НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

КАФЕДРА ТЕХНІЧНОЇ КІБЕРНЕТИКИ

Курсова робота

Розробка АСУ «Orange»

3 дисципліни «Об’єктно-орієнтоване програмування - 2»

Керівник Виконавець

Хмелюк Володимир Сергійович ст. Войчишин Олександр

Олександрович

«Допущений до захисту» зал. книжка № ІТ-5205

гр. ІТ-52

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

(особистий підпис керівника) (особистий підпис виконавця)

«\_\_» \_\_\_\_\_\_\_\_\_\_\_\_2017р. «\_\_» \_\_\_\_\_\_\_\_\_\_\_\_2017р.

Захищений з оцінкою

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

(оцінка)

Київ – 2017

АНОТАЦІЯ

У цьому проекті буде розроблено автоматизовану систему управління «Orange», обґрунтується актуальність теми; розкриється найменування і призначення області застосування програми, що розробляється в курсовій роботі. Будуть розглянуті нові рішення й методи розв'язан­ня задачі, обґрунтується вибір конкретного методу та сформується мате­матична модель й загальна структура програми. Також будуть наведені інструкції до програми, вказівки програміста, висновки.

Дана пояснювальна записка містить 18 сторінок, а також 5 скріншотів  
роботи програми та її використання і діаграму класів.

ЗМІСТ

ВСТУП…………………………………………………………………………...1

1 ПОСТАНОВКА ЗАДАЧІ……………………………………………………..2

2 МАТЕМАТИЧНА МОДЕЛЬ ТА СТРУКТУРА ПРОГРАМИ……………...3

3 ПЕРЕЛІК І ПРИЗНАЧЕННЯ РЕЖИМІВ ТА СТРУКТУРА ДІАЛОГУ…...4

4 СТРУКТУРА ДАНИХ ТА РЕСУРСІВ ПРОГРАМИ…………………….…7

5 ОПИС ПРОГРАМИ………………………………………………………..….9

6 ІНСТРУКЦІЯ АДМІНІСТРАТОРА………………………………………....11

7 КЕРІВНИЦТВО КОРИСТУВАЧА…………………………………………..13

ВИСНОВКИ……………………………………………………………………..15

ВИКОРИСТАНІ ДЖЕРЕЛА…………………………………………………...16

ДОДАТОК А. Текст програми…………………………………………………17

ДОДАТОК Б. Текст програми…………………………………………………31

ВСТУП

Актуальність теми. На сьогоднішній день люди працюють з касовим апаратом, що надає змогу: слідкувати за попитом проданого товару, кількістю певного товару, друкувати чек який дає право на обмін товару. Близько 92% люди використовують касовий апарат. Виходячи з цього можна сказати – вибрана тема роботи є актуальною.

Мета роботи. Основною метою роботи є вивчення ООП та його основних понять (*Наслідування*, *Поліморфізм*, *Інкапсуляція*). Також навчитися працювати з базою даних.

1 ПОСТАНОВКА ЗАДАЧІ

Ціллю цієї курсової роботи розробка АСУ «Orange» для зручної роботи з продажі товару. Є можливість додати новий товар так само є можливість його вилучити.

Програма повинна забезпечувати виконання таких функцій:

* додавання нового товару;
* видалення певного товару;
* вивід списку продуктів;
* збереження чеків;
* додавання даних нового користувача цієї програми;
* видалення продукту якщо він більше не продаватиметься.

Виходячи з цілі потрібно вирішити такі проблеми:

* розробити алгоритм додавання нових продуктів ;
* налаштувати взаємодію з базою даних[1];
* створити UML[2] - діаграми;
* закодувати програму;
* провести тестування;

2 МАТЕМАТИЧНА МОДЕЛЬ ТА СТРУКТУРА ПРОГРАМИ

В даній лабораторній роботі ми маємо головне меню в якому знаходиться три кнопки, при натисканні на які відкриваються відповідні форми. Кожна з яких повинна містити інформацію відповідно до тематики. Для створення цих елементів будуть використовуватися Windows Form[3]. Він чудово підходить для даної теми так як він є не віддільним від коду. Ми можемо оформити його зручно для користувача який в подальшому буде використовувати дану програму.

Існує безліч різних систем управління базами даних (СУБД). В цьому проекті в якості бази даних було вирішено обрати [Acces](#_1_System.Drawing_-)s. Причинами цього є зручне і швидке редагування таблиць бази даних. Також на сайті розробника є повна [документація[4]](#MySQLdocs) .

Для виведення інформації на форму буде використовуватись елемент DataGridView. Він надає потужний і гнучкий спосіб відображення даних в табличному форматі. DataGridView можна використовувати для відображення даних невеликого обсягу тільки для читання, або можна масштабувати його для відображення редагованих дуже великих наборів даних. Для того щоб реалізувати визначену поведінку в додатках, його можна розширити декількома способами. Наприклад, можна програмно задати власні алгоритми сортування, можна створити власні типи комірок. Зовнішній вигляд елемента управління DataGridView легко налаштовується встановленням значень декількох властивостей. Як джерело даних можуть використовуватися сховища даних різних типів, також він може працювати без прив'язаного до нього джерела даних.

Так як покупець може відмінити покупку певного товару до того моменту як за нього заплатить було зроблено метод видалення певного товару який виділявся курсором мишки.

3 ПЕРЕЛІК І ПРИЗНАЧЕННЯ РЕЖИМІВ, СТРУКТУРА ДІАЛОГУ

Програма має тільки один режим роботи, а все управління здійснюється за допомогою елементів керування.

Проектування інтерфейсу дуже важливе, він повинен бути простим і зрозумілим для користувача, забезпечувати швидкий доступ до всіх інструментів і містити інформацію про поточний стан програми.

Головне вікно програми буде містити напис – назву програми, а також три кнопки «Продаж», «Список продуктів», «Вихід з профілю» при натисканні на які можна буде перейти на відповідні форми (Рисунок 1).

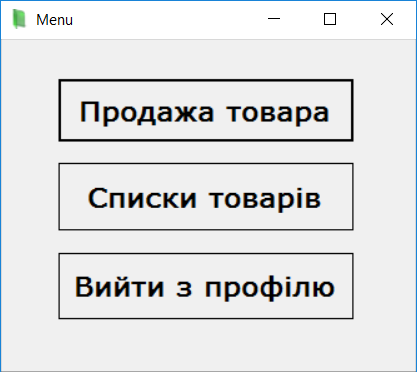


Рисунок 1 – Головна форма клієнта

Форма «Вийти з профілю» матиме перехід на перше спливаюче вікно яке просить нас авторизуватися.

Форма «Продаж» буде мати: кнопку нове замовлення, поле для імені касира, назва товару на продажу, поле для ведення кількості певного товару на продажу, загальна та з ПДВ сума (Рисунок 2).

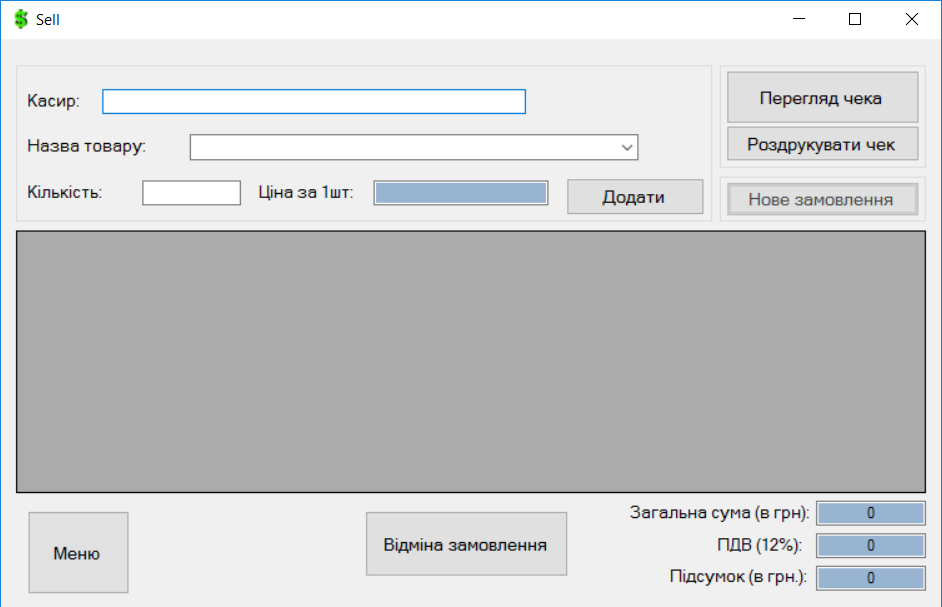


Рисунок 2 – Форма для продажу товару

Форма «Списки товарів» матиме: весь наш товар який ми можемо редагувати (видаляти, помічати як в даний момент відсутній), добавляти новий товар.

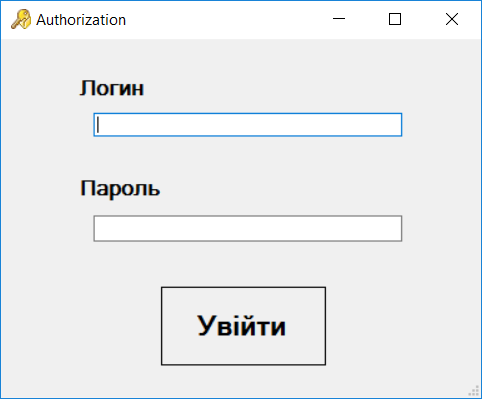


Рисунок 3 – Форма авторизації

Форма «Авторизація» матиме: два поля для ведення даних (логін та пароль) (Рисунок 3).

4 СТРУКТУРА ДАНИХ ТА РЕСУРСІВ ПРОГРАМИ

4.1 Класи авторизації

Клас «Authorization» - точка запуску першої форми.

Методи класу:

* Sign\_in – в ньому знаходиться логін та пароль.
* Authorization\_FormClosing – метод в якому при не коректному воді визивається помилка.
* Registration – цей метод викликається при реєстрації користувача.

4.2 Класи меню

Клас «Menu» - це клас головного меню в якому знаходяться переходи між формами та як основний запуск програми.

Методи класу:

* Sells – метод який запускає форму продажу товару.
* ExitFormAccount – цей метод виходить з профілю та відкриває форму авторизації.
* See – цей метод викликає форму редагування.

4.3 Класи продажу товару

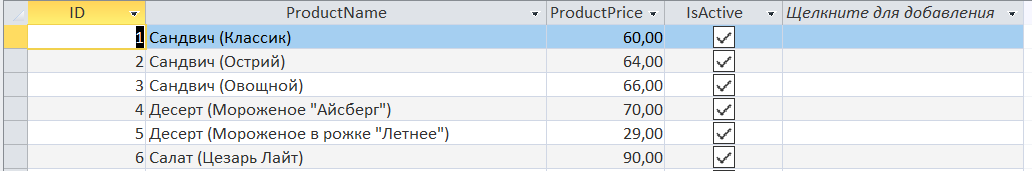
Клас «Sells» - це клас для продажу товарів.

Методи класу:

* CancelOrderButton – відміна заказу, перехід в режим читання.
* NewOrderButton – добавление нового заказу.
* AddToCarButton – додавання та підсумок суми товарів.
* CartDataGridView – виділення певного рядка який буде видалятися.
* DeleteToolStripMenuItem – видалення та підрахунок загальної суми без видаленого товару.
* PrintDocument1 – зміст документа.
* PrintPreivButton – перегляд попередніх чеків.
* PrintOrderButton – друкування чеку.
* ItemNameComboBox – перерахування всього товару.
* CloseSells – закриття форми продажу товару.

4.4 База даних

Для зчитування товару програма використовую базу даних.

 Рисунок 4- Таблиця бази даних

База даних містить 4 елементів:

* ID – в цьому полі знаходиться унікальній номер товару.
* ProductName – в цьому полі знаходиться назва товару.
* ProductPrice – в цьому полі знаходиться сума товару.
* IsActive – присутній товар.

5 ОПИС ПРОГРАМИ

Робота програми в першу чергу починається з запуску «Авторизації» після якої відкриває «Меню» в якому знаходиться три кнопки. Кожна з них відкриває відповідні форми. На цих формах можна додати, змінити, видалити відповідну інформацію.

Подія RegistrationImg\_MouseClick викликає форму «Журнал». Після її відкриття головне вікно програми стає недоступним.

Подія sign\_in\_Click викликає форму «Меню» . Після її відкриття користувач знаходиться в головному меню.

Подія Sells\_Click викликає форму «Продаж». Після її відкриття головне вікно програми стає недоступним.

Подія See\_Click викликає форму «Списки товарів» . Після її відкриття головне вікно програми стає недоступним.

Подія ExitFromAccount\_Click викликає форму «Авторизації». Після її відкриття головне вікно програми стає недоступним.

Кнопка Info відкриває додаткову форму з інструкціями . Розміщена на кажній формі біля кнопки Back.

Подія ProductAddToBOT\_MouseClick відкриває невеличке вікно в якому можна ввести інформацію нового елемента.

Подія CancelOrderButton\_Click відміняє весь заказ.

Подія NewOrderButton\_Click додавання товару в чек.

Подія AddToCartButton\_Click додавання суми товару до основної та моментальний підрахунок з ПДВ.

Подія CartDataGridView\_MouseDown вибір видалення вибраної строки.

Подія deleteToolStripMenuItem\_Click видалення.

Подія printDocument1\_PrintPage зміст документа.

Подія PrintPreviewButton\_Click Попередній перегляд чека

Подія ItemNameComboBox\_SelectedValueChanged зчитування з бази даних товарів.

Подія Close\_Click закриття даної форми та відкриття меню.

Подія CancelOrderButton\_Click відміна заказу, режим читання.

6 КЕРІВНИЦТВО АДМІНІСТРАТОРА

Назва програми «Orange». ЇЇ призначення спрощення продажу товару в кав’ярнях та продуктових магазинів, видача чеку та зберігання їх в нашій пам’яті. Вона написана на мові програмування C# в середовищі розробки Microsoft Visual Studio 2017. Периферійні пристрої використання під час проектування: миша, тачпад, клавіатура.

Перед початком кодування програми потрібно завантажити середовище розробки з сайту <https://www.visualstudio.com>. Встановлення Visual Studio повинне виконуватися від імені облікового запису адміністратора. Однак, після встановлення Visual Studio ці облікові дані не потрібні. Якщо на комп'ютері встановлена ​​ попередня версія Visual Studio, видаліть її. Вимоги середовища розробки до обладнання можна переглянути на сторінці по посиланню [8].

Після встановлення потрібно налаштувати Visual Studio. Можна налаштувати кольори, шрифти, меню, панелі інструментів, розташування вікон і поєднання клавіш, а також створювати шаблони, використовувати зовнішні засоби і управляти розширеннями. Будь-які налаштування Visual Studio можна зробити доступними для інших користувачів або скинути до значень за замовчуванням.

Якщо закрити і повторно відкрити Visual Studio, задані параметри збережуться, оскільки вони автоматично зберігаються в окремому файлі.

Рекомендовані вимоги для стабільної роботи серверу:

(характеристики розраховані на сервер для 200 клієнтів)

* Процесор з частотою 1,6 ГГц або більш потужний.
* 1 гігабайт ( GB ) оперативної пам'яті.
* Мережева плата та доступ до інтернету.
* 200 мегабайт доступного простору на жорсткому диску.
* Жорсткий диск з частотою обертання 5400 об/Хв.

Для запуску програми потрібно скопіювати собі на комп’ютер папку «Orange» і запустити файл «Orange.exe».

Рекомендовані вимоги для стабільної роботи клієнту:

* Процесор з частотою 1,6 ГГц або більш потужний.
* 512 мегабайт ( MB ) оперативної пам'яті.
* Мережева плата та доступ до інтернету.
* 10 мегабайт доступного простору на жорсткому диску.
* Жорсткий диск з частотою обертання 5400 об/Хв.
* Відеоадаптер з підтримкою DirectX 9 і розширенням екрану не менше 1024 x 768.

Для запуску програми потрібно скопіювати собі на комп’ютер папку «Orange» і запустити файл «Orange.exe».

7 КЕРІВНИЦТВО КОРИСТУВАЧА

Назва програми «Orange». Вона слугує для ведення товару та зберігання проданого товару.

Для роботи програми потрібна операційна система Windows і встановлений .NET Framework версій 4.5 і вище. Вона не потребує встановлення додаткових драйверів чи бібліотек. Програма не потребує великих ресурсів від ПК: кількість оперативної пам’яті 1 GB і вище, (Windows XP) чи більш новішу операційну систему, і місце на жорсткому диску 25 Mb. Щоб використовувати можливості програми потрібні периферійні пристрої, а саме: миша (тачпад), клавіатура.

Для запуску потрібно скопіювати собі на комп’ютер папку «Client» і запустити файл «Orange.exe». Щоб видалити програму потрібно видалити папку «Orange».

Основні можливості програми та їх використання:

* Щоб авторизуватись в програмі необхідно вписати свій логін та пароль на формі Authorization, та натиснути кнопку Вхід.
* Щоб зареєструватися потрібно вписати в логін «NewUser», після цього користувача пере направить на форму реєстрації після чого він зможе війти в меню та інші форми.
* На форму редагування можна добавляти нові товари і так само їх видаляти або робити в даний час не активними (відсутність в налички), можна задавати вартість одного товару.
* На формі меню ми маємо доступ до всіх форм які нам потрібні. Сучасний дизайн форм та приязний на очі.
* На формі продажу товару ми оформлюємо товар на продажу. Ми натискаємо на кнопку новий чек. Ми вписуємо ім’я касира в перший рядок (при не виконанні цього користувача повідомить програма), ми можемо швидко знайти так само і вибрати певний товар на продажу, після чого нас в рядку кількість товару вписуємо ту кількість яку повідомив покупець. Після цих дій ми добавляємо товар натиснувши на клавішу добавити. Якщо покупець повідомляє до сплати товару, що він хоче відмовитись від певного товару ми його виділяємо курсором мишки та видаляємо за допомогою правої клавіші мишки. Після того як покупець закінчив вибір ми повідомляємо всю суму та друкуємо йому чек та копія того чека зберігається у нас в файлах.
* Для виходу з програми ми натискаємо на кнопку яка знаходиться на формі продажу товару повернутися в головне меню та ми натискаємо на червоний хрест який знаходиться у верхньому правому кутку.

ВИСНОВКИ

В результаті роботи було вивчено багато нового матеріалу для програмування мовою C#, так як робота з АСУ дає навички програмування. Створено функціональну систему яку можна використовувати в кав’ярнях та продуктових магазинах. Її інтерфейс є зручним і надає швидкий доступ до всіх інструментів, тестування показало, що програма коректно виконує свою роботу. Це дозволяє зробити висновок, що поставлена ціль досягнута.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

Книги:

1. Robert C. Martin «Clean Code» [Classes] 2008 – 135 с.
2. UML діаграми  
   Learning UML 2.0  
   Author: Russ Miles
3. WindowsForm – режим доступу  
   WindowsForms in Action: Second Edition of Windows Forms  
   Author: Erik Brown
4. Access база даних  
   Acces 2013 Bible   
   Author: Майкл Олександр, Дік Куслей

Мережеві ресурси:

1. Системні вимоги для Visual Studio. – Режим доступу: [https://www.visualstudio.com/en-us/productinfo/vs2015-sysrequirements-vs#27](https://www.visualstudio.com/en-us/productinfo/vs2015-sysrequirements-vs%2327)

# Windows Forms – Режим доступу:

<https://ru.wikipedia.org/wiki/Windows_Forms>

ДОДАТОК А

Код програми

static class Program

{

public static Menu FormMenue;

[STAThread]

static void Main()

{

Application.EnableVisualStyles();

Application.SetCompatibleTextRenderingDefault(false);

FormMenue = new Menu();

Application.Run(FormMenue);

}

}

public partial class Authorization : Form

{

public Authorization()

{

InitializeComponent();

}

private void sign\_in\_Click(object sender, EventArgs e)

{

if (Login.Text == "admin" && Password.Text == "admin")

{

this.Hide();

}

else

{

Login.Text = "";

Password.Text = "";

MessageBox.Show("Login or Password is not true");

}

}

private void Authorization\_FormClosing(object sender, FormClosingEventArgs e)

{

Environment.Exit(0);

} }

public partial class Menu : Form

{

public static Authorization FormAuthorization;

public static Sells FormSells;

public Menu()

{

InitializeComponent();

FormSells = new Sells();

FormAuthorization = new Authorization();

FormAuthorization.ShowDialog();

}

private void Sells\_Click(object sender, EventArgs e)

{

this.Visible = false;

FormSells.Show();

}

private void ExitFromAccount\_Click(object sender, EventArgs e)

{

this.Visible = false;

FormAuthorization.ShowDialog();

}

private void See\_Click(object sender, EventArgs e)

{

}

}

public partial class Sells : Form

{

public Sells()

{

InitializeComponent();

}

private List<CartItem> shoppingCart = new List<CartItem>();

private int numberOfItemsPerPage = 0;

private int numberOfItemsPrintedSoFar = 0;

private void Form1\_Load(object sender, EventArgs e)

{

ItemNameComboBox.SelectedValueChanged -= ItemNameComboBox\_SelectedValueChanged;

ItemNameComboBox.DataSource = GetProductName();

ItemNameComboBox.DisplayMember = "ProductName";

ItemNameComboBox.ValueMember = "ProductPrice";

ItemNameComboBox.SelectedIndex = -1;

ItemNameComboBox.SelectedValueChanged += ItemNameComboBox\_SelectedValueChanged;

}

private DataTable GetProductName()

{

DataTable dtProduct = new DataTable();

string connSrtring = ConfigurationManager.ConnectionStrings["dbx"].ConnectionString;

using (OleDbConnection con = new OleDbConnection(connSrtring))

{

using (OleDbCommand cmd = new OleDbCommand("SELECT ProductName, ProductPrice FROM Products WHERE IsActive = true", con))

{

con.Open();

OleDbDataReader reader = cmd.ExecuteReader();

dtProduct.Load(reader);

}

}

return dtProduct;

}

private void CancelOrderButton\_Click(object sender, EventArgs e)//отмена заказа, режим чтения

{

NewOrderButton.Enabled = true;

PrintOrderButton.Enabled = false;

PrintPreviewButton.Enabled = false;

CancelOrderButton.Enabled = false;

ItemGroupBox.Enabled = false;

ClientNameTextBox.Clear();

//возврат

ItemNameComboBox.SelectedIndex = -1;

QuantityTextBox.Clear();

UnitPriceTextBox.Clear();

TotalAmountextBox.Text = "0";

TotalSalesTexBox.Text = "0";

TotalToPayBox.Text = "0";

CartDataGridView.DataSource = null;

shoppingCart.Clear();

}

private void NewOrderButton\_Click(object sender, EventArgs e)//добавление нового заказа, режим ввода

{

NewOrderButton.Enabled = false;

PrintOrderButton.Enabled = true;

PrintPreviewButton.Enabled = true;

CancelOrderButton.Enabled = true;

ItemGroupBox.Enabled = true;

ClientNameTextBox.Focus();

}

private void AddToCartButton\_Click(object sender, EventArgs e)

{

if (IsValidated())

{

CartItem item = new CartItem()

{

ItemName = ItemNameComboBox.Text,

Quantity = Convert.ToInt16(QuantityTextBox.Text.Trim()),

UnitPrice = Convert.ToDecimal(UnitPriceTextBox.Text.Trim()),

TotalPrice = Convert.ToInt16(QuantityTextBox.Text.Trim()) \* Convert.ToDecimal(UnitPriceTextBox.Text.Trim())

};

// Array, Collection, List, DataTable and DataSet

shoppingCart.Add(item);

CartDataGridView.DataSource = null;

CartDataGridView.DataSource = shoppingCart;

//Статус общая сумма (в грн.)

decimal totalAmount = shoppingCart.Sum(x => x.TotalPrice);

TotalAmountextBox.Text = totalAmount.ToString();

//Статус ПДВ(12%)

decimal totalSalesTex = (12 \* totalAmount) / 100;

TotalSalesTexBox.Text = totalSalesTex.ToString();

//Статус Итог (в грн.)

decimal totalToPay = totalAmount + totalSalesTex;

TotalToPayBox.Text = totalToPay.ToString();

ItemNameComboBox.SelectedIndex = -1;

QuantityTextBox.Clear();

UnitPriceTextBox.Clear();

}

}

private bool IsValidated()

{

if (ClientNameTextBox.Text.Trim() == string.Empty)//если имя клиента не введено

{

MessageBox.Show("Введіть ПІБ касира!", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);

ClientNameTextBox.Focus();

return false;

}

if (ItemNameComboBox.SelectedIndex == -1)//если имя товара не введено

{

MessageBox.Show("Виберіть товар!", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);

return false;

}

if (QuantityTextBox.Text.Trim() == string.Empty)//если количество не введено

{

MessageBox.Show("Введіть кількість!", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);

QuantityTextBox.Focus();

return false;

}

else

{

int tempQuantity;

bool isNumeric = int.TryParse(QuantityTextBox.Text.Trim(), out tempQuantity);// защита от некорректного ввода количества, должно вводиться int значение

if (!isNumeric)// если не верно int знаечение

{

MessageBox.Show("Кількість повинна бути цілим числом!", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);

QuantityTextBox.Clear();

QuantityTextBox.Focus();

return false;

}

}

if (UnitPriceTextBox.Text.Trim() == string.Empty)

{

MessageBox.Show("Введіть ціну за одиницю товару!", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);

UnitPriceTextBox.Focus();

return false;

}

else

{

decimal tempUnitPrice;

bool isDecimal = decimal.TryParse(UnitPriceTextBox.Text.Trim(), out tempUnitPrice);// защита от некорректного ввода цену за единицу товара, должно вводиться decimal значение

if (!isDecimal)// если не верно decimal знаечение

{

MessageBox.Show("Ціна повинна бути десяткової!", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);

UnitPriceTextBox.Clear();

UnitPriceTextBox.Focus();

return false;

}

return true;

}

}

private void CartDataGridView\_MouseDown(object sender, MouseEventArgs e)//выбор удаляемой строки

{

if (e.Button == System.Windows.Forms.MouseButtons.Right)

{

var hti = CartDataGridView.HitTest(e.X, e.Y);

CartDataGridView.Rows[hti.RowIndex].Selected = true;

contextMenuStrip1.Show(CartDataGridView, e.X, e.Y);

}

}

private void deleteToolStripMenuItem\_Click(object sender, EventArgs e)//удаление

{

int index = CartDataGridView.CurrentCell.RowIndex;

shoppingCart.RemoveAt(index);

CartDataGridView.DataSource = null;

CartDataGridView.DataSource = shoppingCart;

//Статус общая сумма (в грн.)

decimal totalAmount = shoppingCart.Sum(x => x.TotalPrice);

TotalAmountextBox.Text = totalAmount.ToString();

//Статус НДС(18%)

decimal totalSalesTex = (18 \* totalAmount) / 100;

TotalSalesTexBox.Text = totalSalesTex.ToString();

//Статус Итог (в грн.)

decimal totalToPay = totalAmount + totalSalesTex;

TotalToPayBox.Text = totalToPay.ToString();

}

private void printDocument1\_PrintPage(object sender, System.Drawing.Printing.PrintPageEventArgs e)//содержание документ

{

e.Graphics.DrawString("Касир: " + ClientNameTextBox.Text.Trim(), new Font("Arial", 16, FontStyle.Regular), Brushes.Black, new Point(25, 50));

e.Graphics.DrawString("Дата и время: " + DateTime.Now, new Font("Arial", 12, FontStyle.Regular), Brushes.Black, new Point(25, 80));

e.Graphics.DrawString("------------------------------------------------------------------------------------------------------------------------------------", new Font("Arial", 12, FontStyle.Regular), Brushes.Black, new Point(25, 100));

e.Graphics.DrawString("Назва товару", new Font("Arial", 12, FontStyle.Regular), Brushes.Black, new Point(30, 120));

e.Graphics.DrawString("Кількість", new Font("Arial", 12, FontStyle.Regular), Brushes.Black, new Point(450, 120));

e.Graphics.DrawString("Ціна за 1шт.", new Font("Arial", 12, FontStyle.Regular), Brushes.Black, new Point(580, 120));

e.Graphics.DrawString("Підсумок", new Font("Arial", 12, FontStyle.Regular), Brushes.Black, new Point(720, 120));

e.Graphics.DrawString("------------------------------------------------------------------------------------------------------------------------------------", new Font("Arial", 12, FontStyle.Regular), Brushes.Black, new Point(25, 140));

int yPos = 165;

for (int i = numberOfItemsPrintedSoFar; i < shoppingCart.Count; i++)

{

numberOfItemsPerPage++;

if (numberOfItemsPerPage <= 25)

{

numberOfItemsPrintedSoFar++;

if (numberOfItemsPrintedSoFar <= shoppingCart.Count)

{

e.Graphics.DrawString(shoppingCart[i].ItemName, new Font("Arial", 12, FontStyle.Regular), Brushes.Black, new Point(30, yPos));

e.Graphics.DrawString(shoppingCart[i].Quantity.ToString(), new Font("Arial", 12, FontStyle.Regular), Brushes.Black, new Point(490, yPos));

e.Graphics.DrawString(shoppingCart[i].UnitPrice.ToString(), new Font("Arial", 12, FontStyle.Regular), Brushes.Black, new Point(615, yPos));

e.Graphics.DrawString(shoppingCart[i].TotalPrice.ToString(), new Font("Arial", 12, FontStyle.Regular), Brushes.Black, new Point(720, yPos));

yPos += 30;

}

else

{

e.HasMorePages = false;

}

}

else

{

numberOfItemsPerPage = 0;

e.HasMorePages = true;

return;

}

}

e.Graphics.DrawString("------------------------------------------------------------------------------------------------------------------------------------", new Font("Arial", 12, FontStyle.Regular), Brushes.Black, new Point(25, yPos));

e.Graphics.DrawString("Загальна сума: " + TotalAmountextBox.Text.Trim(), new Font("Arial", 12, FontStyle.Regular), Brushes.Black, new Point(575, yPos + 30));

e.Graphics.DrawString("ПДВ 12%: " + TotalSalesTexBox.Text.Trim(), new Font("Arial", 12, FontStyle.Regular), Brushes.Black, new Point(575, yPos + 60));

e.Graphics.DrawString("------------------------------------------------------------------------------------------------------------------------------------", new Font("Arial", 12, FontStyle.Regular), Brushes.Black, new Point(25, yPos + 90));

e.Graphics.DrawString("Підсумок: " + TotalToPayBox.Text.Trim(), new Font("Arial", 12, FontStyle.Regular), Brushes.Black, new Point(575, yPos + 120));

//Сброс переменных

numberOfItemsPerPage = 0;

numberOfItemsPrintedSoFar = 0;

}

private void PrintPreviewButton\_Click(object sender, EventArgs e)//Попередній перегляд чека

{

printPreviewDialog1.Document = printDocument1;

printPreviewDialog1.ShowDialog();

}

private void PrintOrderButton\_Click(object sender, EventArgs e)

{

printDocument1.Print();

}

private void ItemNameComboBox\_SelectedValueChanged(object sender, EventArgs e)

{

decimal productPrice = Convert.ToDecimal(ItemNameComboBox.SelectedValue);

UnitPriceTextBox.Text = productPrice.ToString();

}

private void CloseSells\_Click(object sender, EventArgs e)

{

Close();

Program.FormMenue.Visible = true;

}

}

public class CartItem

{

[DisplayName("Название товара")]

public string ItemName { get; set; }

[DisplayName("Количество")]

public int Quantity { get; set; }

[DisplayName("Цена за 1 шт.")]

public decimal UnitPrice { get; set; }

[DisplayName("Итог")]

public decimal TotalPrice { get; set; }

}

ДОДАТОК Б

ДІАГРАМА

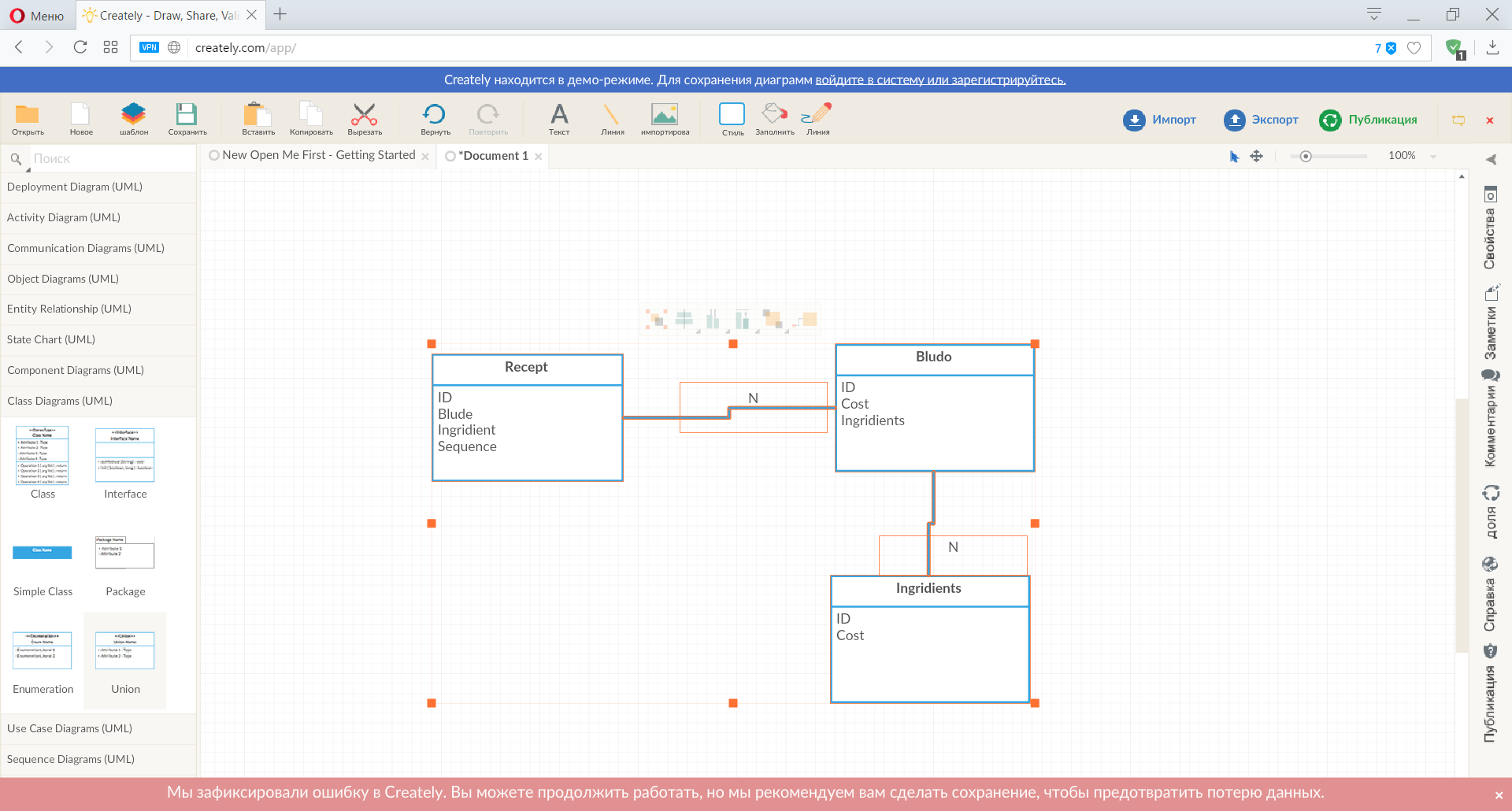


Рисунок 5 – діаграма класів