

Міністерство освіти і науки України
Черкаський національний університет імені Богдана Хмельницького
Факультет обчислювальної техніки, інтелектуальних та управляючих систем
Кафедра інформаційних технологій

КУРСОВА РОБОТА

з дисципліни «Об'єктно-орієнтоване програмування»
на тему: «Програмний продукт для вибору моделі електроінструменту»

Студента 2 курсу КН-18 групи
спеціальності
122 Комп'ютерні науки
(код Назва)

Гайша Я.Ю.
(прізвище та ініціали)

Керівник: Гук В. І.
(прізвище та ініціали)

Оцінка за шкалою:

(національною, кількість балів, ECTS)

Члени комісії:

_____	_____
(підпис)	(прізвище та ініціали)
_____	_____
(підпис)	(прізвище та ініціали)
_____	_____
(підпис)	(прізвище та ініціали)

Черкаси – 2020

Вступ

Мета курсової роботи: розробити програмний продукт, що дозволяє переглядати характеристики електричних інструментів.

Задачі курсової роботи:

- переглянути та проаналізувати варіації додатків чи систем, що допомагають переглядати характеристики електричних інструментів;
- розробити об'єктно-орієнтовану модель додатку;
- створити ієрархію класів на основі розробленої об'єктно-орієнтованої моделі;
- реалізувати спроектований продукт;
- оформити пояснювальну записку та зробити висновки щодо виконання та реалізації програмного продукту.

Можливі варіанти реалізації додатку

Під час проектування програмного продукту були розглянуті наступні варіанти реалізації додатку: Windows Forms, Windows Presentation Foundation(WPF), Web-сторінка.

- Windows Forms – інструмент розробки інтелектуальних клієнтів. Інтелектуальний клієнт – це програма із графічним інтерфейсом, яка може працювати як за наявності, так і за відсутності підключення до мережі Інтернет; надає більш безпечний доступ до ресурсів комп'ютера, ніж звичайні додатки для OS Windows.

Windows Forms додаток представлений у вигляді форм – спеціальних візуальних вікон, на яких виводиться потрібна інформація для користувача. Окрім цього, на форми можуть бути винесені різні елементи управління, в тому числі і для вводу даних користувачем. Елементи управління можуть бути різними: кнопки, текстові поля,

таблиці, списки, перемикачі, веб-сторінки та інші. За допомогою можливостей середовища розробки Visual Studio, під час створення Windows Forms додатків, можна використовувати зручний конструктор форм, на який можна перетягувати елементи управління, задавати їм певні властивості, а також додавати їм обробку подій, таких як завантаження, закривання, натискання, вибір елемента тощо.

Отже, Windows Forms є зручним та легким інструментом для створення додатків для операційної системи Windows.[4]

- Windows Presentation Foundation(WPF)

Платформа WPF дозволяє створювати клієнтські програми для операційної системи Windows зі зручним та інтуїтивно зрозумілим графічним інтерфейсом. В основі WPF знаходиться незалежний від розширення екрану векторний модуль візуалізації, який використовує можливості сучасного графічного обладнання.

Можливості WPF розширює вбудована інструмент розмітки сторінок XAML, який допомагає зробити красивий графічний інтерфейс, практично будь-якого рівня складності, у цьому WPF випереджає Windows Forms. Окрім цього, Windows Presentation Foundation дозволяє описувати взаємодію із користувачем та виконання бізнес-логіки додатку.[5]

- Web-сторінка. Даний вид програмного продукту є найпопулярнішим способом виконання більшості систем на сьогоднішній день. Може бути написана за допомогою мови розмітки HTML, можуть бути додані візуальні стилі за допомогою формальної мови опису зовнішнього вигляду документу CSS. Окрім цього, може бути доданий код мовою програмування JavaScript, правильне використання якого може надати практично будь-який функціонал сторінки. Також, зовнішня частина може бути написана за допомогою сучасних фреймворків React, Angular, VueJS тощо.

Серверна частина може бути додана за допомогою мов програмування PHP, Java, Python та відповідних їм фреймворків, що дозволило би проводити будь-які операції із даними та запитам на цю сторінку.

Отже, Web-сторінка – найкращий вибір сьогодні як для простих, так і для складних та навантажених додатків.

Проектування програмного продукту

Загальний алгоритм роботи додатку:

Робота додатку заключається у наступній послідовності дій:

1. Запуск програми;
2. Перегляд усіх збережених електричних інструментів;
3. Можливе сортування усіх електричних інструментів за певними властивостями;
4. Можливе фільтрування усіх електричних інструментів за окремими параметрами;
5. Додавання нового запису про характеристики електричних інструментів;
6. Наявна можливість зміни інформації про характеристики електричних інструментів;
7. Можливе видалення запису про електричний інструмент;
8. Можливість перегляду повної інформації про обраний електричний інструмент.
9. Вихід із програми.

У додатку А відображений загальний алгоритм роботи програми.

Опис структури класів додатку

Назва додатку – «Програмний продукт для вибору моделі електроінструменту». Так як прототипами додатку слугували Інтернет-сайти, які спеціалізуються на інтернет магазинах продажу електроінструменту, було вирішено розробити наступну структуру класів:

- Клас PowerTool, базовий. Саме до цього класу у процесі виконання програми будуть приводитись похідні класи, які описані нижче. У цьому класі були створені поля: унікальний номер електроінструменту, ціна, повна назва, день, місяць та рік виготовлення, країна виробника, тип електроінструменту та його фотографія.

Базовий клас PowerTool містить 3 властивості: давність товару, об'єм товару, акційна ціна товару.

Базовий клас PowerTool містить 2 віртуальних методи, які будуть перевизначені у класах-нащадках. Даними методами є: ціна доставки, знижка % в залежності від давності товару.

Опис класів-нащадків:

- Клас Bulgarians : PowerTool – даний клас є представленням болгарок. У цьому класі перевизначаються віртуальні методи: ціна доставки - 35 грн для всіх болгарок; знижка – процент знижки дорівнює кількості років товару, але не більше 5%.
- Клас Drills: PowerTool – даний клас є представленням дрилів. У цьому класі перевизначаються віртуальні методи: ціна доставки - 30 грн для всіх дрилів; знижка – за перший рік знижка становить 4%, а за кожний наступний додається 2%, але не більше 12%.
- Клас Grinders: PowerTool – даний клас є представленням шліфувальних машин. У цьому класі перевизначаються віртуальні

методи: ціна доставки - 100 грн для всіх шліфувальних машин; знижка – становить 5% , якщо товару 2 і більше років.

- Клас Perforators : PowerTool – даний клас є представленням перфораторів. У цьому класі перевизначаються віртуальні методи: ціна доставки - 50 грн при об'ємі меншому за 0.03, 100 грн при об'ємі меншому за 0.06, в інших випадках 200 грн для всіх перфораторів; знижка – за перший рік 5%, а за кожний наступний на 2% більше, але не більше 13%.
- Клас Saw : PowerTool – даний клас є представленням пил. У цьому класі перевизначаються віртуальні методи: ціна доставки - 50 грн при об'ємі меншому за 0.03, 100 грн при об'ємі меншому за 0.06, 200 грн при об'ємі меншому за 0.12, в інших випадках 400 грн для всіх пил; знижка – за перший рік 2%, за 2 роки 5%, а за кожний наступний рік на 1% більше, але не більше 8%.
- Клас Screwdrivers : PowerTool – даний клас є представленням шоруповертів. У цьому класі перевизначаються віртуальні методи: ціна доставки - 20 грн для всіх шоруповертів; знижка – за перший рік 2%, а за кожний наступний рік на 2% більше, але не більше 10%.

Ієрархія класів електроінструментів наведено в додатку А.

Опис алгоритмів фільтрування додатку

Даний програмний продукт має можливість сортувати та фільтрувати записи про електроінструмент. Фільтрування записів можна провести за такими параметрами: тип електроінструменту, діапазон ціни.

Окрім цього, був створений метод - `is_filter`, у якому реалізовано фільтрацію електроінструментів за типом, ціною та їх комбінацією. Даний метод приймає електроінструмент і повертає істину якщо він відповідає критеріям фільтрування і хибу в противному випадку.

У додатку В наведені блок-схеми алгоритму роботи фільтрування.

Висновки

Після проектування програмного продукту постає необхідність розробити:

1. усі потрібні класи та структури для реалізації об'єктно-орієнтованої моделі;
2. класи форм, методи виведення та обробки даних;
3. додаткові класи для збереження, прочитання та доступу до даних;
4. інтерфейс користувача.

Реалізація програмного продукту

Реалізація основних класів та методів

Продемонструємо, як побудовані класи електроінструментів та реалізація перевизначених методів базового класу.

Реалізація класу PowerTool

```
public class PowerTool
{
    public int Number { get; set; }
    public string Price { get; set; }
    public string Name { get; set; }
    public int Day { get; set; }
    public int Month { get; set; }
    public int Year { get; set; }
    public string Country { get; set; }
    public int PowerType { get; set; }
    public string photo_file_name { get; set; }
    public int Years //Давність товару, років
    {
        get
        {
            DateTime date1 = new DateTime(Year, Month, Day);
            DateTime sysdate = DateTime.Now;
            var date3 = sysdate.Subtract(date1);
            return (int)(date3.Days / 365);
        }
    }
    public int [] Sides; //масив розмірів
    public double V { get { return Sides[0]*Sides[1]*Sides[2]/1000000.0 ; } } //Об'єм

    public List<string> features;
    public PowerTool()
    { PowerType = 1; }

    public virtual double DeliveryPrice()//ціна доставки
    {
        return 50;
    }
    public virtual double ActionPricepercent()//знижка % в залежності від давності товару
    {
        return 0;
    }
    public double ActionPrice { get { return double.Parse(Price) * (100.0 - ActionPricepercent()) / 100.0; } } //Акційна
    ціна, що враховує знижку
}
}
```


Методи DeliveryPrice(), ActionPricepercent() є віртуальними та у базовому класі повертають по замовчуванню 50 грн – вартість доставки 0% скидки. Властивість ActionPrice – повертає ціну товару з врахуванням процентної скидки.

Реалізація властивості ActionPrice.

```
public double ActionPrice { get { return double.Parse(Price) * (100.0 - ActionPricepercent()) / 100.0; } }
```

Реалізація класу Bulgarians:

```
using System;

namespace WindowsFormsCourseWork
{
    [Serializable]
    public class Perforators : PowerTool
    {
        public Perforators() { }
        public Perforators(PowerTool p) : base(p) { }
        public Perforators(int Number, string Price, string Name, int Day, int Month, int Year, int
PowerType, string Country, string photo_file_name = "") : base(Number, Price, Name, Day, Month, Year,
PowerType, Country, photo_file_name)
        { }
        public override double DeliveryPrice()//ціна доставки
        {
            if(V<0.03) return 50;
            else if(V<0.06)return 100;
            return 200;
        }
        public override double ActionPricepercent()//знижка %
        {
            if (Years >= 5) return 13;
            if (Years >= 4) return 11;
            if (Years >= 3) return 9;
            if (Years >= 2) return 7;
            if (Years >= 1) return 5;
            return 0;
        }
    }
}
```

Реалізація класу Drills:

```
using System;

namespace WindowsFormsCourseWork
{
    [Serializable]
    public class Drills : PowerTool//Дрилі
    {
        public Drills() { }
        public Drills(PowerTool p) : base(p) { }
        public Drills(int Number, string Price, string Name, int Day, int Month, int Year, int
PowerType, string Country, string photo_file_name = "") : base(Number, Price, Name, Day, Month, Year,
PowerType, Country, photo_file_name)
        { }
        public override double DeliveryPrice()//ціна доставки
        {
            return 30;
        }
        public override double ActionPricepercent()//знижка %
        {
            if (Years >= 5) return 12;
            if (Years >= 4) return 10;
            if (Years >= 3) return 8;
            if (Years >= 2) return 6;
            if (Years >= 1) return 4;
            return 0;
        }
    }
}
```

Реалізація класу Grinders:

```
namespace WindowsFormsCourseWork
{
    [Serializable]
    public class Grinders : PowerTool//Шліфувальні машини
    {
        public Grinders() { }
        public Grinders(PowerTool p) : base(p) { }
        public Grinders(int Number, string Price, string Name, int Day, int Month, int Year, int
PowerType, string Country, string photo_file_name = "") : base(Number, Price, Name, Day, Month, Year,
PowerType, Country, photo_file_name)
        { }
        public override double DeliveryPrice()//ціна доставки
        {
            return 100;
        }
        public override double ActionPricepercent()//знижка %
        {
            if (Years >= 2) return 5;
            return 0;
        }
    }
}
```

Релізація класу Perforators:

```
using System;

namespace WindowsFormsCourseWork
{
    [Serializable]
    public class Perforators : PowerTool
    {
        public Perforators() { }
        public Perforators(PowerTool p) : base(p) { }
        public Perforators(int Number, string Price, string Name, int Day, int Month, int Year, int
PowerType, string Country, string photo_file_name = "") : base(Number, Price, Name, Day, Month, Year,
PowerType, Country, photo_file_name)
        { }
        public override double DeliveryPrice()//ціна доставки
        {
            if(V<0.03) return 50;
            else if(V<0.06)return 100;
            return 200;
        }
        public override double ActionPricepercent()//знижка %
        {
            if (Years >= 5) return 13;
            if (Years >= 4) return 11;
            if (Years >= 3) return 9;
            if (Years >= 2) return 7;
            if (Years >= 1) return 5;
            return 0;
        }
    }
}
```

Релізація класу Saw:

```
using System;

namespace WindowsFormsCourseWork
{
    [Serializable]
    public class Saw : PowerTool//Пилки
    {
        public Saw() { }
        public Saw(PowerTool p) : base(p) { }
        public Saw(int Number, string Price, string Name, int Day, int Month, int Year, int PowerType,
string Country, string photo_file_name = "") : base(Number, Price, Name, Day, Month, Year, PowerType,
Country, photo_file_name)
        { }
        public override double DeliveryPrice()//ціна доставки
        {
            if (V < 0.03) return 50;
            else if (V < 0.06) return 100;
            else if (V < 0.12) return 200;
            else return 400;
        }
        public override double ActionPricepercent()//знижка %
    }
```

```

    {
        if (Years >= 5) return 8;
        if (Years >= 4) return 7;
        if (Years >= 3) return 6;
        if (Years >= 2) return 5;
        if (Years >= 1) return 2;
        return 0;
    }
}
}

```

Релізація класу Screwdrivers:

using System;

namespace WindowsFormsCourseWork

{

[Serializable]

public class Screwdrivers: PowerTool//Шоруповерт

{

public Screwdrivers(){ }

public Screwdrivers(PowerTool p):base(p){ }

public Screwdrivers(int Number, string Price, string Name, int Day, int Month, int Year, int
PowerType, string Country, string photo_file_name = "") : base(Number, Price, Name, Day, Month,
Year, PowerType, Country, photo_file_name)

{ }

public override double DeliveryPrice()//цена доставки

{

return 20;

}

public override double ActionPricepercent()//скидка % в зависимость от давности товара и
типа товара

{

if (Years >= 5) return 10;// 10%

if (Years >= 4) return 8;

if (Years >= 3) return 6;

if (Years >= 2) return 4;

if (Years >= 1) return 2;

return 0;

}

}

}

Під час натискання на кнопку для створення нового електроінструменту на формі для створення чи редагування інформації про характеристики, виконується такий метод, який прив'язаний до обробника подій:

```
осилка: 1
private void toolStripButtonAdd_Click(object sender, EventArgs e)
{
    Form_Add_PowerTool form = new Form_Add_PowerTool(lstT, lstPT.Count + 1, Countrys);
    form.ShowDialog();
    if (form.pt != null)
    {
        if (form.pt.PowerType == 1) lstPT.Add(new Screwdrivers(form.pt));
        else if (form.pt.PowerType == 2) lstPT.Add(new Drills(form.pt));
        else if (form.pt.PowerType == 3) lstPT.Add(new Perforators(form.pt));
        else if (form.pt.PowerType == 4) lstPT.Add(new Bulgarians(form.pt));
        else if (form.pt.PowerType == 5) lstPT.Add(new Saw(form.pt));
        else if (form.pt.PowerType == 6) lstPT.Add(new Screwdrivers(form.pt));
        else lstPT.Add(new PowerTool(form.pt));

        int i = lstPT.Count - 1;
        dataGridView2.Rows.Add(lstT[lstPT[i].PowerType - 1].Name, lstPT[i].ActionPrice, lstPT[i].Name, lstPT[i].'
```

Метод toolStripButtonAdd_Click

На початку, цей метод викликає форму для створення нового електроінструменту на якій перевіряється, чи усі поля заповнені. Якщо дані були успішно заповнені то відповідний електроінструмент добавляється в список електроінструментів і виводиться на екран. Також перевіряється тип електроінструменту і в залежності від нього створюється відповідний клас.

Створення екземпляру похідного класу:

```
if (form.pt.PowerType == 1) lstPT.Add(new Screwdrivers(form.pt));
else if (form.pt.PowerType == 2) lstPT.Add(new Drills(form.pt));
else if (form.pt.PowerType == 3) lstPT.Add(new Perforators(form.pt));
else if (form.pt.PowerType == 4) lstPT.Add(new Bulgarians(form.pt));
else if (form.pt.PowerType == 5) lstPT.Add(new Saw(form.pt));
else if (form.pt.PowerType == 6) lstPT.Add(new Screwdrivers(form.pt));
else lstPT.Add(new PowerTool(form.pt));
```

Для виведення усіх електроінструментів з урахуванням фільтру та критерію сортування використовується метод LoadToDataGridView().

Реалізація методу LoadToDataGridView():

```
private void LoadToDataGridView()
{
    dataGridView2.Rows.Clear();
    for (int i = 0; i < lstPT.Count; i++)
    {
        if (is_filter(lstPT[i]))
            dataGridView2.Rows.Add(lstT[lstPT[i].PowerType - 1].Name, lstPT[i].ActionPrice, lstPT[i].ActionDate);
    }
    if (comboBoxFaculty.SelectedIndex != 0)
        toolStripStatusLabel2.Text = "Застосований фільтр! Відображаються не всі дані!";
    else toolStripStatusLabel2.Text = "";
}
```

У методі LoadToDataGridView() відбувається перебір електроінструментів і перевірка їх на відповідність фільтру. Якщо електроінструмент відповідає фільтру то він добавляється до таблиці – dataGridView2.

На формі, де виводиться інформація про обраний електроінструмент (Form_Add_PowerTool), викликається конструктор , який заповнює усі написи інформацією, отриманою із полів електроінструменту, який був прийнятий даною формою через параметр конструктора.

Конструктор форми Form_Add_PowerTool:

```
public Form_Add_PowerTool(List<PowerToolType> lstT, PowerTool pt, List<string> Countrys) :
this(lstT, pt.Number, Countrys)
{
    textBoxPrice.Text = pt.Price;
    textBoxName.Text = pt.Name;
    textBoxWeight.Text = pt.V.ToString();
    comboBoxType.SelectedIndex = pt.PowerType - 1;
    dateTimePicker1.Value = new DateTime(pt.Year, pt.Month, pt.Day);
    this.Text = pt.Price + " № " + pt.Number;

    textBoxWidth.Text = pt.Sides[0].ToString();
    textBoxHeight.Text = pt.Sides[1].ToString();
    textBoxLength.Text = pt.Sides[2].ToString();

    for (int i = 0; i < pt.features.Count; i++)
    {
        listBoxP.Items.Add(pt.features[i]);
    }
    file_name = pt.photo_file_name;
    string new_file_name = "Photo\\" + pt.photo_file_name;
    if (File.Exists(new_file_name))
    {
        image = new Bitmap(new_file_name);
    }
    else
        image = new Bitmap("Photo\\no_pic_person.jpg");
    pictureBox1.Image = image;
    pictureBox1.Invalidate();
    //image.Dispose();
    try
    { comboBoxCountry.SelectedIndex = Int32.Parse(pt.Country); }
    catch
    {
    }
}
```

Отже, під час реалізації додатку були створені віртуальні методи, перевизначені у похідних типах. Під час створення екземпляру він приводиться до базового типу, а під час роботи із додатком дані методи викликаються для отримання необхідного результату.

Опис користувацького інтерфейсу

На початку використання додатку відкривається основна форма(Form1), на якій відображені усі записи в короткому форматі, кількість знайдених записів, випадаючий список, в якому обирається параметр, за яким сортується список та панель фільтрів із кнопкою пошуку за такими характеристиками. Окрім цього, на даній формі є 3 кнопки: створити новий запис, відредагувати чи видалити обраний запис.

Тип	Акційна ціна, грн	Назва	Давність товару, років	Доставка, грн	Виробник	№
Дрилі	3455,04	Дрель-шуруповерт Bosch Pr...	1	30	Китай	2
Перфоратори	3131,13	Перфоратор Bosch Professio...	23	50	Україна	4
Шуруповерти	1534,68	Пила циркулярная DWT HKS...	1	20	Египет	1
Шліфмашинки	1066,56	Угловая шлифмашина Bosch...	2	20	Україна	3
Болгарки	1500	Болгарка B1000	0	35	Австралия	5

Інтерфейс форми Form1

При натисканні кнопок «додати запис» чи «редагувати запис», відкривається форма(Form_Add_PowerTool), на якій задаються характеристики електроінструменту, а також можливе додавання його фото. Дана форма містить усі необхідні поля, які потрібно заповнити. Якщо натискається кнопка «додати»,

ця форма створюється повністю пустою, а якщо кнопка «редагувати» - форма Form_Add_PowerTool відкривається із полями, які автоматично заповнені значеннями обраного електроінструменту, інформація про характеристики якого редагується. На даній формі також знаходиться кнопка, яка зберігає обрані користувачем зміни чи створює новий запис.

Інтерфейс форми Form_Add_PowerTool при додаванні запису:

The screenshot shows a web form titled "Інформація про електроінструмент" (Information about power tool). The form is designed for adding a new record and includes the following fields and controls:

- Повна назва** (Full name): A large text input field.
- Базова ціна, грн** (Base price, UAH): A text input field.
- №** (Number): A text input field containing the value "6".
- Дата виготовлення** (Date of manufacture): A date picker showing "2 сентября 2020" with a dropdown arrow.
- Виробник** (Manufacturer): A dropdown menu showing "Австралия" (Australia).
- Тип** (Type): A dropdown menu showing "Шуруповерти" (Drills).
- Ширина (см)** (Width (cm)): A text input field with "1" and a "..." button.
- Висота (см)** (Height (cm)): A text input field with "1" and a "..." button.
- Довжина (см)** (Length (cm)): A text input field with "1" and a "..." button.
- Об'єм (м^3)** (Volume (m³)): A text input field.
- Image Upload:** A large area with a camera icon, a "Завантажити..." (Upload...) button, and a "Видалити" (Delete) button.
- Key Features:** A section titled "Ключові особливості" (Key features) with a large green text area, a "Добавити..." (Add...) button, and a "Видалити" (Delete) button.
- Save:** A "Зберегти" (Save) button at the bottom center.

Інтерфейс форми Form_Add_PowerTool при редагуванні запису:

3 599 № 2

Повна назва: Дрель-шуруповерт Bosch Professional GSR 12V-30

Базова ціна, грн: 3 599

№: 2

Дата виготовлення: 23 июля 2019

Виробник: Китай

Тип: Дрилі

Ширина (см): 0

Висота (см): 0

Довжина (см): 0

Об'єм (м³): 0

Завантажити...


Видалити

Ключові особливості

Добавити...

Видалити

Зберегти



Натискання на кнопку «видалити» на головній формі видаляє обраний запис та автоматично оновлює сторінку. На вибір є список 3 сортування (по ціні, по давності, по назві).

Вибір для сортування:

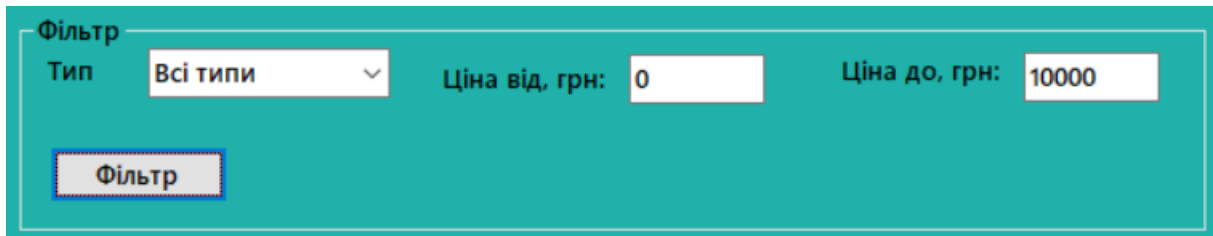
Сортувати по ціні

Сортувати по давності

Сортувати по назві

Під час користування реалізованими у додатку фільтрами, можна обирати такі параметри фільтрування як: тип електроінструменту, діапазон ціни.

Вибір значень параметрів фільтрування:



Фільтр

Тип:

Ціна від, грн:

Ціна до, грн:

Висновок:

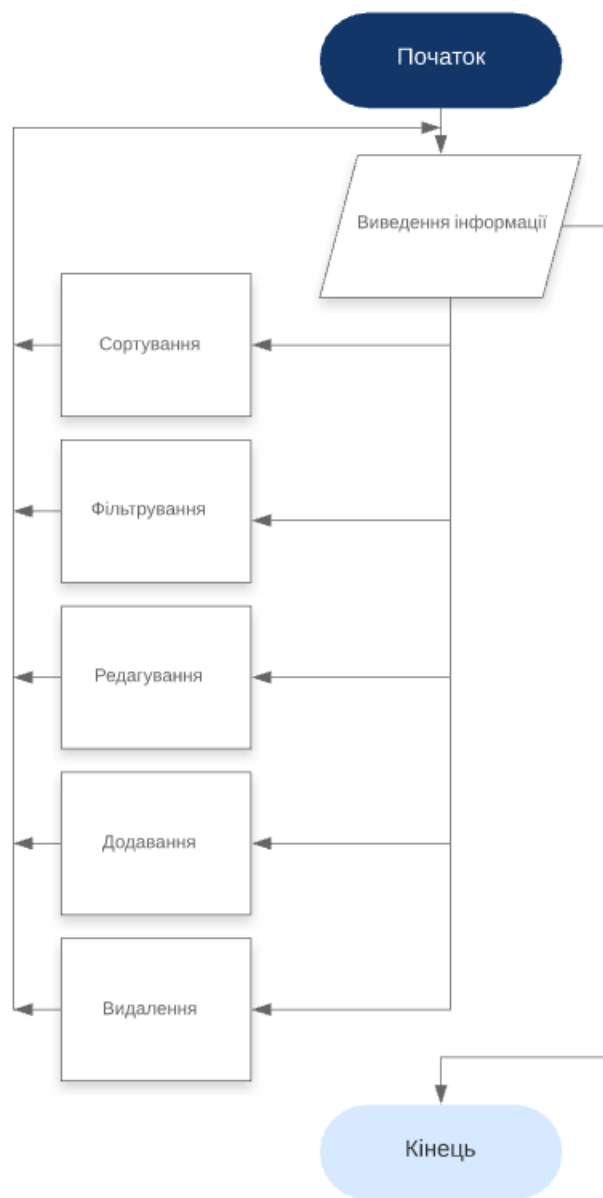
Додаток має базовий функціонал довідника, в якому можна виконувати такі дії: додавати запис, переглядати усі записи, видаляти запис, переглядати детальну інформацію про якийсь конкретний запис, редагувати запис (можливо викликати редагування при подвійному натисненні на об'єкт), а також сортувати та фільтрувати їх за певними параметрами.

Додаток має ієрархію класів та віртуальні методи.

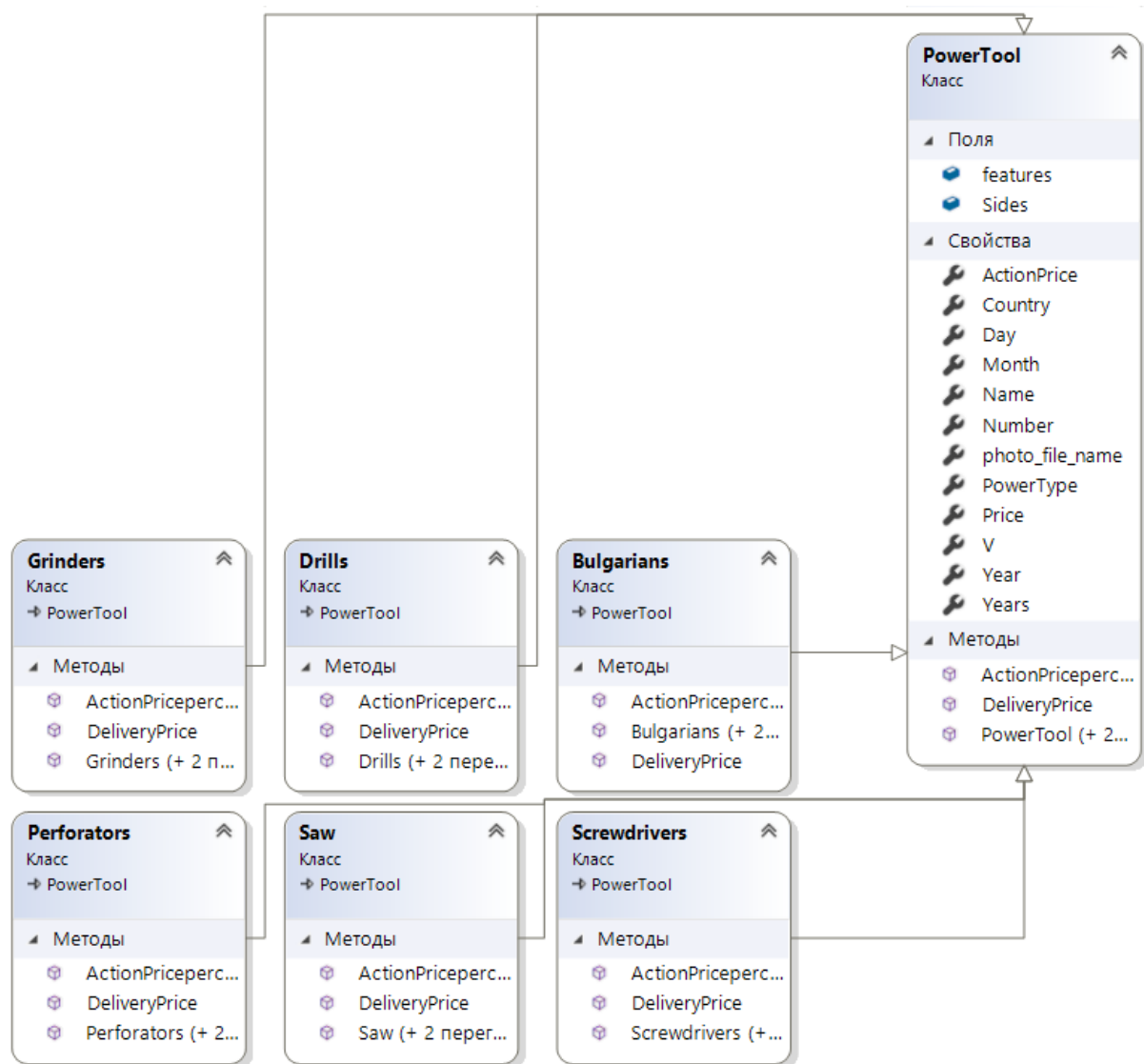
За допомогою цієї програми можна із легкістю працювати, переглядаючи, додаючи чи редагуючи характеристики електроінструментів. Окрім цього, за допомогою реалізованого сортування та фільтрації можна швидко знаходити необхідний запис, який потрібно переглянути.

Додаток був зроблений за допомогою інструменту Windows Forms мовою програмування C#.

Блок-схема загального алгоритму роботи програми:



Додаток А. Ієрархія класів програмного продукту



Додаток В. Блок-схеми алгоритму фільтрації у додатку

