1.Items.Add($"ID:{item.ID}; Виробник:{item.Virobnuk};Об'єм RAM:{item.RAM};Об'єм жорского диску:{item.HD};Кількість ядер:{item.YadraCount};Ціна:[{item.Price}]грн;Діагональ:{item.Diagonal};Динаміки:{item.Dinamics}."); }

}

**private** **void** **radioButton4\_CheckedChanged**(**object** sender, EventArgs e)

{

comboBox1.Visible = **false**;

numericUpDown2.Visible = **true**;

label2.Visible = **true**;

comboBox1.SelectedItem = **null**;

label3.Visible = **true**;

numericUpDown3.Visible = **true**;

listBox1.Items.Clear();

**foreach** (**var** item **in** dbLT.LTs.Local.ToList())

{ listBox1.Items.Add($"ID:{item.ID}; Виробник:{item.Virobnuk};Об'єм RAM:{item.RAM};Об'єм жорского диску:{item.HD};Кількість ядер:{item.YadraCount};Ціна:[{item.Price}]грн;Діагональ:{item.Diagonal};Динаміки:{item.Dinamics}."); }

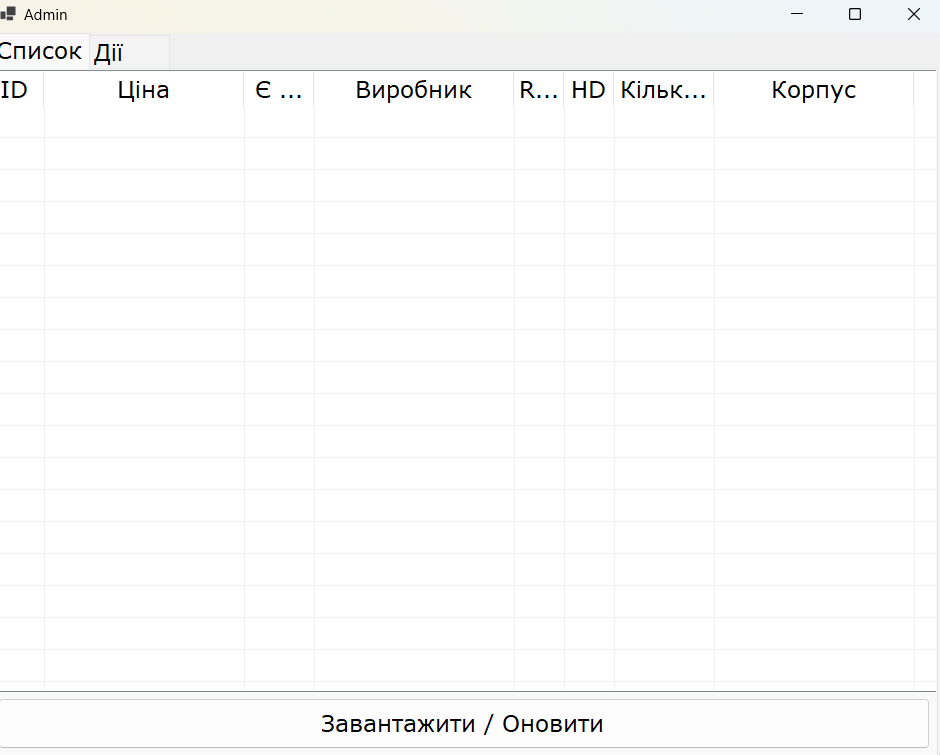
}

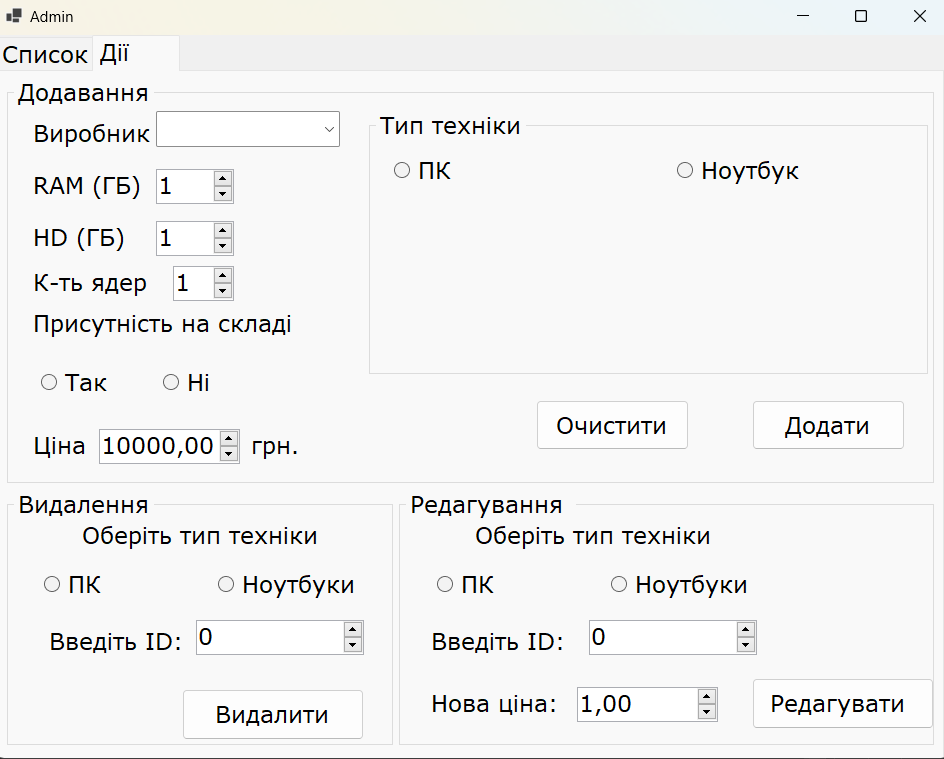
//----------------------------------

}

}

Форма Адміна:





Код:

**namespace** **KursovaWinForms**

{

**public** **partial** **class** **Admin** : Form

{

DDBContext3 dbPC;

DDBContext4 dbLT;

**public** **Admin**()

{

InitializeComponent();

listView1.GridLines = **true**;

listView1.FullRowSelect = **true**;

listView1.View = View.Details;

listView1.Scrollable = **true**;

listView1.MultiSelect = **false**;

listView1.Columns.Add("ID", **50**, HorizontalAlignment.Center);

listView1.Columns.Add("Ціна", **200**, HorizontalAlignment.Center);

listView1.Columns.Add("Є на складі", **70**, HorizontalAlignment.Center);

listView1.Columns.Add("Виробник", **200**, HorizontalAlignment.Center);

listView1.Columns.Add("RAM", **50**, HorizontalAlignment.Center);

listView1.Columns.Add("HD", **50**, HorizontalAlignment.Center);

listView1.Columns.Add("Кількість ядер", **100**, HorizontalAlignment.Center);

listView1.Columns.Add("Корпус", **200**, HorizontalAlignment.Center);

listView1.Columns.Add("Модель", **200**, HorizontalAlignment.Center);

listView1.Columns.Add("Діагональ", **200**, HorizontalAlignment.Center);

listView1.Columns.Add("Динаміки", **200**, HorizontalAlignment.Center);

dbPC = **new** DDBContext3();

dbLT = **new** DDBContext4();

dbPC.Database.EnsureCreated();

dbLT.Database.EnsureCreated();

dbPC.PCs.Load();

dbLT.LTs.Load();

}

//показ

**private** **void** **button1\_Click**(**object** sender, EventArgs e)

{

listView1.Items.Clear();

**foreach** (**var** itemPC **in** dbPC.PCs.Local.ToList())

{

ListViewItem item = **new** ListViewItem();

item.SubItems.Clear();

item.SubItems[**0**].Text = "" + itemPC.ID;

item.SubItems.Add((itemPC.Price).ToString());

item.SubItems.Add((itemPC.isInStock).ToString());

item.SubItems.Add(itemPC.Virobnuk);

item.SubItems.Add((itemPC.RAM).ToString());

item.SubItems.Add((itemPC.HD).ToString());

item.SubItems.Add((itemPC.YadraCount).ToString());

item.SubItems.Add(itemPC.Korpus);

item.SubItems.Add(itemPC.Model);

listView1.Items.Add(item);

}

**foreach** (**var** itemLT **in** dbLT.LTs.Local.ToList())

{

ListViewItem item = **new** ListViewItem();

item.SubItems.Clear();

item.SubItems[**0**].Text = "" + itemLT.ID;

item.SubItems.Add((itemLT.Price).ToString());

item.SubItems.Add((itemLT.isInStock).ToString());

item.SubItems.Add(itemLT.Virobnuk);

item.SubItems.Add((itemLT.RAM).ToString());

item.SubItems.Add((itemLT.HD).ToString());

item.SubItems.Add((itemLT.YadraCount).ToString());

item.SubItems.Add("");

item.SubItems.Add("");

item.SubItems.Add((itemLT.Diagonal).ToString());

item.SubItems.Add(itemLT.Dinamics);

listView1.Items.Add(item);

}

}

//редагувати

**private** **void** **button3\_Click**(**object** sender, EventArgs e)

{

**if** (radioButton8.Checked == **true**)

{

**foreach** (**var** itemP **in** dbPC.PCs.Local.ToList())

{

**if** (itemP.ID == numericUpDown7.Value)

{

itemP.Price = (**double**)numericUpDown8.Value;

DialogResult res = MessageBox.Show("Ви впевнені що хочете змінити ціну?", "Warning", MessageBoxButtons.YesNo, MessageBoxIcon.Warning, MessageBoxDefaultButton.Button2);

**if** (res == DialogResult.Yes) { MessageBox.Show("Оновлено"); dbPC.SaveChanges(); }

}

}

}

**if** (radioButton7.Checked == **true**)

{

**foreach** (**var** itemL **in** dbLT.LTs.Local.ToList())

{

**if** (itemL.ID == numericUpDown7.Value)

{

itemL.Price = (**double**)numericUpDown8.Value;

DialogResult res = MessageBox.Show("Ви впевнені що хочете змінити ціну?", "Warning", MessageBoxButtons.YesNo, MessageBoxIcon.Warning, MessageBoxDefaultButton.Button2);

**if** (res == DialogResult.Yes) { MessageBox.Show("Оновлено"); dbLT.SaveChanges(); }

}

}

}

radioButton7.Checked = **false**;

radioButton8.Checked = **false**;

numericUpDown7.Value = **0**;

numericUpDown8.Value = **1**;

}

**private** **void** **Clean**()

{

comboBox1.SelectedItem = **null**;

numericUpDown5.Value = **11**;

comboBox3.SelectedItem = **null**;

comboBox4.SelectedItem = **null**;

textBox2.Text = "";

radioButton1.Checked = **false**;

radioButton2.Checked = **false**;

radioButton3.Checked = **false**;

radioButton4.Checked = **false**;

numericUpDown1.Value = **1**;

numericUpDown2.Value = **1**;

numericUpDown3.Value = **1**;

numericUpDown4.Value = **10000**;

numericUpDown5.Visible = **false**;

comboBox3.Visible = **false**;

comboBox4.Visible = **false**;

textBox2.Visible = **false**;

label6.Visible = **false**;

label7.Visible = **false**;

}

//видалити

**private** **void** **button4\_Click**(**object** sender, EventArgs e)

{

**if** (radioButton6.Checked == **true**)

{

**foreach** (**var** itemP **in** dbPC.PCs.Local.ToList())

{

**if** (itemP.ID == numericUpDown6.Value)

{

dbPC.Remove(itemP);

DialogResult res = MessageBox.Show("Ви впевнені що хочете видалити?", "Warning", MessageBoxButtons.YesNo, MessageBoxIcon.Warning, MessageBoxDefaultButton.Button2);

**if** (res == DialogResult.Yes) { MessageBox.Show("Видалено"); dbPC.SaveChanges(); }

}

}

}

**if** (radioButton5.Checked == **true**)

{

**foreach** (**var** itemL **in** dbLT.LTs.Local.ToList())

{

**if** (itemL.ID == numericUpDown6.Value)

{

dbPC.Remove(itemL);

DialogResult res = MessageBox.Show("Ви впевнені що хочете видалити?", "Warning", MessageBoxButtons.YesNo, MessageBoxIcon.Warning, MessageBoxDefaultButton.Button2);

**if** (res == DialogResult.Yes) {

MessageBox.Show("Видалено"); dbLT.SaveChanges(); }

}

}

}

radioButton5.Checked = **false**;

radioButton6.Checked = **false**;

numericUpDown6.Value = **0**;

}

//додати

**private** **void** **button6\_Click**(**object** sender, EventArgs e)

{

**bool** isStock;

**if** (comboBox1.SelectedItem != **null**

&& (radioButton1.Checked == **true** || radioButton2.Checked == **true**)

&& (radioButton3.Checked == **true** || radioButton4.Checked == **true**))

{

**if** (radioButton1.Checked == **true**) { isStock = **true**; }

**else** { isStock = **false**; }

**if** (radioButton3.Checked == **true**)

{

**if** (comboBox4.SelectedItem == **null** || textBox2.Text == "")

{ MessageBox.Show("Не введено дані."); }

**else**

{

dbPC.Add(**new** PC((**string**)comboBox1.SelectedItem, (**int**)numericUpDown1.Value,

(**int**)numericUpDown2.Value, (**int**)numericUpDown3.Value, (**double**)numericUpDown4.Value,

isStock, textBox2.Text, (**string**)comboBox4.SelectedItem)); dbPC.SaveChanges();

MessageBox.Show("Додано. Поля будуть очищенні.");

Clean();

}

}

**else**

{

**if** (comboBox3.SelectedItem == **null**)

{ MessageBox.Show("Не введено дані."); }

**else**

{

dbLT.Add(**new** LapTop((**string**)comboBox1.SelectedItem, (**int**)numericUpDown1.Value,

(**int**)numericUpDown2.Value, (**int**)numericUpDown3.Value, (**double**)numericUpDown4.Value,

isStock, (**double**)numericUpDown5.Value, (**string**)comboBox3.SelectedItem)); dbLT.SaveChanges();

MessageBox.Show("Додано. Поля будуть очищенні.");

Clean();

}

}

}

**else**

{

MessageBox.Show("Необрано варіанти.");

}

}

//очистити

**private** **void** **button5\_Click**(**object** sender, EventArgs e)

{

Clean();

}

**private** **void** **radioButton3\_CheckedChanged**(**object** sender, EventArgs e)

{

label6.Visible = **true**;

label6.Text = "Корпус";

label7.Visible = **true**;

label7.Text = "Модель";

comboBox4.Visible = **true**;

textBox2.Visible = **true**;

numericUpDown5.Visible = **false**;

comboBox3.Visible = **false**;

}

**private** **void** **radioButton4\_CheckedChanged**(**object** sender, EventArgs e)

{

comboBox4.Visible = **false**;

textBox2.Visible = **false**;

label6.Visible = **true**;

label6.Text = "Діагональ";

label7.Visible = **true**;

label7.Text = "Динаміки";

numericUpDown5.Visible = **true**;

comboBox3.Visible = **true**;

}

}

}

# **ВИСНОВКИ**

Виконавши цю курсову роботу, я навчилася створювати табличні бази даних за допомогою вбудованого SQLServer. Також я навчилася ефективно розставляти функції та використовувати методи і switch. Це також була гарною практикою у створенні Windows Forms додатків.

# **РЕСУРСИ МЕРЕЖІ ІNTЕRNЕT**

* [Лекція 10. Анонімні методи і лямбда-оператори](https://classroom.google.com/u/1/w/NjIwNjIyMTMyOTA2/t/all)
* <https://itproger.com/ua/spravka/csharp/lambda>
* <http://hilite.me/>