ДОДАТОК А. ЛІСТИНГ ПРОГРАМИ КЕРУВАННЯ

Лістинг 1. Код класу «SearchBLL»

```
using MySqlConnector;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace ProductAccounting.AppCode {
  public class SearchBLL {
    private string _ConnString =
System.Configuration.ConfigurationSettings.AppSettings["CONNECTSQL"];

    private ProductSearch MapProduct(MySqlDataReader reader) {
      return new ProductSearch {
        ProductId = Convert.ToInt32(reader["ProductId"]),
        ProductName = reader["ProductName"].ToString(),
        Model = reader["Model"].ToString(),
        Manufacturer = reader["Manufacturer"].ToString(),
        SalePrice = Convert.ToDecimal(reader["SalePrice"]),
        Quantity = Convert.ToInt32(reader["Quantity"]),
        CategoryId = Convert.ToInt32(reader["CategoryId"]),
        CategoryName = reader["CategoryName"].ToString(),
        WarehousesId = Convert.ToInt32(reader["WarehousesId"]),
        WarehousesName = reader["WarehousesName"].ToString()
      };
    }

    private MySqlConnection GetConnection() {
      return new MySqlConnection(_ConnString);
    }

    //За вказаною назвою або частиною назви ProductName:
    public List<ProductSearch> GetProductsByName(string productName) {
      List<ProductSearch> products = new List<ProductSearch>();
      string query = $@"
        SELECT p.*, c.CategoryName, w.WarehousesName
        FROM products p
        JOIN category c ON p.CategoryId = c.CategoryId
        JOIN warehouses w ON p.WarehousesId = w.WarehousesId
        WHERE p.ProductName LIKE @ProductName";
      using (MySqlConnection connection = GetConnection()) {
        MySqlCommand command = new MySqlCommand(query, connection);
        command.Parameters.AddWithValue("@ProductName", "%" + productName + "%");
        connection.Open();
        using (MySqlDataReader reader = command.ExecuteReader()) {
          while (reader.Read()) {
            products.Add(MapProduct(reader));
```

```
      }
    }
  }
  return products;
}

//За вказаною назвою або частиною назви Model:
public List<ProductSearch> GetProductsByModel(string model) {
  List<ProductSearch> products = new List<ProductSearch>();
  string query = $@"
    SELECT p.*, c.CategoryName, w.WarehousesName
    FROM products p
    JOIN category c ON p.CategoryId = c.CategoryId
    JOIN warehouses w ON p.WarehousesId = w.WarehousesId
    WHERE p.Model LIKE @Model";

  using (MySqlConnection connection = GetConnection()) {
    MySqlCommand command = new MySqlCommand(query, connection);
    command.Parameters.AddWithValue("@Model", "%" + model + "%");
    connection.Open();
    using (MySqlDataReader reader = command.ExecuteReader()) {
      while (reader.Read()) {
        products.Add(MapProduct(reader));
      }
    }
  }

  return products;
}

//За вказаною назвою або частиною назви Manufacturer:

public List<ProductSearch> GetProductsByManufacturer(string manufacturer) {
  List<ProductSearch> products = new List<ProductSearch>();
  string query = $@"
    SELECT p.*, c.CategoryName, w.WarehousesName
    FROM products p
    JOIN category c ON p.CategoryId = c.CategoryId
    JOIN warehouses w ON p.WarehousesId = w.WarehousesId
    WHERE p.Manufacturer LIKE @Manufacturer";

  using (MySqlConnection connection = GetConnection()) {
    MySqlCommand command = new MySqlCommand(query, connection);
    command.Parameters.AddWithValue("@Manufacturer", "%" + manufacturer + "%");
    connection.Open();
    using (MySqlDataReader reader = command.ExecuteReader()) {
      while (reader.Read()) {
        products.Add(MapProduct(reader));
      }
    }
  }
  return products;
```

```
}

//Продукція дешевше за вказану SalePrice:
public List<ProductSearch> GetProductsByMaxPrice(double maxPrice) {
  List<ProductSearch> products = new List<ProductSearch>();
  string query = $@"
    SELECT p.*, c.CategoryName, w.WarehousesName
    FROM products p
    JOIN category c ON p.CategoryId = c.CategoryId
    JOIN warehouses w ON p.WarehousesId = w.WarehousesId
    WHERE p.SalePrice <= @MaxPrice";

  using (MySqlConnection connection = GetConnection()) {
    MySqlCommand command = new MySqlCommand(query, connection);
    command.Parameters.AddWithValue("@MaxPrice", maxPrice);
    connection.Open();
    using (MySqlDataReader reader = command.ExecuteReader()) {
      while (reader.Read()) {
        products.Add(MapProduct(reader));
      }
    }
  }

  return products;
}

//Продукція менша за вказану Quantity:
public List<ProductSearch> GetProductsByMaxQuantity(int maxQuantity) {
  List<ProductSearch> products = new List<ProductSearch>();
  string query = $@"
    SELECT p.*, c.CategoryName, w.WarehousesName
    FROM products p
    JOIN category c ON p.CategoryId = c.CategoryId
    JOIN warehouses w ON p.WarehousesId = w.WarehousesId
    WHERE p.Quantity <= @MaxQuantity";

  using (MySqlConnection connection = GetConnection()) {
    MySqlCommand command = new MySqlCommand(query, connection);
    command.Parameters.AddWithValue("@MaxQuantity", maxQuantity);
    connection.Open();
    using (MySqlDataReader reader = command.ExecuteReader()) {
      while (reader.Read()) {
        products.Add(MapProduct(reader));
      }
    }
  }

  return products;
}

//За ідентифікатором складу:
```

```
    public List<ProductSearch> GetProductsByWarehouseId(int warehouseId) {
      List<ProductSearch> products = new List<ProductSearch>();
      string query = $@"
        SELECT p.*, c.CategoryName, w.WarehousesName
        FROM products p
        JOIN category c ON p.CategoryId = c.CategoryId
        JOIN warehouses w ON p.WarehousesId = w.WarehousesId
        WHERE p.WarehousesId = @WarehouseId";

      using (MySqlConnection connection = GetConnection()) {
        MySqlCommand command = new MySqlCommand(query, connection);
        command.Parameters.AddWithValue("@WarehouseId", warehouseId);
        connection.Open();
        using (MySqlDataReader reader = command.ExecuteReader()) {
          while (reader.Read()) {
            products.Add(MapProduct(reader));
          }
        }
      }
      return products;
    }

  }
}

public class ProductSearch {
  public int ProductId { get; set; }
  public string ProductName { get; set; }
  public string Model { get; set; }
  public string Manufacturer { get; set; }
  public decimal SalePrice { get; set; }
  public int Quantity { get; set; }
  public int CategoryId { get; set; }
  public string CategoryName { get; set; }
  public int WarehousesId { get; set; }
  public string WarehousesName { get; set; }

  public string ToString(int orderNumber) {
    return $"№: {orderNumber}\r\n" +
          $"Назва продукції: {ProductName}\r\n" +
          $"Модель: {Model}\r\n" +
          $"Виробник: {Manufacturer}\r\n" +
          $"Ціна: {SalePrice:C}\r\n" +
          $"Кількість: {Quantity}\r\n" +
          $"Категорія: {CategoryName}\r\n" +
          $"Склад: {WarehousesName}\r\n" +
          "_____";
  }
}
```

Лістинг 2. Код класу «WarehousesProvider»

```
using MySqlConnector;
```

```
using ProductAccounting.AppCode;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace ProductAccounting.Providers {
  internal class WarehousesProvider {
    private string _ConnString =
System.Configuration.ConfigurationSettings.AppSettings["CONNECTSQL"];

    public void InsertWarehouses(string WarehousesName, string Location, string Description) {
      MySqlConnection connection = new MySqlConnection(_ConnString);
      string query = "INSERT INTO Warehouses (WarehousesName, Location, Description)
VALUES(@WarehousesName, @Location, @Description)";
      MySqlCommand cmd = new MySqlCommand(query, connection);
      cmd.Parameters.AddWithValue("@WarehousesName", WarehousesName);
      cmd.Parameters.AddWithValue("@Location", Location);
      cmd.Parameters.AddWithValue("@Description", Description);
      connection.Open();
      cmd.ExecuteNonQuery();
      connection.Close();
    }

    public Warehouses SelectedWarehousesByWarehousesId(int WarehousesId) {
      Warehouses selectedWarehouses = new Warehouses();
      MySqlConnection connection = new MySqlConnection(_ConnString);
      string query = "SELECT * FROM Warehouses WHERE WarehousesId=" +
WarehousesId.ToString();
      MySqlCommand command = new MySqlCommand(query, connection);
      connection.Open();
      MySqlDataReader reader = command.ExecuteReader();
      while (reader.Read()) {
        selectedWarehouses.WarehousesId = Convert.ToInt32(reader["WarehousesId"]);
        selectedWarehouses.WarehousesName = reader["WarehousesName"].ToString();
        selectedWarehouses.Location = reader["Location"].ToString();
        selectedWarehouses.Description = reader["Description"].ToString();
      }
      reader.Close();
      connection.Close();
      return selectedWarehouses;
    }


    public List<Warehouses> GetAllWarehouses() {
      int i = 0;
      List<Warehouses> WarehousesList = new List<Warehouses>();
      MySqlConnection connection = new MySqlConnection(_ConnString);
      string query = "SELECT * FROM Warehouses  ORDER BY WarehousesName ASC";
      MySqlCommand command = new MySqlCommand(query, connection);
      connection.Open();
```

```
      MySqlDataReader reader = command.ExecuteReader();
      while (reader.Read()) {
        Warehouses selectedWarehouses = new Warehouses();
        selectedWarehouses.Number = ++i;
        selectedWarehouses.WarehousesId = Convert.ToInt32(reader["WarehousesId"]);
        selectedWarehouses.WarehousesName = reader["WarehousesName"].ToString();
        selectedWarehouses.Location = reader["Location"].ToString();
        selectedWarehouses.Description = reader["Description"].ToString();
        WarehousesList.Add(selectedWarehouses);
      }
      reader.Close();
      connection.Close();

      if (WarehousesList.Count == 0) {
        Warehouses noWarehouses = new Warehouses();
        noWarehouses.WarehousesId = 0;
        noWarehouses.Message = NamesMy.NoDataNames.NoDataInWarehouses;
        WarehousesList.Add(noWarehouses);
      }
      return WarehousesList;
    }

    public void UpdateWarehouses(string WarehousesName, string Location, string Description,
int WarehousesId) {
      MySqlConnection connection = new MySqlConnection(_ConnString);
      string query = "UPDATE Warehouses SET WarehousesName = @WarehousesName,
Location=@Location, Description = @Description " +
          " WHERE WarehousesId = @WarehousesId";
      MySqlCommand cmd = new MySqlCommand(query, connection);
      cmd.Parameters.AddWithValue("@WarehousesName", WarehousesName);
      cmd.Parameters.AddWithValue("@Location", Location);
      cmd.Parameters.AddWithValue("@Description", Description);
      cmd.Parameters.AddWithValue("@WarehousesId", WarehousesId);
      connection.Open();
      cmd.ExecuteNonQuery();
      connection.Close();
    }

    public void DeleteWarehouses(int WarehousesId) {
      MySqlConnection connection = new MySqlConnection(_ConnString);
      string query = "DELETE FROM Warehouses WHERE WarehousesId = @WarehousesId";
      MySqlCommand cmd = new MySqlCommand(query, connection);
      cmd.Parameters.AddWithValue("@WarehousesId", WarehousesId);
      connection.Open();
      cmd.ExecuteNonQuery();
      connection.Close();
    }



  }
}
```

```
public class Warehouses {
  private int _Number;
  private int _WarehousesId;
  private string _WarehousesName;
  private string _Location;
  private string _Description;
  private string _Message;

  public Warehouses() {
    _Number = 0;
    _WarehousesId = 0;
    _WarehousesName = String.Empty;
    _Location = String.Empty;
    _Description = String.Empty;
    _Message = String.Empty;
  }

  public int Number {
    set { _Number = value; }
    get { return _Number; }
  }
  public int WarehousesId {
    set { _WarehousesId = value; }
    get { return _WarehousesId; }
  }
  public string WarehousesName {
    set { _WarehousesName = value; }
    get { return _WarehousesName; }
  }
  public string Location {
    set { _Location = value; }
    get { return _Location; }
  }
  public string Description {
    set { _Description = value; }
    get { return _Description; }
  }
  public string Message {
    set { _Message = value; }
    get { return _Message; }
  }
}
```

Лістинг 3. Код класу «WarehousesForm»

```
using ProductAccounting.AppCode;
using ProductAccounting.Providers;
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
```

```
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace ProductAccounting.Forms.Dictionary {
  public partial class WarehousesForm : Form {
    private int _selectedRowIndex = 0;
    private ValidationMy _validation = new ValidationMy();
    private WarehousesProvider _WarehousesProvider = new WarehousesProvider();
    private List<Warehouses> _WarehousesList = new List<Warehouses>();


    public WarehousesForm() {
      InitializeComponent();
      DataLoad();
    }

    private void AddBtn_Click(object sender, EventArgs e) {
      if (IsDataEnteringCorrect()) {
        _WarehousesProvider.InsertWarehouses(WarehousesNameTBox.Text, LocationTBox.Text,
DescriptionTBox.Text);
        DataLoad();
        ClearAllControls();
      }
    }

    private void ClearBtn_Click(object sender, EventArgs e) {
      ClearAllControls();
    }

    private void ExitBtn_Click(object sender, EventArgs e) {
      this.Close();
    }

    private void WarehousesGridView_CellClick(object sender, DataGridViewCellEventArgs e)
{
      if (e.RowIndex >= 0 && WarehousesGridView[0, e.RowIndex].Value.ToString() !=
_WarehousesList[0].Message) {
        _selectedRowIndex = e.RowIndex;
        UpdateWarehousesForm updateWarehousesForm = new
UpdateWarehousesForm(Convert.ToInt32(WarehousesGridView[0,
e.RowIndex].Value.ToString()));
        updateWarehousesForm.ShowDialog();
        DataLoad();
      }
    }

    private void DataLoad() {
      int firstRowIndex = 0;
      if (WarehousesGridView.FirstDisplayedScrollingRowIndex > 0) {
```

```
      firstRowIndex = WarehousesGridView.FirstDisplayedScrollingRowIndex;
    }
    try {
      _WarehousesList = _WarehousesProvider.GetAllWarehouses();
      LoadDataInWarehousesGridView(_WarehousesList);
      if (_selectedRowIndex == WarehousesGridView.Rows.Count) {
        _selectedRowIndex = WarehousesGridView.Rows.Count - 1;
      }
      if (_selectedRowIndex >= 0) {
        WarehousesGridView.FirstDisplayedScrollingRowIndex = firstRowIndex;
        WarehousesGridView.Rows[_selectedRowIndex].Selected = true;
      }
    } catch { }
  }

  private void LoadDataInWarehousesGridView(List<Warehouses> WarehousesList) {
    WarehousesGridView.DataSource = null;
    WarehousesGridView.Columns.Clear();
    WarehousesGridView.AutoGenerateColumns = false;
    WarehousesGridView.RowHeadersVisible = false;

    WarehousesGridView.DataSource = WarehousesList;

    if (WarehousesList.Count > 0) {
      if (WarehousesList[0].Message == NamesMy.NoDataNames.NoDataInWarehouses) {
        DataGridViewColumn messageColumn = new DataGridViewTextBoxColumn();
        messageColumn.DataPropertyName = "Message";
        messageColumn.Width = WarehousesGridView.Width -
NamesMy.SizeOptins.MinusSizePanel;
        WarehousesGridView.Columns.Add(messageColumn);
      } else {
        DataGridViewColumn DetailIdColumn = new DataGridViewTextBoxColumn();
        DetailIdColumn.DataPropertyName = "WarehousesId";
        WarehousesGridView.Columns.Add(DetailIdColumn);
        WarehousesGridView.Columns[0].Visible = false;

        DataGridViewColumn numberColumn = new DataGridViewTextBoxColumn();
        numberColumn.HeaderText = "№ ";
        numberColumn.DataPropertyName = "Number";
        numberColumn.DefaultCellStyle.Alignment =
DataGridViewContentAlignment.MiddleRight;
        numberColumn.Width = NamesMy.SizeOptins.NumberSize;
        WarehousesGridView.Columns.Add(numberColumn);

        DataGridViewColumn WarehousesNameColumn = new DataGridViewTextBoxColumn();
        WarehousesNameColumn.HeaderText = "Назва";
        WarehousesNameColumn.DataPropertyName = "WarehousesName";
        WarehousesNameColumn.Width = NamesMy.SizeOptins.NameSize;
        WarehousesGridView.Columns.Add(WarehousesNameColumn);

        DataGridViewColumn LocationColumn = new DataGridViewTextBoxColumn();
        LocationColumn.HeaderText = "Місцезнаходження";
```

```
        LocationColumn.DataPropertyName = "Location";
        LocationColumn.Width = NamesMy.SizeOptins.NameSize;
        WarehousesGridView.Columns.Add(LocationColumn);

      }
      for (int i = 0; i < WarehousesGridView.Columns.Count; i++) {
        WarehousesGridView.Columns[i].HeaderCell.Style.BackColor = Color.LightGray;
      }
    }
  }

  private void ClearAllControls() {
    WarehousesNameTBox.Text = String.Empty;
    DescriptionTBox.Text = String.Empty;
  }

  private bool IsDataEnteringCorrect() {
    bool isCorrect = true;
    if (_validation.IsDataEntering(WarehousesNameTBox.Text)) {
      WarehousesNameValiadtionLbl.Text = NamesMy.ProgramButtons.RequiredValidation;
    } else {
      WarehousesNameValiadtionLbl.Text = NamesMy.ProgramButtons.ErrorValidation;
      isCorrect = false;
    }
    if (_validation.IsDataEntering(LocationTBox.Text)) {
      LocationValiadtionLbl.Text = NamesMy.ProgramButtons.RequiredValidation;
    } else {
      LocationValiadtionLbl.Text = NamesMy.ProgramButtons.ErrorValidation;
      isCorrect = false;
    }
    return isCorrect;
  }

 }
}
```

Лістинг 4. Код класу «UpdateWarehousesForm»

```
using ProductAccounting.AppCode;
using ProductAccounting.Providers;
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace ProductAccounting.Forms.Dictionary {
 public partial class UpdateWarehousesForm : Form {
  private int _WarehousesId = 0;
```

```
    private Warehouses _selectedWarehouses = new Warehouses();
    private WarehousesProvider _WarehousesProvider = new WarehousesProvider();
    private ValidationMy _Validation = new ValidationMy();

    public UpdateWarehousesForm(int WarehousesId) {
      InitializeComponent();
      _WarehousesId = WarehousesId;
      LoadAllDate();
    }

    private void SaveBtn_Click(object sender, EventArgs e) {
      if (IsDataEnteringCorrect()) {
        _WarehousesProvider.UpdateWarehouses(WarehousesNameTBox.Text,
LocationTBox.Text, DescriptionTBox.Text, _WarehousesId);
        this.Close();
      }
    }

    private void DeleteBtn_Click(object sender, EventArgs e) {
      if (MessageBox.Show("Ви дійсно хочете видалити цей елемент?", "Видалити",
MessageBoxButtons.YesNo) == DialogResult.Yes) {
        _WarehousesProvider.DeleteWarehouses(_WarehousesId);
        this.Close();
      }
    }

    private void ExitBtn_Click(object sender, EventArgs e) {
      this.Close();
    }

    private void LoadAllDate() {
      _selectedWarehouses =
_WarehousesProvider.SelectedWarehousesByWarehousesId(_WarehousesId);
      WarehousesNameTBox.Text = _selectedWarehouses.WarehousesName;
      LocationTBox.Text = _selectedWarehouses.Location;
      DescriptionTBox.Text = _selectedWarehouses.Description;
    }

    private bool IsDataEnteringCorrect() {
      bool isCorrect = true;
      if (_Validation.IsDataEntering(WarehousesNameTBox.Text)) {
        WarehousesNameValiadtionLbl.Text = NamesMy.ProgramButtons.RequiredValidation;
      } else {
        WarehousesNameValiadtionLbl.Text = NamesMy.ProgramButtons.ErrorValidation;
        isCorrect = false;
      }
      if (_Validation.IsDataEntering(LocationTBox.Text)) {
        LocationValiadtionLbl.Text = NamesMy.ProgramButtons.RequiredValidation;
      } else {
        LocationValiadtionLbl.Text = NamesMy.ProgramButtons.ErrorValidation;
        isCorrect = false;
      }
```

```
    return isCorrect;
   }

  }
}
```

Лістинг 5. Код класу «SearchProductForm»

```
using ProductAccounting.AppCode;
using ProductAccounting.Providers;
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace ProductAccounting.Forms.Search {
  public partial class SearchProductForm : Form {
    private SearchBLL _SearchBLL = new SearchBLL();
    private List<ProductSearch> _ProductSearch = new List<ProductSearch>();
    private ValidationMy _validation = new ValidationMy();
    private WarehousesProvider _WarehousesProvider = new WarehousesProvider();
    private List<Warehouses> _WarehousesL = new List<Warehouses>();

    public SearchProductForm() {
      InitializeComponent();
      LoadWarehouses();
    }

    private void LoadWarehouses() {
      _WarehousesL = _WarehousesProvider.GetAllWarehouses();
      WarehousesCBox.DataSource = _WarehousesL;
      WarehousesCBox.ValueMember = "WarehousesId";
      WarehousesCBox.DisplayMember = "WarehousesName";
    }

    private void ProductNameSearchBtn_Click(object sender, EventArgs e) {
      if (_validation.IsDataEntering(ProductNameTBox.Text)) {
        ProductNameValiadtionLbl.Text = NamesMy.ProgramButtons.RequiredValidation;
        _ProductSearch = _SearchBLL.GetProductsByName(ProductNameTBox.Text);
        DisplayProductSearchResults(_ProductSearch);
      } else {
        ProductNameValiadtionLbl.Text = NamesMy.ProgramButtons.ErrorValidation;
      }
    }

    private void ModelSearchBtn_Click(object sender, EventArgs e) {
      if (_validation.IsDataEntering(ModelTBox.Text)) {
```

```
        ModelValiadtionLbl.Text = NamesMy.ProgramButtons.RequiredValidation;
        _ProductSearch = _SearchBLL.GetProductsByModel(ModelTBox.Text);
        DisplayProductSearchResults(_ProductSearch);
      } else {
        ModelValiadtionLbl.Text = NamesMy.ProgramButtons.ErrorValidation;
      }
    }

    private void ManufacturerSearchBtn_Click(object sender, EventArgs e) {
      if (_validation.IsDataEntering(ManufacturerTBox.Text)) {
        ManufacturerValiadtionLbl.Text = NamesMy.ProgramButtons.RequiredValidation;
        _ProductSearch = _SearchBLL.GetProductsByManufacturer(ManufacturerTBox.Text);
        DisplayProductSearchResults(_ProductSearch);
      } else {
        ManufacturerValiadtionLbl.Text = NamesMy.ProgramButtons.ErrorValidation;
      }
    }

    private void WarehousesSearchBtn_Click(object sender, EventArgs e) {
      if (Convert.ToInt32(WarehousesCBox.SelectedValue) > 0) {
        WarehousesValiadtionLbl.Text = NamesMy.ProgramButtons.RequiredValidation;
        _ProductSearch =
_SearchBLL.GetProductsByWarehouseId(Convert.ToInt32(WarehousesCBox.SelectedValue));
        DisplayProductSearchResults(_ProductSearch);
      } else {
        WarehousesValiadtionLbl.Text = NamesMy.ProgramButtons.ErrorValidation;
      }
    }

    private void SalePriceSearchBtn_Click(object sender, EventArgs e) {
      if (_validation.IsDataConvertToDouble(SalePriceTBox.Text)) {
        SalePriceValiadtionLbl.Text = NamesMy.ProgramButtons.RequiredValidation;
        _ProductSearch =
_SearchBLL.GetProductsByMaxPrice(Convert.ToDouble(SalePriceTBox.Text));
        DisplayProductSearchResults(_ProductSearch);
      } else {
        SalePriceValiadtionLbl.Text = NamesMy.ProgramButtons.ErrorValidation;
      }
    }

    private void QuantitySearchBtn_Click(object sender, EventArgs e) {
      if (_validation.IsDataConvertToInt(QuantityTBox.Text)) {
        QuantityValiadtionLbl.Text = NamesMy.ProgramButtons.RequiredValidation;
        _ProductSearch =
_SearchBLL.GetProductsByMaxQuantity(Convert.ToInt32(QuantityTBox.Text));
        DisplayProductSearchResults(_ProductSearch);
      } else {
        QuantityValiadtionLbl.Text = NamesMy.ProgramButtons.ErrorValidation;
      }
    }
```

```
    private void DisplayProductSearchResults(List<ProductSearch> products) {
      ProductSearchResultsTBox.Clear();
      int orderNumber = 1;
      foreach (var product in products) {
        ProductSearchResultsTBox.AppendText(product.ToString(orderNumber) +
Environment.NewLine + Environment.NewLine);
        orderNumber++;
      }
    }

  }
}
```

ДОДАТОК Б. ДЕМОНСТРАЦІЙНИЙ МАТЕРІАЛ

Формат	Позиція	ПОЗНАЧЕННЯ	НАЙМЕНУВАННЯ	Кільк.	Примітки
			Текстові документи		
А4	1	ГЮІК. 442351.013 ПЗ	Пояснювальна записка	90	с.
			Інші документи		
	2		Демонстраційний матеріал у	13	с.
			вигляді презентації		

					ГЮІК. 442351.013 ПЗ			
Змін.	Арк.	Номер докум.	Підпис	Дата				
Розроб.		Трохін			Розробка інформаційно-пошукової системи обліку продукції для автоматизованої багаторівневої внутрішньо-складської підсистеми зберігання *Відомість кваліфікаційної роботи*	Літера	Аркуш	Аркушів
Перевір.		Чала				Н	1	1
						Кафедра КІТАР ХНУРЕ		
Н.контр.		Бронніков						
Затв.		Невлюдов						