Міністерство освіти і науки України

Херсонський політехнічний фаховий коледж

Державного університету “Одеська політехніка”

# ПРОГРАМНИЙ ДОДАТОК

«Облік Та аналіз товарів в господарських супермаркетах»

Пояснювальна записка до курсового проекту

ХПФК.000352 - 23 81 01

Аркушів - 64

**В курсовому проекті використано ідеї, результати і тексти інших авторів та є посилання на відповідні джерела**

Керівник проекту

Т.Є. Багмет

.05.2022

Розробив студент

С.О. Тоцький

.05.2022

Нормоконтролер

С.О. Бабикін

.05.2022

ЗМІСТ

С.

ВСТУП 4

2 ПОСТАНОВКА ЗАДАЧІ 5

2.1 Характеристика предметної області 5

2.2 Вимоги до програми 6

2.3 Структура вхідних даних 7

2.4 Структура вихідних даних 8

3 КОМП’ЮТЕРНА СИСТЕМА 9

3.1 Технічні характеристики комп’ютера та зовнішніх пристроїв 9

3.2 Вибір програмних засобів та операційної системи 9

4 ОБ’ЄКТНО-ОРІЄНТОВАНЕ ПРОЕКТУВАННЯ 12

4.1 Інтерфейс програми 12

4.2 Алгоритм рішення задачі 13

5 ПРОГРАМУВАННЯ ТА ТЕСТУВАННЯ 16

5.1 Розробка програми 16

5.2 Етапи відладки 17

5.3 Типи помилок 18

ВИСНОВКИ 19

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ 20

Додаток А Схема алгоритму програми 21

Додаток Б Лістинг програми 22

Додаток В Роздрук результатів роботи програми 58

ВСТУП

Автоматизація є дуже важливою в бізнесі особливо в торговому. Також

автоматизація прожесів зменшить кількість помилок спричинених людським

фактором. Часто підприємці автоматизовують свій бізнес через збільшення

даних та неможливість керування ними. У чому саме полягає автоматизація, це

запровадження додатку в свій бізнес, до цього додатку прив’язується база

даних з якої будуть братись данні. Такий додаток буде застосовуватись в сфері торгівлі господарських товарів.

Область торгівлі господарських товарів включає в себе великий асортимент продукції. Саме тому автоматизація даної галузі є виправданою і доцільною. Тому що користувач може задовільнити більшість своїх потреб в одному магазині.

Метою курсової роботи є створення додатку який буде забезпечувати закупівлю товарів, необхідні маніпуляції з ними такі я переоцінка, аналіз потрібної інформацію про продаж товарів .

Завданням дослідження − розробити адекватний меті дослідження інструментарій: додаток робота якого буде спрямована на мінімізацію помилок

і на пришвидшення роботи магазину та підвищення прибутковості з

максимальним скороченням працівників що призведе до зменшення витрат на

заробітні плати та підвищення прибутковості магазину.

Методи дослідження даного об’єкта. Моніторинг та аналіз інформаційних джерел об’єкта. Аналіз дає змогу розділяти предмети дослідження на складові частини. Моніторинг - метод постійного спостереження за станом окремих об'єктів, перебігом певних процесів в окремих системах. Даний метод є дуже продуктивним, він допомагає оцінити стан у якому перебуває предметна область на даний час.

2 ПОСТАНОВКА ЗАДАЧІ

2.1 Характеристика предметної області

У курсовій роботі відповідно до завдання автоматизується діяльність 5 магазинів госоподарських товарів.

Кожна система управління базами даних, залежно від її призначення оперує тією чи іншою предметною областю. Виявлення предметної області є необхідним етапом розробки будь-якої програмного засобу. На цьому етапі виявляються інформаційні потреби всієї сукупності користувачів майбутньої системи.

Предметною областю даного ПЗ є комп'ютерний журнал. У даній предметній області необхідна своя база даних, тому що у журналі є багато різноманітної інформації.

База даних вирішує такі: облік товару, видача даних постачальників і поставлених ними товарах (фірма-постачальник, його реквізити, найменування товарів, характеристики, ціни), обчислює суми оплати.

Далі можна назвати засадничі поняття даної предметної області, саме:

1) замовлення

2) магазини

3) постачальники

4) постачання

5) співробітники

6) товари

Вихідні даних про магазини: кожен магазин має торговельний зал та свій власний склад. У фірми є постачальники, здійснюють поставку товарів до відповідних складів.

Також у кожному магазині та в офісі є свій штат співробітників: менеджери й консультанти,які здійснюють продаж товару, бухгалтер, кладовщики, грузчики-водії.

Клієнт, приходячи до магазину, взаємодіє з менеджером, який, своєю чергою, здійснює продаж товару. Операції з грошима і відбиття чека здійснює як і менеджер так і продавець. Купуючи товар, клієнт отримує чек із зазначенням товару та її вартості, виходячи з якого зі складу йому відвантажують необхідний товар.

2.2 Вимоги до програми

Програмне забезпечення призначене лише для адміністрації та кількох менеджерів офісу. ПЗ повинне забезпечувати цілісність даних для виключення непередбачених помилок через некоректне відображення інформації.

До головних функцій програми буде входити:

1. Можливість реєстрації товару, що містить:

* назву;
* оптову та роздрібна ціну;
* фото;
* одиниці виміру;
* країну виробника;

1. Реєстрація робітників, що містить:

* ПІБ робітника;
* дата народження;
* дата прийому на роботу;
* дата народження;
* посада та місце роботи;
* заробітну плату;

1. Закупка товару, що містить:

* конкретні товари;
* ціна товарів;
* кількість товарів;
* склад доставки;

1. Можливість роздрукувати накладну;
2. Можливість роздрукувати чеки;
3. Видалення товару, працівника або документу закупки;
4. Можливість робити переоцінку товарів за будь-якими ознаками;
5. Можливість редагування товару;

Програмне забезпечення розробляється для операційної системи Windows (7/8/10) на мові програмування Delphi. ПЗ повинно забезпечувати легке оновлення інформації, відображення якої передбачається цим технічним завданням.

2.3 Структура вхідних даних

Структура вхідних даних представлена у таблиці 2.1.

Таблиця 2.1 Структура вхідних даних

|  |  |  |  |
| --- | --- | --- | --- |
| Ідентифікатор | Тип | Діапазон значень | Пояснення |
| connectionString | String |  | Вибір БД для підключення |
| query | String |  | Обробка запитів |
| Tov\_Id | Integer | 0…999999 | ID товару |
| Tov\_Name | String |  | Назва товару |
| Per\_Id | Integer | 0…999999 | ID робітника |
| Per\_Name | String |  | Ім’я робітника |
| Per\_Surname | String |  | Прізвище робітника |
| Pr\_Id | String | 0…999999 | ID закупки |
| Pr\_Date | Date |  | Дата закупки |
| Pr\_IsConfirm | Boolean | True/False | Підтвердження закупки |
| Sl\_Id | Integer | 0…999999 | ID чеку |
| Sl\_Date | Date |  | Дата проведення чеку |
| St\_Id | Integer | 0…999999 | ID складу |
| St\_Name | String |  | Назва складу |
| Pl\_Id | String | 0…999999 | ID магазину |
| Pl\_Name | String |  | Назва магазину |

2.4 Структура вихідних даних

Структура вихідних даних представлена у таблиці 2.2.

Таблиця 2.2 Структура вихідних даних

|  |  |  |  |
| --- | --- | --- | --- |
| Ідентифікатор | Тип | Діапазон значень | Пояснення |
| Tov\_Id | Integer | 0…999999 | ID товару |
| Tov\_Name | String |  | Назва товару |
| Per\_Id | Integer | 0…999999 | ID робітника |
| Per\_Name | String |  | Ім’я робітника |
| Per\_Surname | String |  | Прізвище робітника |
| Pr\_Id | String | 0…999999 | ID закупки |
| Pr\_Date | Date |  | Дата закупки |
| Pr\_IsConfirm | Boolean | True/False | Підтвердження закупки |
| Sl\_Id | Integer | 0…999999 | ID чеку |
| Sl\_Date | Date |  | Дата проведення чеку |
| Tov\_Id | Integer | 0…999999 | ID товару |
| Tov\_Name | String |  | Назва товару |
| Per\_Id | Integer | 0…999999 | ID робітника |

Також до вихідних даних належать накладна та чек в форматі “.xls”.

3 КОМП’ЮТЕРНА СИСТЕМА

3.1 Технічні характеристики комп’ютера та зовнішніх пристроїв

Курсовий проект розроблявся на ПК з наступною технічною складовою частиною:

* Процесор AMD Ryzen 7 4800 H 2.9 GHz;
* Оперативна пам’ять 16 ГБ DDR4 3200 MHz;
* Відеокарта NVIDIA GTX 1650 Ti;
* Монітор 1920x1080 60Hz;
* Операційна система Windows 10 Pro;

Програма запускається та буде працювати на комп’ютері з наступними технічними характеристиками:

* Процесор з частотою 2.0 GHz;
* Об’єм оперативної пам’яті 1024МБ;
* Відеокарта об’ємом 512 МБ;
* Операційна система починаючи з Windows 7;

3.2 Вибір програмних засобів та операційної системи

Програмне забезпечення по своєму призначенню підрозділяють на системне, прикладне та інструментальне.

Системне програмне забезпечення - комплекс програм для управління ресурсами комп'ютера (центральним процесором, пам'яттю, введенням і виведенням даних), підтримання працездатності системи обробки інформації, підвищення ефективності її використання.

Цей клас програмного забезпечення носить загальний характер застосування і не залежить від предметної області. До системного програмного забезпечення пред'являються високі вимоги надійності роботи, зручності та ефективності використання. Як правило, комп'ютер купується з встановленим системним програмним забезпеченням. Системне програмне забезпечення включає в себе базову та сервісне програмне забезпечення.

Операційна система - великий набір програм для управління взаємодією всіх компонентів комп'ютера в процесі роботи і виконання поставлених завдань. Забезпечує роботу і взаємодію комп'ютерних пристроїв в цілому як системи, координує зв'язок з іншими пристроями в мережі, а користувачам і прикладним програмам надає ресурси і інтерфейс - засіб взаємодії з комп'ютером.

Середовищем розробки програмного забезпечення я обрав Embarcadero Delphi 10.4.2 Sydney, бо середовище містить великий функціонал та можливості для розробки, що дозволяє вдосконалити власне програмне забезпечення та спростити роботу з ним.

Delphi має власний та зручний інтерфейс з комфортним дизайном, який можна змінити власноруч. Також середовище має різні модифікації для мов програмування різного рівня.

Для розробки свого програмного забезпечення я обрав мову програмування Delphi (яка раніше називалася Object Pascal). Я вважаю, що ця мова має багато плюсів порівняно з іншими. Також ми вивчаємо Delphi в коледжі, що спрощує роботу у декілька разів.

Для розробки курсового проекту я обрав операційну систему Windows, тому що це найбільш розповсюджена і комфортна операційна система. Операційна система має досить великий функціонал і зручна у роботі. Більшість користувачів ПК обирають саме операційну систему Windows. Переваги операційної системи Windows:

* доступна по-справжньому багатовіконна робота. На екран можна вивести одночасно кілька додатків і гнучко перемикатися;
* при підключенні декількох моніторів додатки можна виводити на окремі дисплеї - це і є справжні Вікна, концепція роботи паралельно в декількох додатках;
* всі нові додатки повністю адаптовані до будь-яких типів пристроїв;
* після придбання програми, вона автоматично буде працювати на всіх комп'ютерах, що належать користувачеві: на робочому столі, планшеті і смартфоні під управлінням Windows;
* якщо виникають питання, найбільш ймовірно знайти відповіді в мережі, знову ж таки через її поширеність.

4 ОБ’ЄКТНО-ОРІЄНТОВАНЕ ПРОЕКТУВАННЯ

4.1 Інтерфейс програми

Цінність будь-якої програми визначається не тільки її здатністю безпомилково і якісно вирішувати поставлену задачу, а й тим, наскільки легко і зручно цією програмою можна користуватися. Навіть найпростіша, і в той же час, функціональна програма виявиться незатребуваною, якщо користувач (тобто кінцевий споживач програми) при роботі з нею швидко втомлюється, дратується, відчуває себе некомфортно. Під призначеним для користувача інтерфейсом (англ. User interface) будемо розуміти своєрідний комунікаційний канал (сполучна ланка), за допомогою якого користувач взаємодіє з комп'ютером.

З точки зору користувача саме інтерфейс є кінцевим продуктом. Інтерфейс являє собою сукупність використовуваних в програмі засобів для введення даних, способів відображення інформації на екрані дисплея, елементів для управління різними етапами рішення задачі. Основне призначення (головна мета) призначеного для користувача інтерфейсу - забезпечити короткий і зручний шлях до отримання результату.

Інтерфейс користувача комп’ютерного додатку включає:

* Ясність;
* Виразність;
* Відповідність;
* Послідовність;
* Продуктивність;

Невдалі інтерфейси сильно уповільнюють роботу користувачів, сприяють швидкої стомлюваності, великому числу помилок і, як наслідок, відбивають у користувачів подальше бажання працювати з програмою. У зв'язку з цим розробнику програми слід враховувати відомі рекомендації провідних фахівців в області створення призначеного для користувача інтерфейсу.

Саме тому було розроблено максимально комфортний і зручний для користування інтерфейс (див. рис. 4.1)

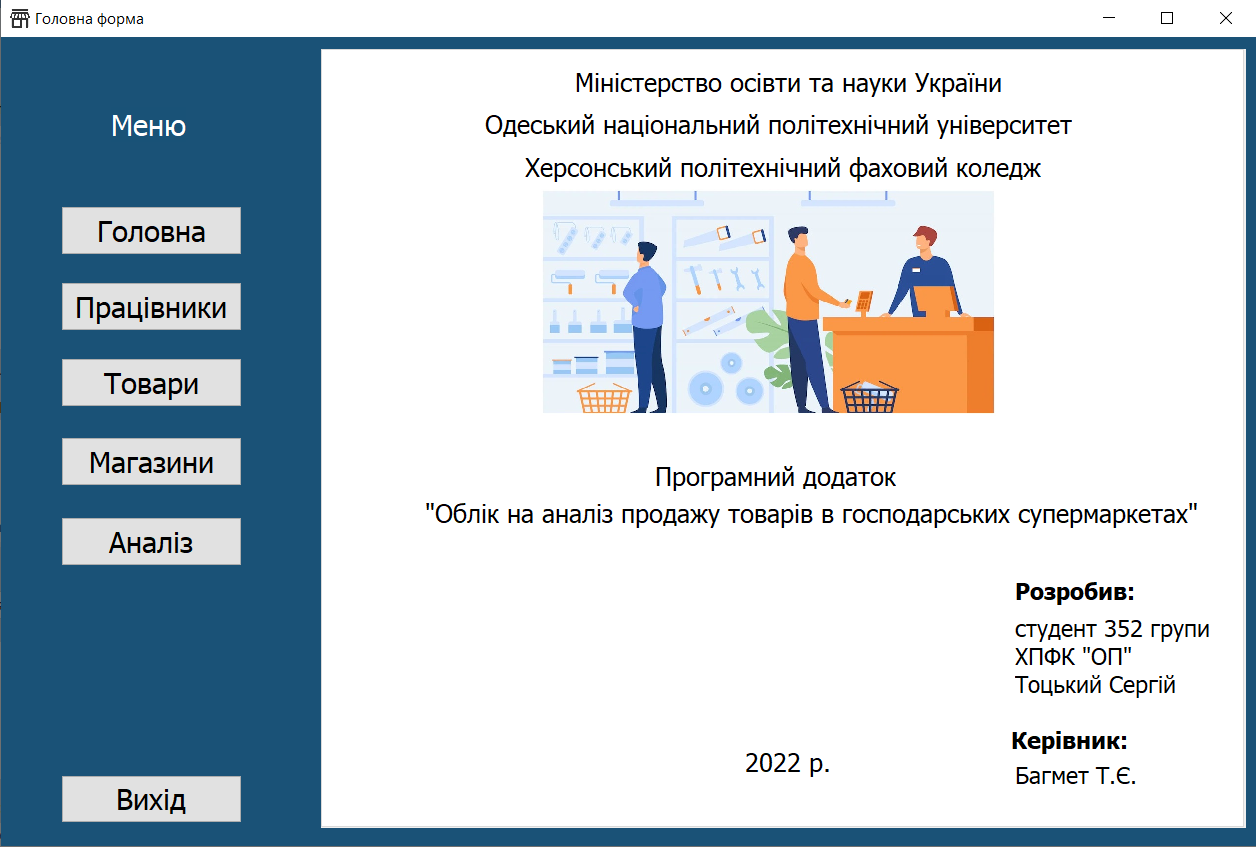


Рисунок 4.1 – Головне меню

4.2 Алгоритм рішення задачі

Програма в даному курсовому проекті складається з багатьох екранних форм, кожна за яких виконує певну функцію та має унікальну логіку роботи.

При запуску програми спочатку відкривається головна форма. Зліва на формі знаходиться кнопки навігації, що відкривають відповідну вкладку.

procedure TForm2.MainBtnClick(Sender: TObject); - відкриває початкову вкладку;

procedure TForm2.EmployeeBtnClick(Sender: TObject); - відкриває вкладку для роботи з працівниками;

procedure TForm2.TovarsBtnClick(Sender: TObject); - відкриває вкладку для роботи з товарами;

procedure TForm2.MagazBtnClick(Sender: TObject); - відкриває вкладку для перегляду чеків та наявності товарів на магазинах;

procedure TForm2.Button5Click(Sender: TObject); - відкриває вкладку для процедур аналізу.

При відкритті вкладки роботи з працівниками з’являється 3 кнопки. При натисканні кожної кнопки буде виконана відповідна процедура.

procedure TForm2.Button2Click(Sender: TObject); - відкриває форму для перегляду списку всіх працівників;

procedure TForm2.Button2Click(Sender: TObject); - відкриває форму для зміни місця роботи працівника;

procedure TForm2.Button2Click(Sender: TObject); - відкриває форму для додавання нового працівника;

При відкритті вкладки роботи з товарами з’являється 3 кнопки. При натисканні кожної кнопки буде виконана відповідна процедура.

procedure TForm2.tovarsClick(Sender: TObject); - відкриває форму для перегляду списку всіх товарів;

procedure TForm2.Button6Click(Sender: TObject);- відкриває форму для переоцінки товарів;

procedure TForm2.Button7Click(Sender: TObject); - відкриває форму для перегляду списку всіх документів закупки;

При відкритті вкладки роботи з мазинами з’являється перелік магазинів та 12 кнопок. Кожні 2 кнопки відповідають певному магазину. При натисканні кожної кнопки буде виконана відповідна процедура.

procedure TForm2.Button8Click(Sender: TObject); - перегляд наявності товарів на складі головному офісі.

procedure TForm2.Button9Click(Sender: TObject); - перегляд наявності товару на першому магазині.

procedure TForm2.Button15Click(Sender: TObject); - перегляд чеків на першому магазині.

procedure TForm2.Button10Click(Sender: TObject); - перегляд наявності товару на другому магазині.

procedure TForm2.Button16Click(Sender: TObject); - перегляд чеків на другому магазині.

procedure TForm2.Button11Click(Sender: TObject); - перегляд наявності товару на третьому магазині.

procedure TForm2.Button17Click(Sender: TObject); - перегляд чеків на третьому магазині.

procedure TForm2.Button11Click(Sender: TObject); - перегляд наявності товару на четвертому магазині.

procedure TForm2.Button17Click(Sender: TObject); - перегляд чеків на четвертому магазині.

procedure TForm2.Button11Click(Sender: TObject); - перегляд наявності товару на п’ятому магазині.

procedure TForm2.Button17Click(Sender: TObject); - перегляд чеків на п’ятому магазині.

procedure TForm2.Button12Click(Sender: TObject); - перегляд наявності товару на шостому магазині.

procedure TForm2.Button19Click(Sender: TObject); - перегляд чеків на шостому магазині.

5 ПРОГРАМУВАННЯ ТА ТЕСТУВАННЯ

5.1 Розробка програми

Розроблена програма призначена лише для операційної системи Windows, тому програма несумісна з іншими операційними системами (наприклад: Linux, Debian, MacOS тощо).

Після запуску програми, користувачу буде доступна головна форма з описом та головною інформацією, а також інформацію про студента, який розробив цю програму.

Для переходу на інші форми, користувач повинен обрати потрібний пункт меню.

На головній формі в меню доступні наступні пункти:

* Головна;
* Працівники;
* Товари;
* Магазини;
* Аналіз;

Обравши пункт “Працівники”, користувач переходить на вкладку для роботи з працівникам. На даній вкладці користувач має змогу переглянути список всіх працівників, змінити місце роботи або зареєструвати нового після чого він з’явиться в реєстрі працівників. Для переходу до іншої вкладки користувач може використати конопки навігації, що знаходять зліва.

Обравши пункт “Товари ”, користувач переходить на вкладку для роботи з товарами. Користувач має змогу переглянути каталог товарів, зробити переоцінку товарів або закупити нові товари. Зареєструвати товар можна із форми закупки або з каталогу товарів.

Обравши пункт “Магазини”, користувач переходить на вкладку для роботи з наявними магазинами. Користувач бачить чи наявний зв’язок з кожним магазином та може переглянути чеки, що були проведені на кожному магазині та перелік наявних товарів на кожному магазині.

Якщо користувач вирішить обрати пункт “Аналіз”, то він перейде на вкладку для роботи з аналітичними звітами.

Інформація про програму та розробника відображена на головній формі програми.

5.2 Етапи відладки

Відкладка - це процес виявлення причин виникнення помилки та її подальшого виправлення (на відміну від тестування, що є процесом виявлення самого факту існування помилки).

Процес відкладки включає в себе:

* Вироблення системних вимог;
* Вироблення вимог до програмного забезпечення;
* Аналіз та проектування;
* Реалізацію - процес написання коду;
* Продуктизацію – процес вияснення відношення програмного забезпечення вимогам і доведення до них;
* Експлуатацію – процес використання програми.

В ході розробки програмного забезпечення виникали деякі помилки різного типу. Приклад синтаксичної помилки в курсовому проекті (див. рис. 5.1)

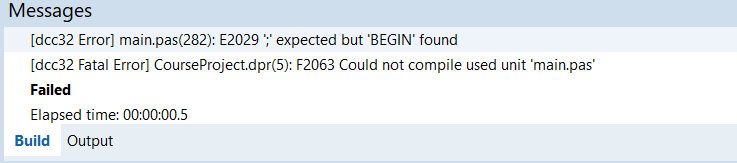


Рисунок 5.1 – Синтаксична помилка

Також зустрічалися ще й помилки виконання. Помилки виконання це означає, що трапилося непередбачене. Приклад такої помилки зображено на рисунку 5.2.

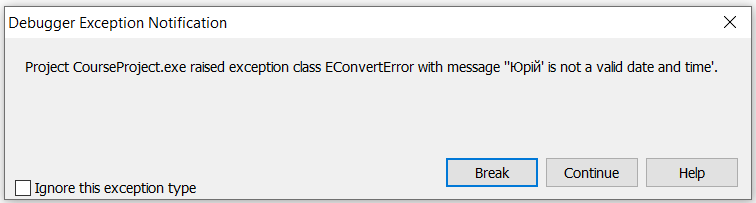


Рисунок 5.2 – Помилка виконання

5.3 Типи помилок

З огляду на різноманітність джерел помилок при складанні плану тестування класифікують помилки на два типи: 1 - синтаксичні; 2 - семантичні (смислові).

* Синтаксичні помилки - будь-який транслятор зможе виконати програму тільки в тому випадку, якщо програма синтаксично правильна.
* Семантичні помилки - зміст написаної програми відрізняється від того, що ви мали на увазі.
* Помилкивиконання – цей тип помилок зазвичай виникає під час виконання програми. Якщо в програмі виникає виключення, то це означає, що у ході виконання сталося щось непередбачене: наприклад, програмі було передано некоректне значення, або програма спробувала розділити якесь значення на нуль, що є неприпустимим.

У курсовому проекті зустрічалися помилки виконання, та синтаксичні помилки, які були успішно виправлені.

ВИСНОВКИ

Для розробки поставленої задачі було обрано мову програмування Object Pascal. Object Pascal - це мова високого рівня, яка підтримує структурне і об'єктно-орієнтоване проектування. Перевага мови складається з легкого читання коду, швидкої компіляції, і використання декількох модульних файлів для модульного програмування. Хоча Object Pascal не складна мова, але вона дає основні поняття програмування, через це вона добре підходить початківцям. Саме тому вона дуже добре підійшла для реалізації цього курсового проекту

Програмне забезпечення може знадобитися компаніям, які хочуть спростити та автоматизувати свою роботу з багатьма магазинами, а саме реєстрація працівників, товарів, облік продаж тощо. Програма прискорює продуктивність компанії за допомогою швидкого доступу до даних.

Також у додатку було створено простий інтерфейс, оскільки складний або сильно кольоровий інтерфейс відволікав би від роботи та концентрація користувача розсіювалась.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Багмет Т.Є. Методичні вказівки щодо виконання курсового проекту з дисципліни «Об’єктно орієнтоване програмування»
2. Архангельский А.Я. «Программирование в Delphi 7»
3. Введение в Delphi

http://msk.edu.ua/ivk/Informatika/Uch\_posobiya/Programmirovanie/Delphi\_textbook/delphi-book\_1.htm l

1. Осіпов Д.Л «Базы данных и Delphi»
2. Культін Н.Б. «Основи програмування в Delphi 2007»
3. Вадим Шищук «Основи програмування на алгоритмічній мові Pascal»

Додаток А

Схема алгоритму програми

Початок

Обирає вихід

Головна форма

Кінець

Перехід на вкдадку для роботи з даними

Перехід на потрібний розділ

Вводить дані

Підтверджує дані

Успішно

Обирає меню

так

ні

Додаток Б

Лістинг програми

unit main;

interface

uses

Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants, System.Classes, Vcl.Graphics,

Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.StdCtrls, Vcl.ComCtrls,

Vcl.Imaging.pngimage, Vcl.ExtCtrls, PersonList, ChangePlace, PersonEdit, TovarList, ChangePrice,

Vcl.Imaging.jpeg, InnerPurchaseList, System.Net.URLClient,

System.Net.HttpClient, System.Net.HttpClientComponent, IdBaseComponent,

IdComponent, IdTCPConnection, IdTCPClient, IdTelnet, IdRawBase, IdRawClient,

IdIcmpClient, PresenceList, Saleslist;

type

TForm2 = class(TForm)

PageControl1: TPageControl;

TabSheet1: TTabSheet;

TabSheet2: TTabSheet;

TabSheet3: TTabSheet;

TabSheet4: TTabSheet;

MainBtn: TButton;

EmployeeBtn: TButton;

TovarsBtn: TButton;

MagazBtn: TButton;

Label1: TLabel;

Label2: TLabel;

Label3: TLabel;

Label4: TLabel;

Image1: TImage;

Button1: TButton;

Button2: TButton;

Button3: TButton;

Button4: TButton;

tovars: TButton;

Button5: TButton;

Label5: TLabel;

Button6: TButton;

TabSheet5: TTabSheet;

Button7: TButton;

Label6: TLabel;

Image2: TImage;

Label7: TLabel;

Image3: TImage;

Image4: TImage;

Label8: TLabel;

Image5: TImage;

Label9: TLabel;

Image6: TImage;

Label10: TLabel;

Label11: TLabel;

Image7: TImage;

Timer1: TTimer;

IdIcmpClient1: TIdIcmpClient;

Button8: TButton;

Button9: TButton;

Button10: TButton;

Button11: TButton;

Button12: TButton;

Button13: TButton;

Button14: TButton;

Button15: TButton;

Button16: TButton;

Button17: TButton;

Button18: TButton;

Button19: TButton;

Label12: TLabel;

Label13: TLabel;

Label14: TLabel;

Label15: TLabel;

Label16: TLabel;

Label17: TLabel;

Label18: TLabel;

Label19: TLabel;

Label20: TLabel;

Image8: TImage;

Image9: TImage;

Image10: TImage;

Button20: TButton;

Button21: TButton;

Button22: TButton;

Image11: TImage;

Label21: TLabel;

procedure MainBtnClick(Sender: TObject);

procedure FormCreate(Sender: TObject);

procedure EmployeeBtnClick(Sender: TObject);

procedure TovarsBtnClick(Sender: TObject);

procedure MagazBtnClick(Sender: TObject);

procedure Button1Click(Sender: TObject);

procedure Button2Click(Sender: TObject);

procedure Button3Click(Sender: TObject);

procedure Button4Click(Sender: TObject);

procedure tovarsClick(Sender: TObject);

procedure Button6Click(Sender: TObject);

procedure Button7Click(Sender: TObject);

procedure Timer1Timer(Sender: TObject);

procedure Button8Click(Sender: TObject);

procedure Button9Click(Sender: TObject);

procedure Button10Click(Sender: TObject);

procedure Button11Click(Sender: TObject);

procedure Button12Click(Sender: TObject);

procedure Button13Click(Sender: TObject);

procedure Button5Click(Sender: TObject);

procedure Button15Click(Sender: TObject);

procedure Button16Click(Sender: TObject);

procedure Button17Click(Sender: TObject);

procedure Button18Click(Sender: TObject);

procedure Button19Click(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form2: TForm2;

implementation

{$R \*.dfm}

procedure TForm2.MagazBtnClick(Sender: TObject);

begin

PageControl1.ActivePageIndex:= 3;

end;

procedure TForm2.MainBtnClick(Sender: TObject);

begin

PageControl1.ActivePageIndex:= 0;

end;

procedure TForm2.Timer1Timer(Sender: TObject);

begin

IdIcmpClient1.host := '172.217.16.36';

IdIcmpClient1.ping;

if IdIcmpClient1.ReplyStatus.ReplyStatusType = rsEcho then

Image2.Picture.LoadFromFile('C:\Users\bdvys\Desktop\CourseProject\assets\green\_dot.png')

else

Image2.Picture.LoadFromFile('C:\Users\bdvys\Desktop\CourseProject\assets\grey\_dot.png');

IdIcmpClient1.host := '172.217.16.36';

IdIcmpClient1.ping;

if IdIcmpClient1.ReplyStatus.ReplyStatusType = rsEcho then

Image3.Picture.LoadFromFile('C:\Users\bdvys\Desktop\CourseProject\assets\green\_dot.png')

else

Image3.Picture.LoadFromFile('C:\Users\bdvys\Desktop\CourseProject\assets\grey\_dot.png');

IdIcmpClient1.host := '172.217.16.36';

IdIcmpClient1.ping;

if IdIcmpClient1.ReplyStatus.ReplyStatusType = rsEcho then

Image4.Picture.LoadFromFile('C:\Users\bdvys\Desktop\CourseProject\assets\green\_dot.png')

else

Image4.Picture.LoadFromFile('C:\Users\bdvys\Desktop\CourseProject\assets\grey\_dot.png');

IdIcmpClient1.host := '172.217.16.36';

IdIcmpClient1.ping;

if IdIcmpClient1.ReplyStatus.ReplyStatusType = rsEcho then

Image5.Picture.LoadFromFile('C:\Users\bdvys\Desktop\CourseProject\assets\green\_dot.png')

else

Image5.Picture.LoadFromFile('C:\Users\bdvys\Desktop\CourseProject\assets\grey\_dot.png');

IdIcmpClient1.host := '172.217.16.36';

IdIcmpClient1.ping;

if IdIcmpClient1.ReplyStatus.ReplyStatusType = rsEcho then

Image6.Picture.LoadFromFile('C:\Users\bdvys\Desktop\CourseProject\assets\green\_dot.png')

else

Image6.Picture.LoadFromFile('C:\Users\bdvys\Desktop\CourseProject\assets\grey\_dot.png');

IdIcmpClient1.host := '172.217.16.36';

IdIcmpClient1.ping;

if IdIcmpClient1.ReplyStatus.ReplyStatusType = rsEcho then

Image7.Picture.LoadFromFile('C:\Users\bdvys\Desktop\CourseProject\assets\green\_dot.png')

else

Image7.Picture.LoadFromFile('C:\Users\bdvys\Desktop\CourseProject\assets\grey\_dot.png');

end;

procedure TForm2.TovarsBtnClick(Sender: TObject);

begin

PageControl1.ActivePageIndex:= 2;

end;

procedure TForm2.tovarsClick(Sender: TObject);

begin

Form6 := TForm6.Create(Application.MainForm);

Form6.Show;

end;

procedure TForm2.Button10Click(Sender: TObject);

begin

PresenceListForm := TPresenceListForm.Create(Self);

PresenceListForm.ViewPresenceList(3);

end;

procedure TForm2.Button11Click(Sender: TObject);

begin

PresenceListForm := TPresenceListForm.Create(Self);

PresenceListForm.ViewPresenceList(4);

end;

procedure TForm2.Button12Click(Sender: TObject);

begin

PresenceListForm := TPresenceListForm.Create(Self);

PresenceListForm.ViewPresenceList(5);

end;

procedure TForm2.Button13Click(Sender: TObject);

begin

PresenceListForm := TPresenceListForm.Create(Self);

PresenceListForm.ViewPresenceList(6);

end;

procedure TForm2.Button15Click(Sender: TObject);

begin

SalesListForm := TSalesListForm.Create(nil);

SalesListForm.SalesListView(2);

end;

procedure TForm2.Button16Click(Sender: TObject);

begin

SalesListForm := TSalesListForm.Create(nil);

SalesListForm.SalesListView(3);

end;

procedure TForm2.Button17Click(Sender: TObject);

begin

SalesListForm := TSalesListForm.Create(nil);

SalesListForm.SalesListView(4);

end;

procedure TForm2.Button18Click(Sender: TObject);

begin

SalesListForm := TSalesListForm.Create(nil);

SalesListForm.SalesListView(5);

end;

procedure TForm2.Button19Click(Sender: TObject);

begin

SalesListForm := TSalesListForm.Create(nil);

SalesListForm.SalesListView(6);

end;

procedure TForm2.Button1Click(Sender: TObject);

begin

Form2.Close;

end;

procedure TForm2.Button2Click(Sender: TObject);

begin

Form1 := TForm1.Create(Self);

Form1.Show;

end;

procedure TForm2.Button3Click(Sender: TObject);

begin

Form5 := TForm5.Create(Application.MainForm);

Form5.Show;

end;

procedure TForm2.Button4Click(Sender: TObject);

begin

Form3 := Tform3.Create(Application.MainForm);

Form3.AddPerson;

end;

procedure TForm2.Button5Click(Sender: TObject);

begin

PageControl1.ActivePageIndex:= 4;

end;

procedure TForm2.Button6Click(Sender: TObject);

begin

Form8 := TForm8.Create(Application.MainForm);

Form8.Show;

end;

procedure TForm2.Button7Click(Sender: TObject);

begin

InnerPurchaseListForm := TInnerPurchaseListForm.Create(Self);

InnerPurchaseListForm.Show;

end

procedure TForm2.Button8Click(Sender: TObject);

begin

PresenceListForm := TPresenceListForm.Create(Self);

PresenceListForm.ViewPresenceList(1);

end;

procedure TForm2.Button9Click(Sender: TObject);

begin

PresenceListForm := TPresenceListForm.Create(Self);

PresenceListForm.ViewPresenceList(2);

end;

procedure TForm2.EmployeeBtnClick(Sender: TObject);

begin

PageControl1.ActivePageIndex:= 1;

end;

procedure TForm2.FormCreate(Sender: TObject);

begin

TabSheet1.TabVisible := False;

TabSheet2.TabVisible := False;

TabSheet3.TabVisible := False;

TabSheet4.TabVisible := False;

TabSheet5.TabVisible := False;

PageControl1.ActivePageIndex := 0;

Application.Icon.LoadFromFile('C:\Users\bdvys\Desktop\CourseProject\shop48.ico');

end;

end.

unit ChangePlace;

interface

uses

Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants, System.Classes, Vcl.Graphics,

Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.StdCtrls, Vcl.DBCtrls, Vcl.Mask,

Data.DB, Data.Win.ADODB, Vcl.Grids, Vcl.DBGrids, Vcl.ExtCtrls;

type

TForm5 = class(TForm)

DataSource1: TDataSource;

ADOQuery1: TADOQuery;

Panel1: TPanel;

DBGrid1: TDBGrid;

Edit1: TEdit;

Edit2: TEdit;

Label1: TLabel;

btn: TButton;

Edit4: TEdit;

ComboBox1: TComboBox;

procedure DBGrid1CellClick(Column: TColumn);

procedure FormCreate(Sender: TObject);

procedure btnClick(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form5: TForm5;

id: integer;

implementation

{$R \*.dfm}

procedure TForm5.btnClick(Sender: TObject);

var sql: string;

begin

sql := ADOQuery1.SQL.Text;

ADOQuery1.SQL.Clear;

//ADOQuery1.SQL.Add(sql);

ADOQuery1.SQL.Add('update Person set Pe\_PlaceOfWork = ');

ADOQuery1.SQL.Add(IntToStr(ComboBox1.ItemIndex + 3));

ADOQuery1.SQL.Add(', Pe\_Posistion = ''');

ADOQuery1.SQL.Add(Edit4.Text);

ADOQuery1.SQL.Add(''' where Pe\_Id = ');

ADOQuery1.SQL.Add(IntToStr(id));

ADOQuery1.ExecSQL;

ADOQuery1.Close;

ADOQuery1.SQL.Clear;

ADOQuery1.SQL.Add(sql);

ADOQuery1.Open;

end;

procedure TForm5.DBGrid1CellClick(Column: TColumn);

var name : string;

begin

name := VarToStr(DBGrid1.Fields[1].Value) + ' ' + VarToStr(DBGrid1.Fields[2].Value);

id := DBGrid1.Fields[0].Value;

Combobox1.ItemIndex := 0;

Edit1.Text := name;

Edit2.Text := DBGrid1.Fields[4].Value;

ComboBox1.ItemIndex := DBGrid1.Fields[3].Value - 3;

Edit4.Text := ADOQuery1.FieldByName('Pe\_Posistion').Value;

end;

procedure TForm5.FormCreate(Sender: TObject);

begin

ADOQuery1.Open;

end;

end.

unit EditWage;

interface

uses

Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants, System.Classes, Vcl.Graphics,

Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Data.DB, Data.Win.ADODB, Vcl.StdCtrls,

Vcl.Mask, Vcl.DBCtrls;

type

TForm4 = class(TForm)

DataSource1: TDataSource;

ADOQuery1: TADOQuery;

DBEdit1: TDBEdit;

Label1: TLabel;

Label2: TLabel;

Button1: TButton;

Edit1: TEdit;

procedure Button1Click(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

procedure EditWageView(id: integer);

end;

var

Form4: TForm4;

ID\_ : integer;

implementation

{$R \*.dfm}

procedure TForm4.Button1Click(Sender: TObject);

begin

ADOQuery1.Sql.Clear;

ADOQuery1.Sql.Add('Update Person Set Pe\_Wage = ' + Edit1.Text + 'Where Pe\_Id = ' + IntToStr(ID\_));

ADOQuery1.ExecSQL;

Form4.Close;

end;

procedure TForm4.EditWageView(ID : integer);

begin

ADOQuery1.Parameters.ParamByName('ID').Value := ID;

ADOQuery1.Open;

ID\_ := ID;

Edit1.Text := '';

Form4.Show;

end;

end.

unit PersonEdit;

interface

uses

Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants, System.Classes, Vcl.Graphics,

Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.StdCtrls, Data.Win.ADODB, Data.DB,

Vcl.Mask, Vcl.DBCtrls, Vcl.ExtCtrls, EditWage, Vcl.Menus;

type

TForm3 = class(TForm)

Label1: TLabel;

Label2: TLabel;

Label3: TLabel;

Label4: TLabel;

ADOQuery1: TADOQuery;

DataSource1: TDataSource;

Label5: TLabel;

Label6: TLabel;

Label7: TLabel;

Label8: TLabel;

Label9: TLabel;

DBEdit1: TDBEdit;

DBEdit2: TDBEdit;

DBEdit3: TDBEdit;

DBEdit4: TDBEdit;

DBEdit5: TDBEdit;

DBEdit6: TDBEdit;

DBEdit7: TDBEdit;

DBEdit8: TDBEdit;

Label10: TLabel;

DBEdit9: TDBEdit;

Label11: TLabel;

DBEdit10: TDBEdit;

DBImage1: TDBImage;

Button1: TButton;

MainMenu1: TMainMenu;

N1: TMenuItem;

Button2: TButton;

procedure Button1Click(Sender: TObject);

procedure N1Click(Sender: TObject);

procedure Button2Click(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

procedure ViewPersonEdit(id: integer);

procedure AddPerson;

end;

var

Form3: TForm3;

implementation

{$R \*.dfm}

procedure TForm3.Button1Click(Sender: TObject);

begin

Form4 := TForm4.Create(Self);

Form4.EditWageView(StrToInt(VarToStr(ADOQuery1.Parameters.ParamByName('Id').Value)));

end;

procedure TForm3.Button2Click(Sender: TObject);

begin

ADOQuery1.Post;

end;

procedure TForm3.N1Click(Sender: TObject);

begin

ADOQuery1.Refresh;

end;

procedure TForm3.ViewPersonEdit(id: integer);

begin

try

ADOQuery1.Parameters.ParamByName('Id').Value := id;

ADOQuery1.Open;

Form3.Caption := Form3.Caption + IntToStr(ADOQuery1.FieldByName('Pe\_Id').Value);

Form3.Show;

except

end;

end;

procedure TForm3.AddPerson;

begin

try

Form3.Caption := 'Реєстрація нового працівника';

Button2.Visible := True;

Button1.Visible := False;

ADOQuery1.Active := True;

Form3.Show;

except

end;

end;

end.

unit PersonEdit;

interface

uses

Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants, System.Classes, Vcl.Graphics,

Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.StdCtrls, Data.Win.ADODB, Data.DB,

Vcl.Mask, Vcl.DBCtrls, Vcl.ExtCtrls, EditWage, Vcl.Menus;

type

TForm3 = class(TForm)

Label1: TLabel;

Label2: TLabel;

Label3: TLabel;

Label4: TLabel;

ADOQuery1: TADOQuery;

DataSource1: TDataSource;

Label5: TLabel;

Label6: TLabel;

Label7: TLabel;

Label8: TLabel;

Label9: TLabel;

DBEdit1: TDBEdit;

DBEdit2: TDBEdit;

DBEdit3: TDBEdit;

DBEdit4: TDBEdit;

DBEdit5: TDBEdit;

DBEdit6: TDBEdit;

DBEdit7: TDBEdit;

DBEdit8: TDBEdit;

Label10: TLabel;

DBEdit9: TDBEdit;

Label11: TLabel;

DBEdit10: TDBEdit;

DBImage1: TDBImage;

Button1: TButton;

MainMenu1: TMainMenu;

N1: TMenuItem;

Button2: TButton;

procedure Button1Click(Sender: TObject);

procedure N1Click(Sender: TObject);

procedure Button2Click(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

procedure ViewPersonEdit(id: integer);

procedure AddPerson;

end;

var

Form3: TForm3;

implementation

{$R \*.dfm}

procedure TForm3.Button1Click(Sender: TObject);

begin

Form4 := TForm4.Create(Self);

Form4.EditWageView(StrToInt(VarToStr(ADOQuery1.Parameters.ParamByName('Id').Value)));

end;

procedure TForm3.Button2Click(Sender: TObject);

begin

ADOQuery1.Post;

end;

procedure TForm3.N1Click(Sender: TObject);

begin

ADOQuery1.Refresh;

end;

procedure TForm3.ViewPersonEdit(id: integer);

begin

try

ADOQuery1.Parameters.ParamByName('Id').Value := id;

ADOQuery1.Open;

Form3.Caption := Form3.Caption + IntToStr(ADOQuery1.FieldByName('Pe\_Id').Value);

Form3.Show;

except

end;

end;

procedure TForm3.AddPerson;

begin

try

Form3.Caption := 'Реєстрація нового працівника';

Button2.Visible := True;

Button1.Visible := False;

ADOQuery1.Active := True;

Form3.Show;

except

end;

end;

end.

unit PresenceList;

interface

uses

Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants, System.Classes, Vcl.Graphics,

Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Data.DB, Data.Win.ADODB, Vcl.Grids,

Vcl.DBGrids, TovarEdit;

type

TPresenceListForm = class(TForm)

DBGrid1: TDBGrid;

ADOQuery1: TADOQuery;

DataSource1: TDataSource;

procedure DBGrid1CellClick(Column: TColumn);

private

{ Private declarations }

public

{ Public declarations }

procedure ViewPresenceList(storage\_id: integer);

end;

var

PresenceListForm: TPresenceListForm;

implementation

{$R \*.dfm}

procedure TPresenceListForm.DBGrid1CellClick(Column: TColumn);

begin

Form7 := TForm7.Create(Self);

Form7.ViewTovarEdit(DBGrid1.Fields[0].Value);

end;

procedure TPresenceListForm.ViewPresenceList(storage\_id: integer);

var Flags: TReplaceFlags;

begin

ADOQuery1.Parameters.ParamByName('St\_Id').Value := storage\_id;

ADOQuery1.Open;

TNumericField(ADOQuery1.FieldByName('Tov\_Price')).DisplayFormat := '0 грн.';

Flags := [rfReplaceAll];

PresenceListForm.Caption := 'Наявність на ' + StringReplace(VarToStr(ADOQuery1.FieldByName('St\_Name').Value), 'Склад', 'складі', Flags);

PresenceListForm.Show;

end;

end.

unit Saleslist;

interface

uses

Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants, System.Classes, Vcl.Graphics,

Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Data.DB, Vcl.Grids, Vcl.DBGrids,

Data.Win.ADODB, A7Rep;

type

TSalesListForm = class(TForm)

DataSource1: TDataSource;

ADOQuery1: TADOQuery;

DBGrid1: TDBGrid;

ADOQuery2: TADOQuery;

DataSource2: TDataSource;

procedure DBGrid1CellClick(Column: TColumn);

private

{ Private declarations }

public

{ Public declarations }

procedure SalesListView(id: integer);

end;

var

SalesListForm: TSalesListForm;

implementation

{$R \*.dfm}

procedure TSalesListForm.DBGrid1CellClick(Column: TColumn);

var

i : Integer;

s, sum : Double;

Rep: TA7Rep;

begin

if not ADOQuery2.IsEmpty then

ADOQuery2.Close;

ADOQuery2.Parameters.ParamByName('ID').Value := DBGrid1.Fields[0].Value;

ADOQuery2.Open;

Rep := TA7Rep.Create(nil);

Rep.OpenTemplate('C:\Users\bdvys\Desktop\CourseProject\Excel\Товарний чек.xls');

Rep.PasteBand('Title');

Rep.SetValueF('#Date#', ADOQuery2.FieldByName('SL\_Date').AsString);

Rep.SetValue('#ID#', ADOQuery2.FieldByName('SL\_Id').AsString);

ADOQuery2.First;

for i := 1 to ADOQuery2.recordcount do

begin

Rep.PasteBand('Line');

Rep.SetValue('#N#', i);

Rep.SetValue('#Title#', ADOQuery2.FieldByName('Tov\_Name').AsString);

Rep.SetValue('#Unit#', 'шт.');

Rep.SetValue('#Amount#', ADOQuery2.FieldByName('STL\_Amount').AsString);

s := ADOQuery2.FieldByName('Tov\_Price').AsFloat \* ADOQuery2.FieldByName('STL\_Amount').AsFloat;

Rep.SetValue('#Price#', ADOQuery2.FieldByName('Tov\_Price').AsFloat);

sum := sum + s;

ADOQuery2.Next;

end;

Rep.PasteBand('Foot');

Rep.SetValue('#Summa#', sum);

Rep.SetValue('#PL\_Name#', ADOQuery2.FieldByName('PL\_Name').AsString);

Rep.Show;

end;

procedure TSalesListForm.SalesListView(id: integer);

begin

ADOQuery1.Parameters.ParamByName('ID').Value := id;

ADOQuery1.Open;

SalesListForm.show;

end;

end.

unit ChangePrice;

interface

uses

Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants, System.Classes, Vcl.Graphics,

Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.StdCtrls, Vcl.DBCtrls, Vcl.ExtCtrls,

Data.DB, Vcl.Grids, Vcl.DBGrids, Data.Win.ADODB, TovarEdit;

type

TForm8 = class(TForm)

Panel1: TPanel;

RadioGroup1: TRadioGroup;

Button3: TButton;

Label1: TLabel;

DBGrid1: TDBGrid;

DataSource1: TDataSource;

ADOQuery1: TADOQuery;

DataSource2: TDataSource;

ADOQuery2: TADOQuery;

ComboBox1: TComboBox;

Label2: TLabel;

Edit1: TEdit;

ADOQuery3: TADOQuery;

DataSource3: TDataSource;

procedure DBGrid1CellClick(Column: TColumn);

procedure RadioGroup1Click(Sender: TObject);

procedure FormCreate(Sender: TObject);

procedure ComboBox1Change(Sender: TObject);

procedure Button3Click(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form8: TForm8;

implementation

{$R \*.dfm}

procedure TForm8.Button3Click(Sender: TObject);

var Filter : string;

i : integer;

begin

if RadioGroup1.ItemIndex = 0 then

begin

Filter := 'where Dep\_Id = ' + IntToStr(ComboBox1.ItemIndex)

end;

if RadioGroup1.ItemIndex = 1 then

begin

Filter := 'where Tov\_Id in (';

for i := 0 to DBGrid1.SelectedRows.Count - 1 do

begin

DBGrid1.DataSource.DataSet.Bookmark := DBGrid1.SelectedRows[i];

Filter := Filter + DBGrid1.Fields[0].AsString;

if i < DBGrid1.SelectedRows.Count - 1 then

Filter := Filter + ', '

else

Filter := Filter + ')'

end;

end;

ADOQuery3.SQL.Clear;

ADOQuery3.SQL.Add('update tovar set Tov\_Price = Tov\_Price + Tov\_Price\*(:Percent / 100) from tovar inner join Depart on Dep\_Id = Tov\_Depart');

ADOQuery3.Sql.Add(Filter);

ADOQuery3.Parameters.ParamByName('Percent').Value := strtoint(Edit1.Text);

ADOQuery3.ExecSQL();

ShowMessage('Оновленно - ' + inttostr(DBGrid1.SelectedRows.Count) + ' товарів');

ADOQuery1.Refresh;

end;

procedure TForm8.ComboBox1Change(Sender: TObject);

var sql : string;

begin

ADOQuery1.Close;

ADOQuery1.Sql.Clear;

ADOQuery1.Sql.Add('select \* from Tovar inner join depart on Dep\_Id = Tov\_Depart inner join Measuring on Ms\_Id = Tov\_Measuring');

if ComboBox1.ItemIndex > 0 then

ADOQuery1.Sql.Add(' where Dep\_Id = ' + IntToStr(ComboBox1.ItemIndex));

ADOQuery1.Open;

TNumericField(ADOQuery1.FieldByName('Tov\_Price')).DisplayFormat := '0 грн.';

Label1.Caption := 'Обрано - ' + IntToStr(ADOQuery1.RecordCount);

end;

procedure TForm8.DBGrid1CellClick(Column: TColumn);

begin

if Column.Index = 0 then

begin

Form7 := TForm7.Create(Self);

Form7.ViewTOvarEdit(DBGrid1.Fields[0].Value);

end;

Label1.Caption := 'Обрано - ' + IntToStr(DBGrid1.SelectedRows.Count);

end;

procedure TForm8.FormCreate(Sender: TObject);

begin

ADOQuery1.Open;

TNumericField(ADOQuery1.FieldByName('Tov\_Price')).DisplayFormat := '0 грн.';

ADOQuery2.Open;

ADOQuery2.First;

while not ADOQuery2.Eof do

begin

ComboBox1.Items.Add(ADOQuery2.FieldByName('Dep\_Name').AsString);

ADOQuery2.Next;

end;

end;

procedure TForm8.RadioGroup1Click(Sender: TObject);

begin

if RadioGroup1.ItemIndex = 1 then

begin

ComboBox1.Enabled := False;

DBGrid1.Options := [dgTitles,dgIndicator,dgColumnResize,dgColLines,dgRowLines,dgTabs,dgConfirmDelete,dgCancelOnExit,dgMultiSelect,dgTitleClick,dgTitleHotTrack];

end;

if RadioGroup1.ItemIndex = 0 then

begin

ComboBox1.Enabled := True;

DBGrid1.Options := [dgTitles,dgIndicator,dgColumnResize,dgColLines,dgRowLines,dgTabs,dgConfirmDelete,dgCancelOnExit,dgTitleClick,dgTitleHotTrack];

end;

end;

end.

unit InnerPurchaseList;

interface

uses

Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants, System.Classes, Vcl.Graphics,

Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Data.DB, Vcl.Grids, Vcl.DBGrids,

Vcl.Menus, InnerPurchaseEdit, Data.Win.ADODB;

type

TInnerPurchaseListForm = class(TForm)

MainMenu1: TMainMenu;

makenew: TMenuItem;

DBGrid1: TDBGrid;

DataSource1: TDataSource;

ADOQuery1: TADOQuery;

N1: TMenuItem;

procedure makenewClick(Sender: TObject);

procedure FormCreate(Sender: TObject);

procedure N1Click(Sender: TObject);

procedure DBGrid1CellClick(Column: TColumn);

procedure DBGrid1DrawColumnCell(Sender: TObject; const Rect: TRect;

DataCol: Integer; Column: TColumn; State: TGridDrawState);

private

{ Private declarations }

public

{ Public declarations }

end;

var

InnerPurchaseListForm: TInnerPurchaseListForm;

implementation

{$R \*.dfm}

procedure TInnerPurchaseListForm.DBGrid1CellClick(Column: TColumn);

begin

InnerPurchaseEditForm := TInnerPurchaseEditForm.Create(Self);

InnerPurchaseEditForm.InnerPurchaseEditFormView(DBGrid1.Fields[0].Value);

end;

procedure TInnerPurchaseListForm.DBGrid1DrawColumnCell(Sender: TObject;

const Rect: TRect; DataCol: Integer; Column: TColumn; State: TGridDrawState);

begin

if Column.Field.DataSet.FieldByName('Pr\_IsConifrm').AsString = 'так' then

begin

DBGrid1.canvas.brush.color := cl3DLight; //не имеет фокуса

DBGrid1.DefaultDrawColumnCell(Rect, DataCol, Column, State);

end;

end;

procedure TInnerPurchaseListForm.FormCreate(Sender: TObject);

begin

ADOQuery1.Open;

end;

procedure TInnerPurchaseListForm.makenewClick(Sender: TObject);

begin

InnerPurchaseEditForm := TInnerPurchaseEditForm.Create(Self);

InnerPurchaseEditForm.InnerPurchaseEditFormView(0);

end;

procedure TInnerPurchaseListForm.N1Click(Sender: TObject);

begin

ADOQuery1.Close;

ADOQuery1.Open;

end;

end.

unit TovarEdit;

interface

uses

Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants, System.Classes, Vcl.Graphics,

Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Data.DB, Data.Win.ADODB, Vcl.DBCtrls,

Vcl.StdCtrls, Vcl.Mask, Vcl.ExtCtrls, IdBaseComponent, IdComponent,

IdTCPConnection, IdTCPClient, IdHTTP, System.Net.URLClient,

System.Net.HttpClient, System.Net.HttpClientComponent, JPEG;

type

TForm7 = class(TForm)

ADOQuery1: TADOQuery;

DataSource1: TDataSource;

Label1: TLabel;

Label2: TLabel;

Label3: TLabel;

Label4: TLabel;

Label5: TLabel;

Label6: TLabel;

DBEdit1: TDBEdit;

DBEdit2: TDBEdit;

DBEdit3: TDBEdit;

DBEdit4: TDBEdit;

DBEdit5: TDBEdit;

DBEdit6: TDBEdit;

Image1: TImage;

NetHTTPClient1: TNetHTTPClient;

Label7: TLabel;

DBEdit7: TDBEdit;

ADOQuery2: TADOQuery;

DataSource2: TDataSource;

Button1: TButton;

DBEdit8: TDBEdit;

procedure Button1Click(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

procedure ViewTovarEdit(id: integer);

end;

var

Form7: TForm7;

implementation

{$R \*.dfm}

procedure TForm7.Button1Click(Sender: TObject);

var new\_id: integer;

begin

ADOQuery1.Post;

ADOQuery2.Open;

new\_id := ADOQuery2.FieldByName('Tov\_Id').Value;

ADOQuery2.Close;

ADOQuery2.SQL.Clear;

ADOQuery2.SQL.Add('insert into Presence values(' + IntToStr(new\_id) + ', 0, 1)');

ADOQuery2.ExecSQL;

ViewTovarEdit(new\_id);

end;

procedure TForm7.ViewTovarEdit(id: integer);

var Stream: TMemoryStream;

Jpeg: TJPEGImage;

begin

if id > 0 then

begin

try

ADOQuery1.Parameters.ParamByName('Id').Value := id;

ADOQuery1.Open;

TNumericField(ADOQuery1.FieldByName('Tov\_Price')).DisplayFormat := '0 грн.';

TNumericField(ADOQuery1.FieldByName('Tov\_PriceOpt')).DisplayFormat := '0 грн.';

Button1.Visible := false;

Label7.Visible := false;

DBEdit7.Visible := false;

Form7.Caption := 'Карточка товару' + ' - ' + IntToStr(ADOQuery1.FieldByName('Tov\_Id').Value);

Stream := TMemoryStream.Create();

Jpeg := TJPEGImage.Create();

Try

NetHTTPClient1.Get(ADOQuery1.FieldByName('Tov\_Photo').Value, Stream);

Stream.Position := 0;

if pos('.jpg',ADOQuery1.FieldByName('Tov\_Photo').Value) <> 0 then

begin

jpeg.LoadFromStream(Stream);

Image1.Picture.Assign(jpeg);

end

else

begin

Image1.Picture.LoadFromStream(Stream)

end;

Finally

FreeAndNil(Stream);

Form7.ClientWidth := 510 + Image1.Width;

Form7.ClientHeight := 18 + Image1.Height;

if Form7.ClientHeight < 270 then

Form7.ClientHeight := 270;

End;

except

end;

end

else

begin

Form7.Caption := 'Додати новий товар';

Label7.Visible := true;

DBEdit7.Visible := true;

Button1.Visible := true;

ADOQuery1.Active := true;

DBEdit4.DataField := 'Tov\_Depart';

DBEdit5.DataField := 'Tov\_Measuring';

Image1.Picture.LoadFromFile('C:\Users\bdvys\Desktop\CourseProject\assets\help\_pic.png');

Form7.ClientWidth := 510 + Image1.Width;

Form7.ClientHeight := 18 + Image1.Height;

if Form7.ClientHeight < 270 then

Form7.ClientHeight := 270;

end;

Form7.Show;

end;

end.

unit TovarList;

interface

uses

Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants, System.Classes, Vcl.Graphics,

Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Data.DB, Vcl.Grids, Vcl.DBGrids,

Vcl.Menus, Data.Win.ADODB, TovarEdit;

type

TForm6 = class(TForm)

MainMenu1: TMainMenu;

N1: TMenuItem;

N2: TMenuItem;

DBGrid1: TDBGrid;

N3: TMenuItem;

N4: TMenuItem;

N5: TMenuItem;

N6: TMenuItem;

N7: TMenuItem;

N8: TMenuItem;

N9: TMenuItem;

N10: TMenuItem;

N11: TMenuItem;

ADOQuery1: TADOQuery;

DataSource1: TDataSource;

N12: TMenuItem;

procedure N1Click(Sender: TObject);

procedure N3Click(Sender: TObject);

procedure N4Click(Sender: TObject);

procedure N5Click(Sender: TObject);

procedure N6Click(Sender: TObject);

procedure N7Click(Sender: TObject);

procedure N8Click(Sender: TObject);

procedure N9Click(Sender: TObject);

procedure N10Click(Sender: TObject);

procedure N11Click(Sender: TObject);

procedure FormCreate(Sender: TObject);

procedure DBGrid1CellClick(Column: TColumn);

procedure N12Click(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

procedure QRefresh(Depart: integer);

end;

var

Form6: TForm6;

implementation

{$R \*.dfm}

procedure TForm6.DBGrid1CellClick(Column: TColumn);

begin

Form7 := TForm7.Create(Self);

Form7.ViewTovarEdit(DBGrid1.Fields[0].Value);

end;

procedure TForm6.FormCreate(Sender: TObject);

begin

ADOQuery1.Open;

TNumericField(ADOQuery1.FieldByName('Tov\_Price')).DisplayFormat := '0 грн.';

end;

procedure TForm6.N10Click(Sender: TObject);

begin

QRefresh(8);

end;

procedure TForm6.N11Click(Sender: TObject);

begin

QRefresh(9);

end;

procedure TForm6.N12Click(Sender: TObject);

begin

Form7 := TForm7.Create(Self);

Form7.ViewTovarEdit(0);

end;

procedure TForm6.N1Click(Sender: TObject);

begin

if not ADOQuery1.IsEmpty then

ADOQuery1.Close;

ADOQuery1.Open;

end;

procedure TForm6.N3Click(Sender: TObject);

begin

QRefresh(1);

end;

procedure TForm6.N4Click(Sender: TObject);

begin

QRefresh(2);

end;

procedure TForm6.N5Click(Sender: TObject);

begin

QRefresh(3);

end;

procedure TForm6.N6Click(Sender: TObject);

begin

QRefresh(4);

end;

procedure TForm6.N7Click(Sender: TObject);

begin

QRefresh(5);

end;

procedure TForm6.N8Click(Sender: TObject);

begin

QRefresh(6);

end

procedure TForm6.N9Click(Sender: TObject);

begin

QRefresh(7);

end;

procedure TForm6.QRefresh(Depart: integer);

begin

ADOQuery1.Close;

ADOQuery1.Sql.Clear;

ADOQuery1.SQL.Text := 'select \* from Tovar inner join depart on Dep\_Id = Tov\_Depart inner join Measuring on Ms\_Id = Tov\_Measuring';

ADOQuery1.Sql.Add('');

if Depart = 0 then

ADOQuery1.Sql.Add('')

else

ADOQuery1.Sql.Add('where Dep\_Id = ' + IntToStr(Depart));

ADOQuery1.Open;

end;

end.

unit InnerPurchaseEdit;

interface

uses

Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants, System.Classes, Vcl.Graphics,

Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Data.DB, Data.Win.ADODB, Vcl.StdCtrls,

Vcl.DBCtrls, Vcl.Grids, Vcl.DBGrids, Vcl.ExtCtrls, TovarEdit, Vcl.Menus, ComObj,

A7Rep;

type

TInnerPurchaseEditForm = class(TForm)

Panel1: TPanel;

DBGrid1: TDBGrid;

DBGrid2: TDBGrid;

Panel2: TPanel;

Panel3: TPanel;

Button1: TButton;

Button2: TButton;

DataSource1: TDataSource;

ADOQuery1: TADOQuery;

ComboBox1: TComboBox;

ADOQuery2: TADOQuery;

DataSource2: TDataSource;

DataSource3: TDataSource;

ADOQuery3: TADOQuery;

ComboBox2: TComboBox;

PopupMenu1: TPopupMenu;

N1: TMenuItem;

Edit1: TEdit;

Button5: TButton;

MainMenu1: TMainMenu;

N2: TMenuItem;

N3: TMenuItem;

Label1: TLabel;

Label2: TLabel;

Label3: TLabel;

Label4: TLabel;

Label5: TLabel;

procedure FormCreate(Sender: TObject);

procedure ComboBox1Change(Sender: TObject);

procedure DBGrid2CellClick(Column: TColumn);

procedure Button2Click(Sender: TObject);

procedure Button1Click(Sender: TObject);

procedure ComboBox2Change(Sender: TObject);

procedure DBGrid1CellClick(Column: TColumn);

procedure N1Click(Sender: TObject);

procedure N2Click(Sender: TObject);

procedure N3Click(Sender: TObject);

procedure Button5Click(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

procedure InnerPurchaseEditFormView(id\_ : integer);

end;

var

InnerPurchaseEditForm: TInnerPurchaseEditForm;

id: integer;

implementation

{$R \*.dfm}

procedure TInnerPurchaseEditForm.InnerPurchaseEditFormView(id\_ : integer);

begin

id := id\_;

n2.Caption := 'Підтвердити';

if id = 0 then

begin

ADOQuery1.Close;

ADOQuery1.SQL.Clear;

ADOQuery1.SQL.Add('insert into Purchase(Pr\_Storage) values (1)');

ADOQuery1.ExecSQL;

ADOQuery1.Close;

ADOQuery1.SQL.Clear;

ADOQuery1.SQL.Add('select top 1 \* from Purchase order by Pr\_Id desc');

ADOQuery1.Open;

InnerPurchaseEditFormView(ADOQuery1.FieldByName('Pr\_Id').Value);

end

else

begin

InnerPurchaseEditForm.Caption := 'Документ закупки - ' + IntToStr(id);

ADOQuery3.Parameters.ParamByName('ID').Value := id;

ADOQuery3.Open;

ADOQuery1.Close;

ADOQuery1.SQL.Clear;

ADOQuery1.SQL.Add('select \* from Purchase where Pr\_Id = ' + IntToStr(id));

ADOQuery1.Open;

ComboBox2.ItemIndex := ADOQuery1.FieldByName('Pr\_Storage').Value - 1;

if ADOQuery3.FieldByName('Pr\_Seller').AsString <> '' then

Button5.Caption := 'Редагувати';

Edit1.Text := ADOQuery3.FieldByName('Pr\_Seller').AsString;

TNumericField(ADOQuery3.FieldByName('Summa')).DisplayFormat := '0 грн.';

TNumericField(ADOQuery3.FieldByName('Tov\_PriceOpt')).DisplayFormat := '0 грн.';

TNumericField(ADOQuery2.FieldByName('Tov\_PriceOpt')).DisplayFormat := '0 грн.';

if ADOQuery1.FieldByName('Pr\_IsConifrm').Value = True then

begin

InnerPurchaseEditForm.Caption := 'Документ закупки - ' + IntToStr(id) + ' ЗАТВЕРДЖЕНО ✅';

N2.Caption := 'Відмінити';

N2.Enabled := False;

Button2.Enabled := False;

Button1.Enabled := False;

Button5.Enabled := False;

Edit1.Enabled := False;

ComboBox2.Enabled := False;

ComboBox1.Enabled := False;

DBGrid1.ReadOnly := True;

end;

end;

InnerPurchaseEditForm.Show;

end;

procedure TInnerPurchaseEditForm.N1Click(Sender: TObject);

begin

ADOQuery1.Close;

ADOQuery1.SQL.Clear;

ADOQuery1.SQL.Add('delete from PurchaseTovList where PTL\_Doc = ' + IntToStr(id) + 'and PTL\_Tovar = ' + VarToStr(DBGrid1.Fields[0].Value));

ADOQuery1.ExecSQL;

ADOQuery3.Close;

ADOQuery3.Parameters.ParamByName('ID').Value := id;

ADOQuery3.Open;

end;

procedure TInnerPurchaseEditForm.N2Click(Sender: TObject);

begin

ADOQuery1.Close;

ADOQuery1.Sql.Clear;

ADOQuery1.Sql.Add('update Purchase set Pr\_IsConifrm = iif(Pr\_IsConifrm = 1, 0, 1) where Pr\_Id = ' + IntToStr(id));

ADOQuery1.ExecSQL;

InnerPurchaseEditFormView(id);

end;

procedure TInnerPurchaseEditForm.N3Click(Sender: TObject);

var

i : Integer;

s, sum : Double;

Rep: TA7Rep;

begin

Rep := TA7Rep.Create(nil);

Rep.OpenTemplate('C:\Users\bdvys\Desktop\CourseProject\Excel\Товарна накладна.xls');

Rep.PasteBand('Title');

Rep.SetValue('#VENDOR#',ADOQuery3.FieldByName('Pr\_Seller').AsString);

Rep.SetValue('#BUY#','ООО Товари для дому');

Rep.SetValueF('#D#', ADOQuery3.FieldByName('Pr\_DateConfirmStr').AsString);

Rep.SetValue('#NOTE#','-');

//Rep.SetComment('#ID#','Здесь мы комментируем если нужно'); // Обязательно делаем комментарий ПЕРЕД тем как вставим значение в ячейку, иначе значение затрет метку и SetComment не найдет куда писать коммент

Rep.SetValue('#ID#', ADOQuery3.FieldByName('Pr\_Id').AsString);

ADOQuery3.First;

for i := 1 to ADOQuery3.recordcount do

begin

Rep.PasteBand('Line');

Rep.SetValue('#N#', i);

Rep.SetValue('#NAME#', ADOQuery3.FieldByName('Tov\_Name').AsString);

Rep.SetValue('#UNIT#', 'шт.');

Rep.SetValue('#QUANT#', ADOQuery3.FieldByName('PTL\_Amount').AsString);

s := ADOQuery3.FieldByName('Tov\_PriceOpt').AsFloat \* ADOQuery3.FieldByName('PTL\_Amount').AsFloat;

Rep.SetValue('#PRICE#', ADOQuery3.FieldByName('Tov\_PriceOpt').AsFloat);

sum := sum + s;

ADOQuery3.Next;

end;

Rep.PasteBand('Foot');

Rep.SetValue('#SUMMA#', sum);

Rep.SetValue('#CURRENT\_DATE#', DateToStr(now));

Rep.Show;

end;

procedure TInnerPurchaseEditForm.Button1Click(Sender: TObject);

var sql: string;

i: integer;

begin

sql := 'insert into PurchaseTovList(PTL\_Doc, PTL\_Tovar) select ' + IntToStr(id) + ', Tov\_Id from tovar where Tov\_Id in (';

for i := 0 to (DBGrid2.SelectedRows.Count - 1) do

begin

DBGrid2.DataSource.DataSet.Bookmark := DBGrid2.SelectedRows[i];

sql := sql + DBGrid2.Fields[0].AsString;

if i < DBGrid2.SelectedRows.Count - 1 then

sql := sql + ', '

else

sql := sql + ')'

end;

ADOQuery1.Close;

ADOQuery1.SQL.Clear;

ADOQuery1.SQL.Add(sql);

ADOQuery1.ExecSQL;

ADOQuery3.Close;

ADOQuery3.Parameters.ParamByName('ID').Value := id;

ADOQuery3.Open;

end;

procedure TInnerPurchaseEditForm.Button2Click(Sender: TObject);

begin

Form7 := TForm7.Create(Self);

Form7.ViewTovarEdit(0);

end;

procedure TInnerPurchaseEditForm.Button5Click(Sender: TObject);

begin

ADOQuery1.Close;

ADOQuery1.SQL.Clear;

ADOQuery1.SQL.Add('update Purchase set Pr\_Seller = ''' + edit1.Text + ''' where Pr\_Id = ' + IntToStr(id));

ADOQuery1.ExecSQL;

end;

procedure TInnerPurchaseEditForm.ComboBox1Change(Sender: TObject);

begin

ADOQuery2.Close;

ADOQuery2.Sql.Clear;

ADOQuery2.Sql.Add('select \* from Tovar inner join Depart on Dep\_Id = Tov\_Depart');

if ComboBox1.ItemIndex > 0 then

ADOQuery2.Sql.Add(' where Dep\_Id = ' + IntToStr(ComboBox1.ItemIndex));

ADOQuery2.Open;

TNumericField(ADOQuery2.FieldByName('Tov\_Price')).DisplayFormat := '0 грн.';

end;

procedure TInnerPurchaseEditForm.ComboBox2Change(Sender: TObject);

begin

ADOQuery1.Close;

ADOQuery1.Sql.Clear;

ADOQuery1.Sql.Add('update Purchase set Pr\_Storage = ' + IntToStr(ComboBox2.ItemIndex + 1) + 'where Pr\_Id = ' + IntToStr(id));

ADOQuery1.ExecSQL;

end;

procedure TInnerPurchaseEditForm.DBGrid1CellClick(Column: TColumn);

begin

if Column.Index = 0 then

begin

Form7 := TForm7.Create(Self);

Form7.ViewTOvarEdit(DBGrid1.Fields[0].Value);

end;

end;

procedure TInnerPurchaseEditForm.DBGrid2CellClick(Column: TColumn);

begin

if Column.Index = 0 then

begin

Form7 := TForm7.Create(Self);

Form7.ViewTOvarEdit(DBGrid2.Fields[0].Value);

end;

end;

procedure TInnerPurchaseEditForm.FormCreate(Sender: TObject);

begin

ADOQuery1.Open;

ADOQuery2.Open;

TNumericField(ADOQuery2.FieldByName('Tov\_Price')).DisplayFormat := '0 грн.';

ADOQuery1.First;

while not ADOQuery1.Eof do

begin

ComboBox1.Items.Add(ADOQuery1.FieldByName('Dep\_Name').AsString);

ADOQuery1.Next;

end;

ADOQuery1.Close;

ADOQuery1.SQL.Clear;

ADOQuery1.SQL.Add('select \* from Storage');

ADOQuery1.Open;

ADOQuery1.First;

while not ADOQuery1.Eof do

begin

ComboBox2.Items.Add(ADOQuery1.FieldByName('St\_Name').AsString);

ADOQuery1.Next;

end;

end;

end.

Додаток В

Роздрук результатів роботи програми

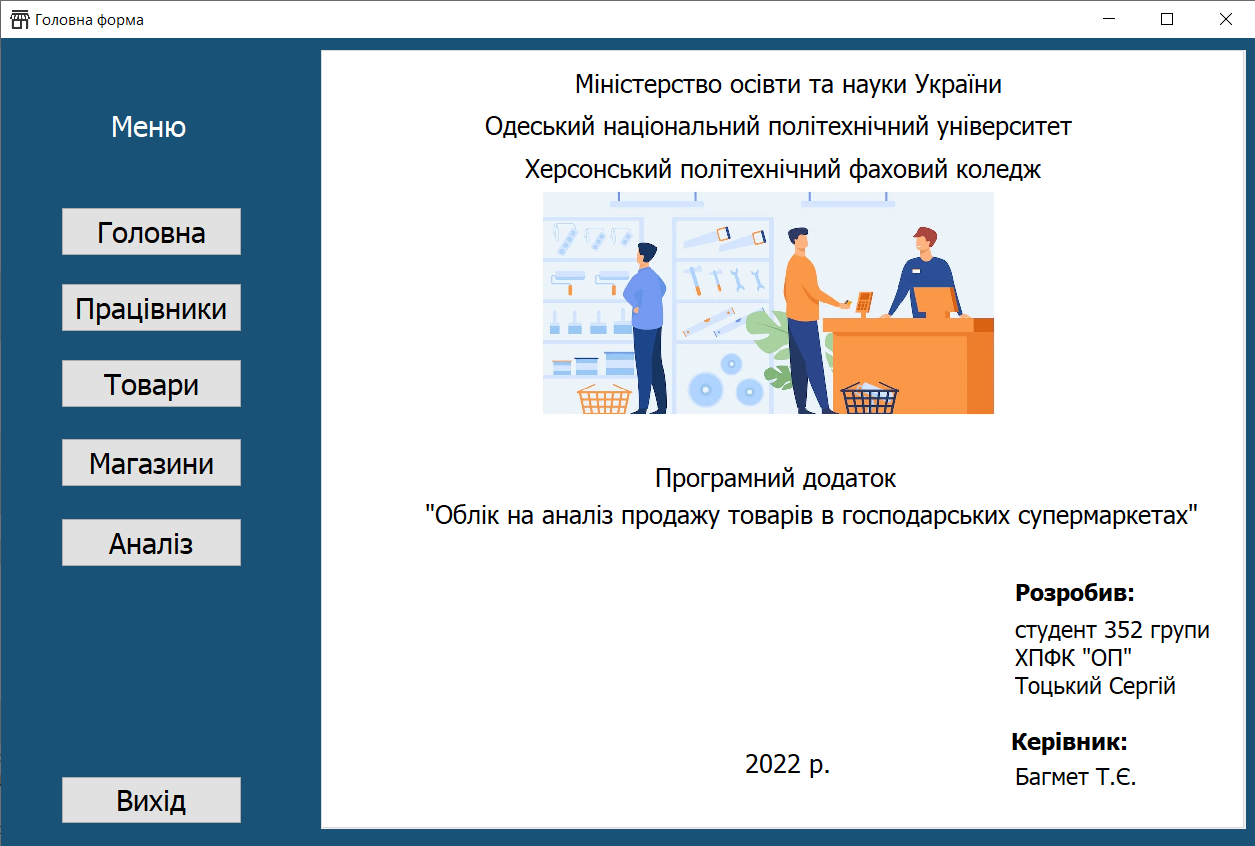


Рисунок А.1 – Головна форма

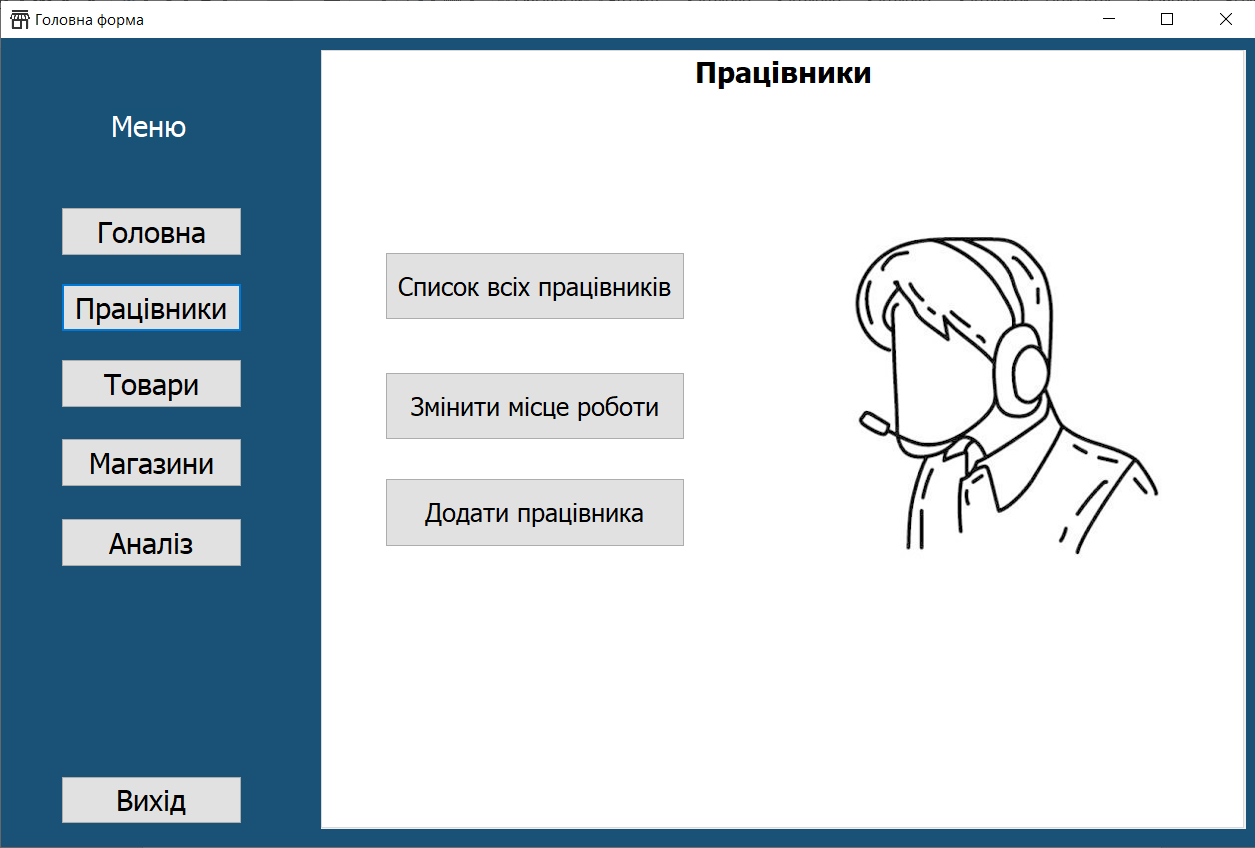


Рисунок А.1 – Вкладка працівники

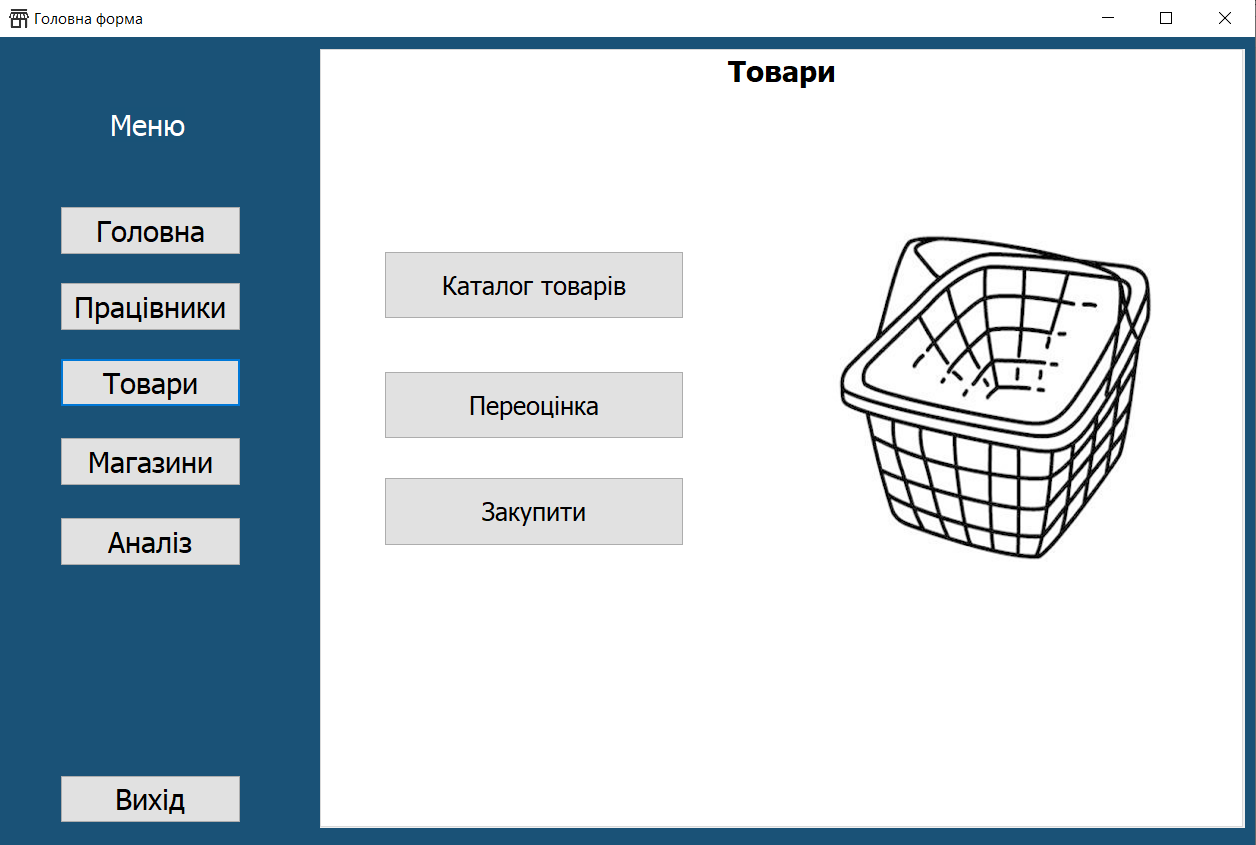


Рисунок А.1 – Вкладка товари



Рисунок А.1 – Вкладка магазини

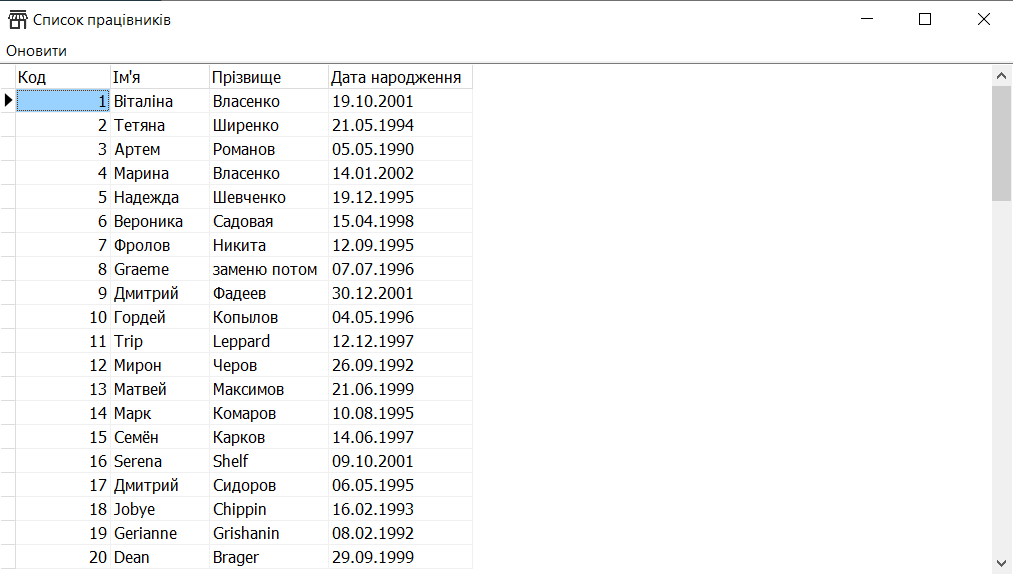


Рисунок А.1 – Форма зі списком всіх працівників

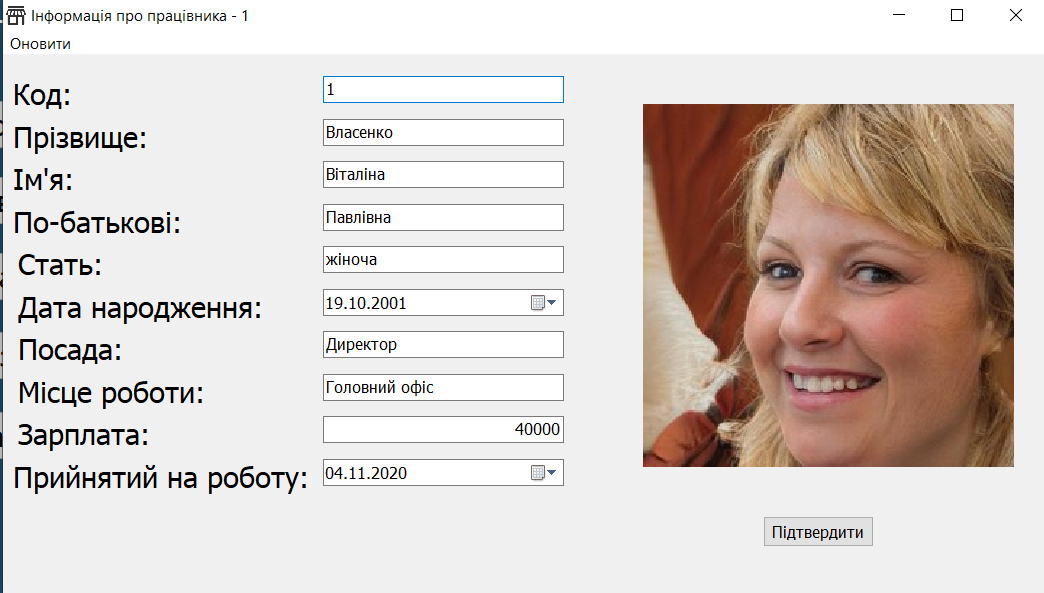


Рисунок А.1 – Форма інформації про працівника

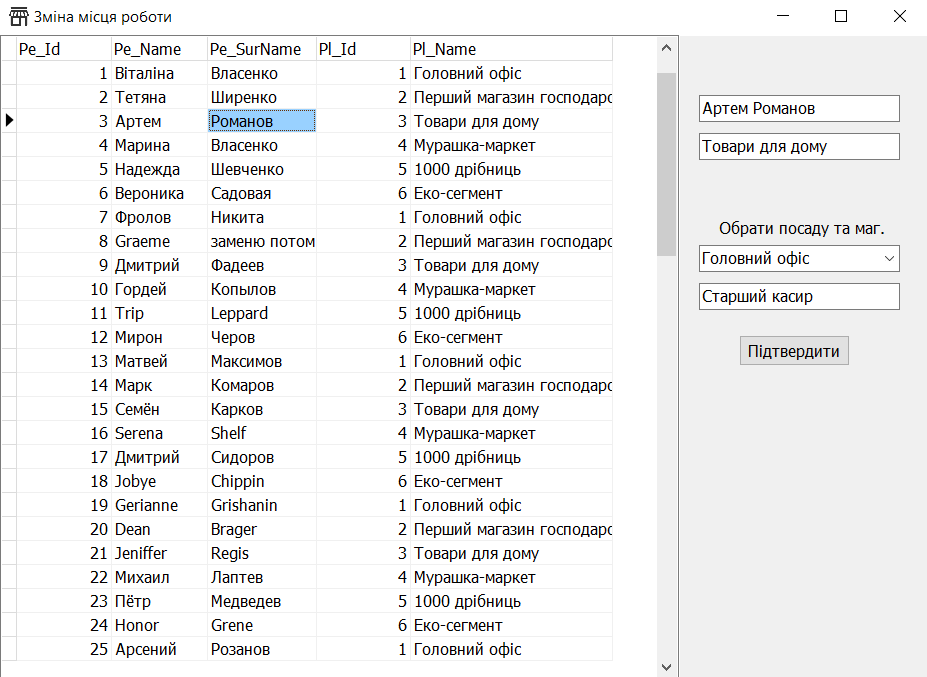


Рисунок А.1 – Форма для зміни місця роботи працівника

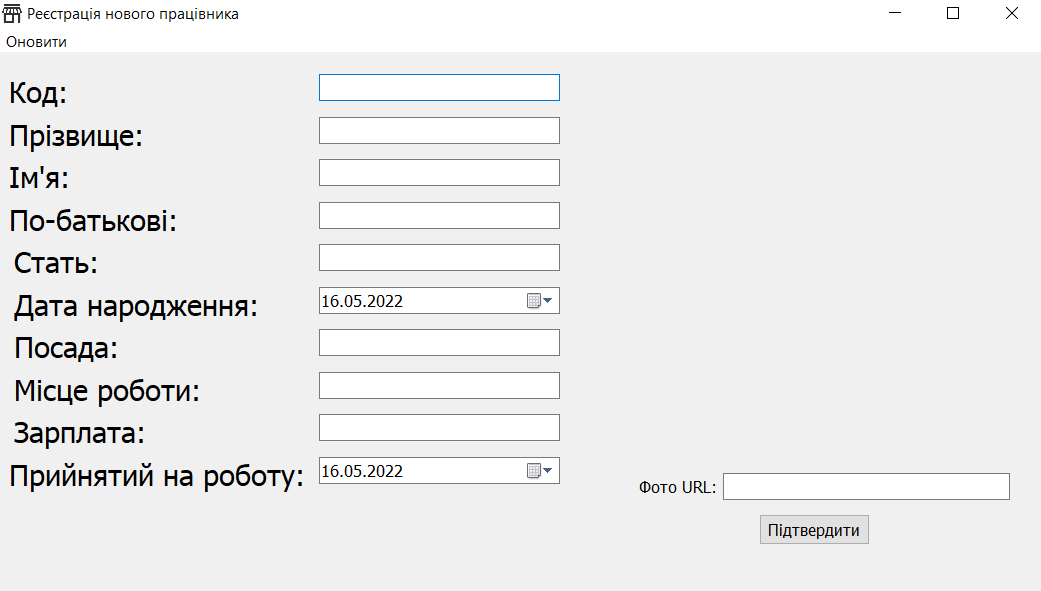


Рисунок А.1 – Форма реєстрації нового співробітника

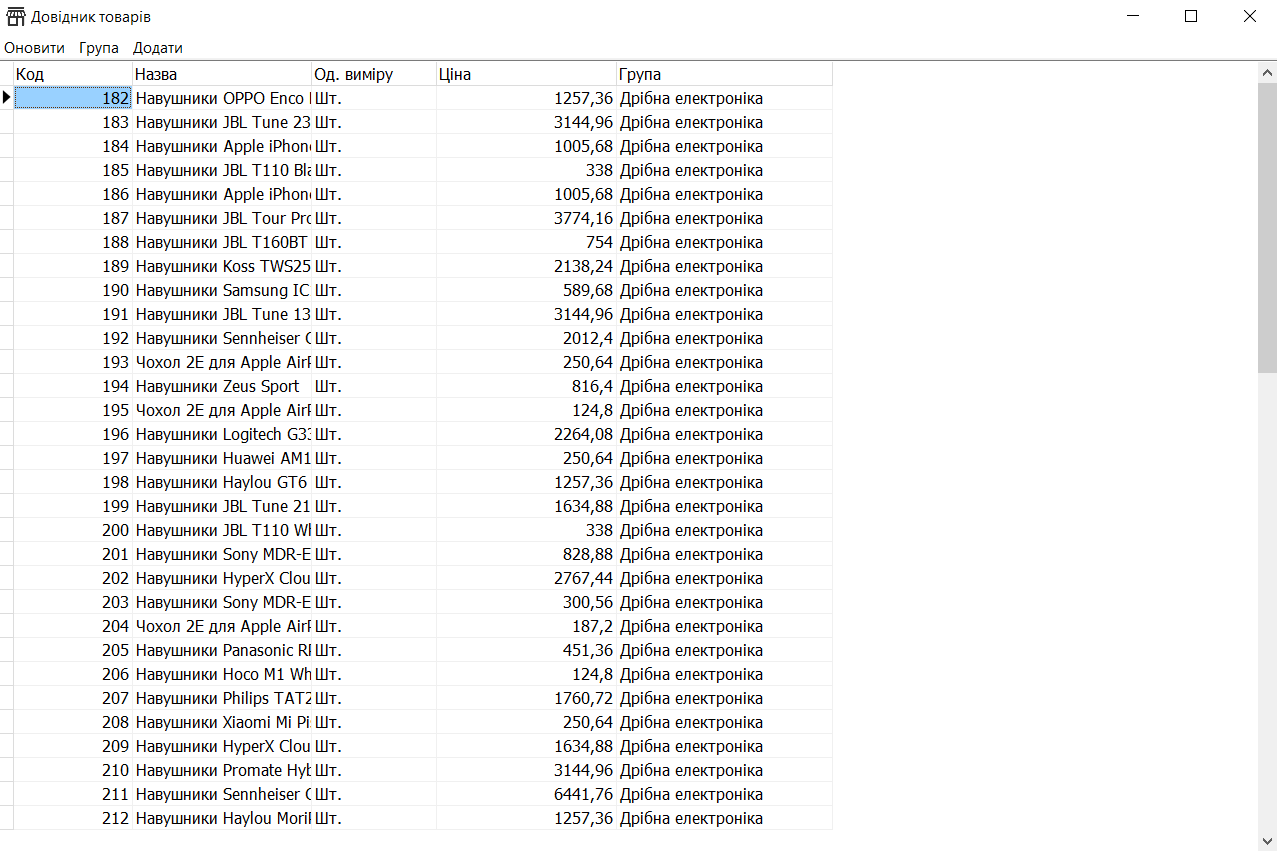


Рисунок А.1 – Форма списку всіх товарів

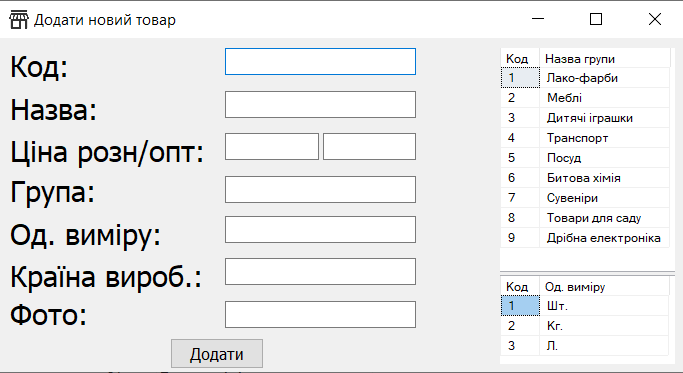


Рисунок А.1 – Форма реєстрації нового товару

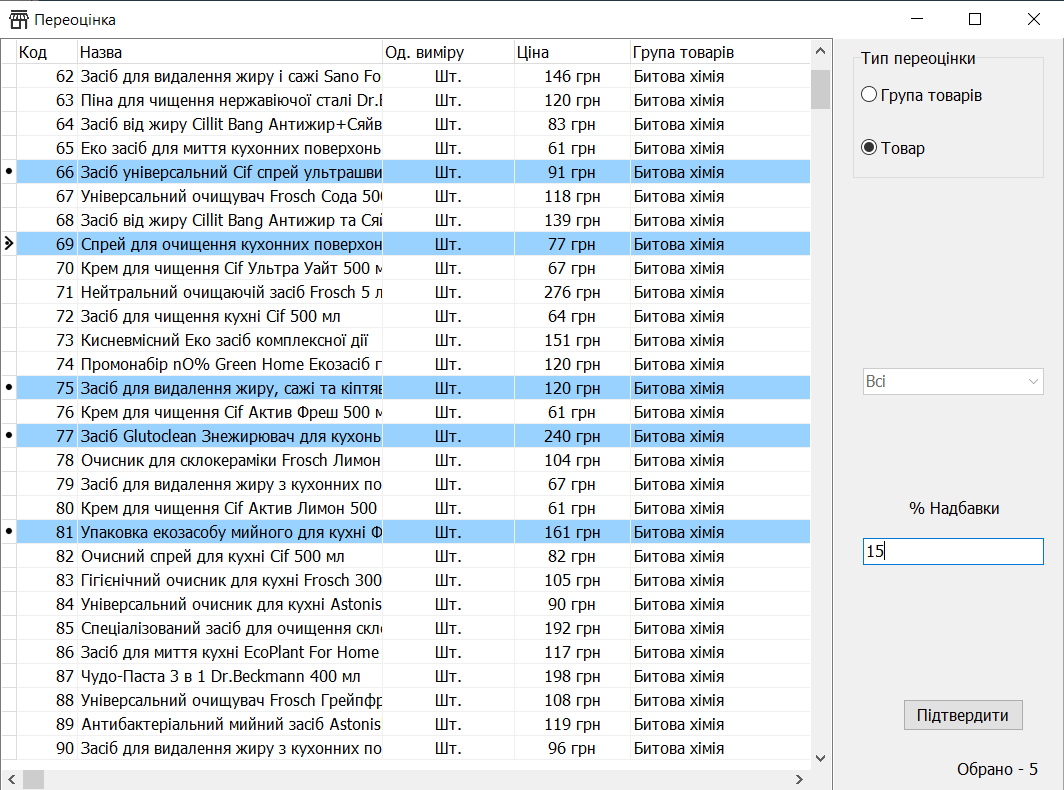


Рисунок А.1 – Форма для переоцінки товарів

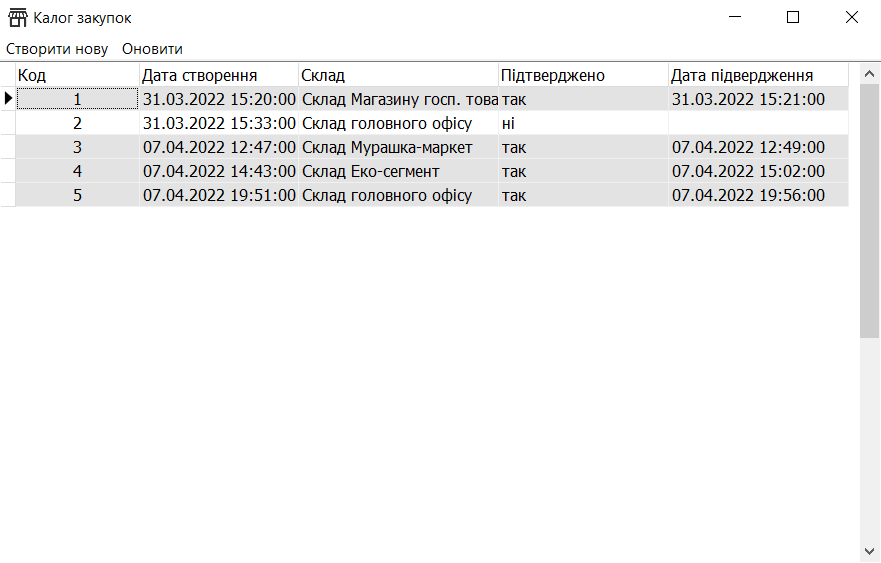


Рисунок А.1 – Форма каталогу закупок

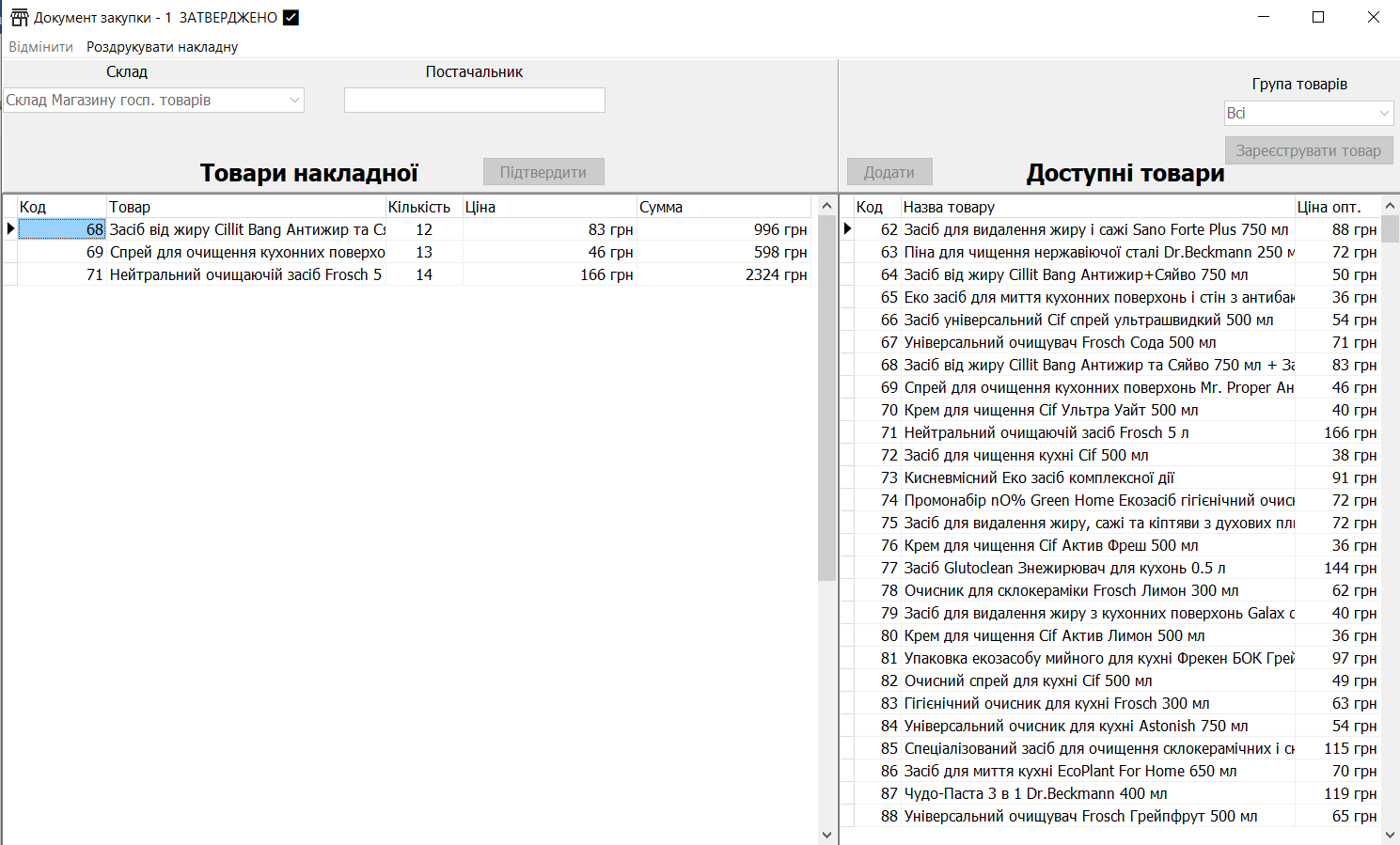


Рисунок А.1 – Форма реєстрації нової закупки

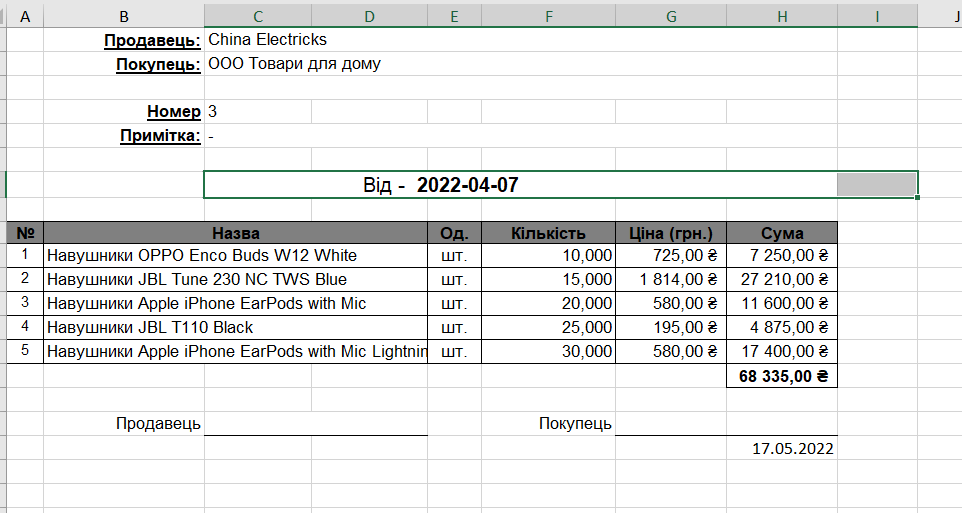


Рисунок А.1 – роздрук документу накладної на закупку

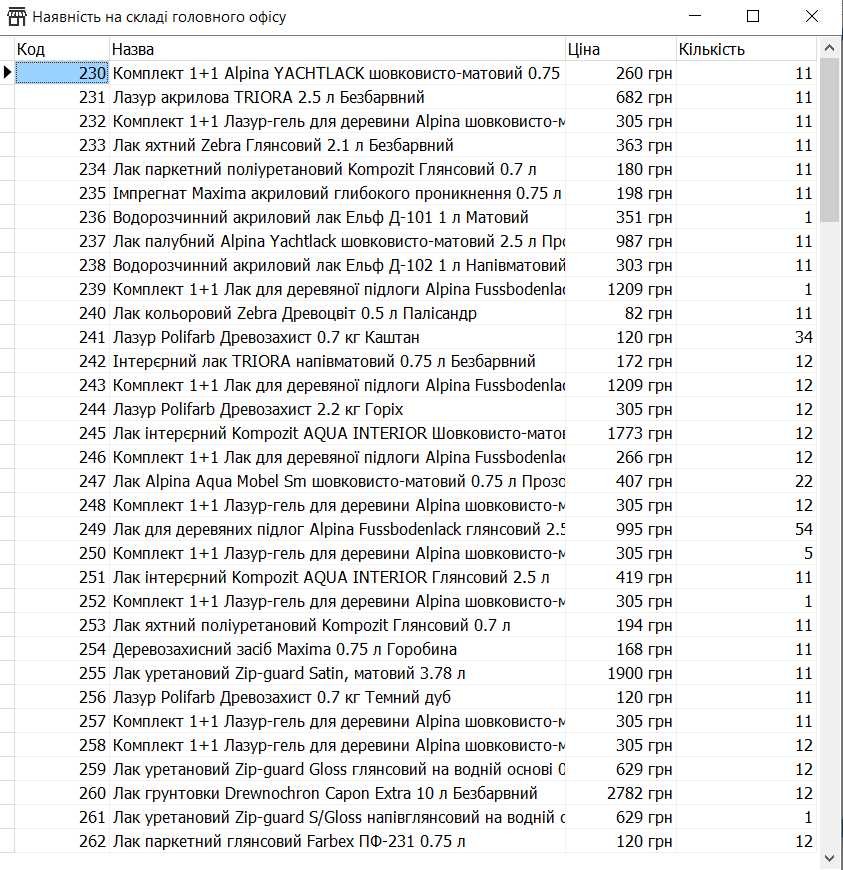


Рисунок А.1 – Форма форма наявності товару на конкретному складі

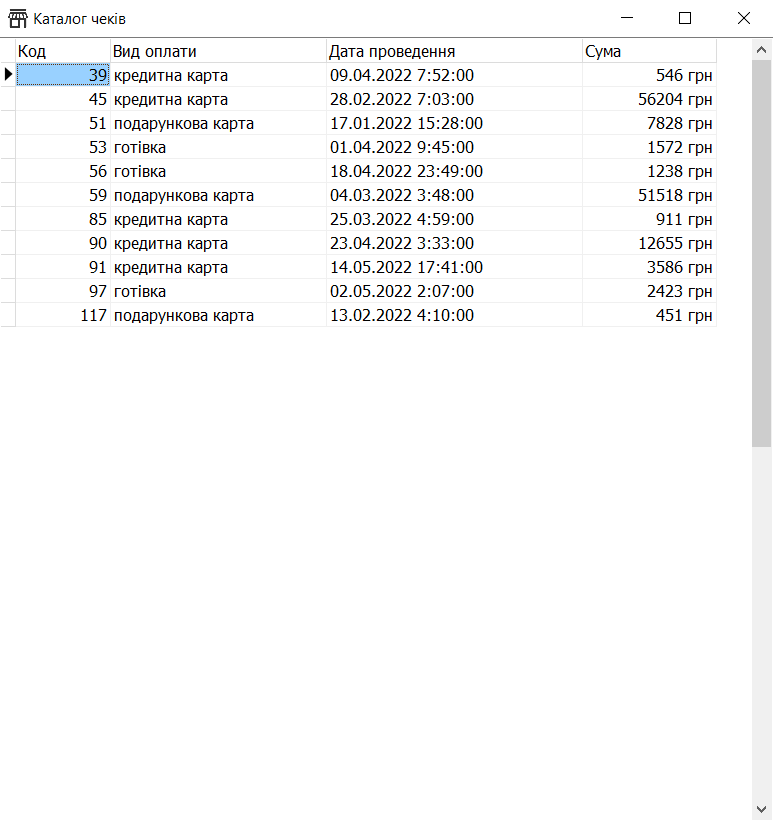


Рисунок А.1 – Форма каталогу чеків на конкретному магазині

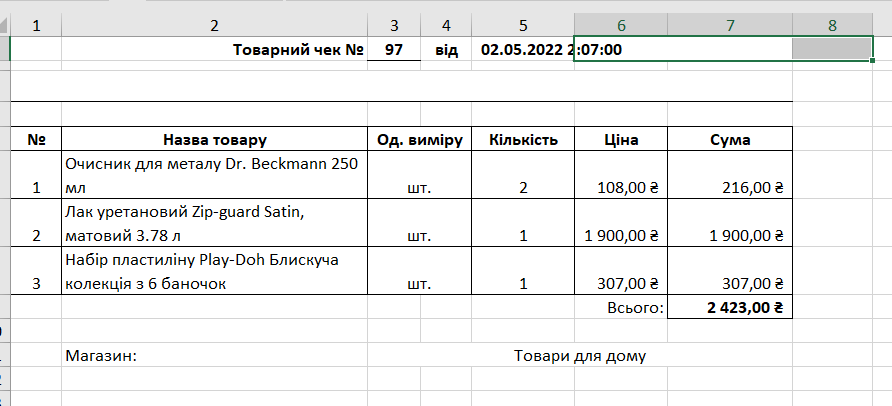


Рисунок А.1 – Роздрук чеку