ЗМІСТ

[ВСТУП 3](#_Toc435470225)

[1 ПОСТАНОВКА ЗАДАЧІ 4](#_Toc435470226)

[2 ХАРАКТЕРИСТИКА ПРОГРАМНИХ ЗАСОБІВ РЕАЛІЗАЦІЇ ПРОЕКТУ 7](#_Toc435470227)

[2.1 Опис середовища та мови програмування розробки програмного продукту 7](#_Toc435470228)

[2.2 Опис використаної СУБД в курсовому проекті 9](#_Toc435470229)

[3 ПРОЕКТУВАННЯ ТА РЕАЛІЗАЦІЯ ПРОГРАМНОГО ПРОДУКТУ 10](#_Toc435470230)

[3.1 Опис структури бази даних 10](#_Toc435470231)

[3.2 Проектування інтерфейсу програми 13](#_Toc435470232)

[3.4 Структурна схема взаємодії складових програми 32](#_Toc435470233)

[3.5 Контроль коректності вхідних та вихідних даних 33](#_Toc435470234)

[4 РОЗРОБКА ІНСТРУКЦІЇ КОРИСТУВАЧА 36](#_Toc435470235)

[ВИСНОВКИ 48](#_Toc435470236)

[СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 49](#_Toc435470237)

Додаток А 50

Додаток Б 52

Додаток В 53

# ВСТУП

Облік зерна на елеваторі досить трудомісткий процес, що вимагає великої кількості паперів, через що пошук і підрахунок потрібної інформації вимагає багато часу, що призводить до збільшення кількості помилок. У зв’язку з цим була б актуальна програма, що автоматизує процес обліку зерна на елеваторі. Вона не повинна вимагати спеціальних знань від користувача та повинна бути призначення для використання на елеваторах та на малих агропідприємствах для ведення обліку зерна. Програма повинна забезпечувати функції редагування даних, фільтрування, сортування, побудови звітів.

# 1 ПОСТАНОВКА ЗАДАЧІ

Тема даного курсового проекту – «Розробка програмного додатку обліку роботи елеватора».

В рамках даного курсового проекту потрібно організувати базу даних для завантаження і вивантаження елеватора, розробити програмний додаток який дозволяє працювати з розробленою базою даних.

Програма повинна виконувати такі функції:

* додавання інформацію про завантаження зерна на склад в базу даних ;
* додавання інформацію про вивантаження зерна зі складу в базу даних;
* видалення інформацію про завантаження/вивантаження з бази даних;
* перегляд інформацію про дані зерна в сховищі;
* сортування по даті завантаження елеватору, дата відвантаження елеватору, типу зерна, вазі;
* фільтрування записів за номерами сховищ, де зберігається зерно, за типами зерна;
* виведення інформації про завантаження/вивантаження зерна у введених користувачем діапазонах дат;
* створення звіту з результатами обліку роботи складу.

Крім того, програма повинна перевіряти усі введенні дані на коректність та видавати повідомлення у випадку помилкового вводу.

Початкові дані містять:

1 Дані про завантаження складу:

* назва культури (строкове значення);
* вага (ціле значення);
* грязність (ціле значення);
* вологість(ціле значення);
* клейковина (ціле значення);
* номер складу(ціле значення);
* дата прибуття(значення дата);
* тип прибуття на склад(автомобіль або потяг).

1. Вивантаження зерна:

* назва культури(строкове значення);
* вага (ціле значення);
* грязність(ціле значення);
* вологість(ціле значення);
* клейковина (ціле значення);
* дата відвантаження(ціле значення);
* номер сховищу(ціле значення);
* тип відправлення(автомобіль або потяг).

1. Сільськогосподарські культури:

* назва культури(строкове значення);
* тип культури (зернова, круп'яна, технічна);
* призначення (посівна, круп'яна, фуражна);

1. Сховища:

* номер(ціле значення);
* об’єм( ціле значення);
* вентиляція(наявність вентиляції);
* тип завантаження (строкове значення);
* завантаження/вивантаження залізною дорогою (наявність такого завантаження);
* завантаження/вивантаження автомобільним транспортом (наявність такого завантаження);
* назва культури (строкове значення);
* об’єм продукції на складі( ціле значення);

Мінімальні вимоги до конфігурації персонального комп’ютеру:

* тип процесора Intel Pentium 1400 MHz або краще;
* обсяг оперативного запам'ятовуючого пристрою 256 Мб і більше;
* обсяг вільного місця на жорсткому диску 124 Мб;
* операційна систем Microsoft Windows XP,Vista, 7.

Рекомендовані вимоги до конфігурації персонального комп’ютеру:

* тип процесора Intel Pentium 1600 MHz або краще;
* обсяг оперативного запам'ятовуючого пристрою 512 Мб і більше;
* обсяг вільного місця на жорсткому диску 475 Мб і більше;
* операційна систем Microsoft Windows XP,Vista, 7, 8.1, 10.

# 2 ХАРАКТЕРИСТИКА ПРОГРАМНИХ ЗАСОБІВ РЕАЛІЗАЦІЇ ПРОЕКТУ

# 2.1 Опис середовища та мови програмування розробки програмного продукту

Мною було обране це середовище програмування тому, що воно дозволяє створювати візуальний інтерфейс, має інтуїтивно зрозумілий синтаксис. Також середовище програмування широко використовується та воно добре документовано, що дозволило швидше виконати поставлено задачу.

Embarcadero Delphi (Borland Delphi) — це об'єктно-орієнтоване інтегроване середовище швидкої розробки програмного забезпечення. Середовище Embarcadero Delphi працює під Microsoft Windows і підтримує розробку програмного забезпечення для операційних систем Microsoft Windows x86 та x64, Mac OS x86,Apple iOS та Android на мові програмування Delphi (Object Pascal). Embarcadero Delphi входить до групи засобів розробки Embarcadero RAD Studio. При цьому Delphi дозволяє звести до мінімуму об'єм програмного коду, який вводиться вручну. В склад Delphi входять засоби, необхідні для розробки, тестування та встановлення програм, включаючи велику за обсягом бібліотеку компонентів (VCL *-* Visual Components Library), засоби візуального проектування, шаблони програм і форм. Середовище проектування Delphi є відкритою системою і дозволяє використовувати як компоненти VCL, так і компоненти від сторонніх розробників, або власні компоненти. Також сильною стороною Delphi є можливість використання функцій WinAPI.

В системі Delphi використовується спеціалізована версія мови програмування Паскаль, що постійно вдосконалюється; вона називається Delphi (в шостій і більш ранішніх варіантах системи Delphi вона називалась Object Pascal - "Об'єктний Паскаль"). Ця версія включає набір розширень, орієнтованих тільки на застосування в рамках середовища Delphi і призначених для прискореного створювання програм.

Елементами мови є набори компонентів, які дозволяють створювати додатки за найрізноманітнішими тематиками. Компоненти володіють наборами властивостей, що характеризують їх особливості. Крім властивостей, компоненти містять методи - програмний код, який обробляє значення властивостей та події - повідомлення, які компонент приймає від програми. Всі програми в Delphi XE будуються по наступному принципу: в їхній головній частині з розширенням .DPR зберігається тільки виклик декількох команд, які відкривають головне вікно, а також виконують завершальні дії. Решта всього програмного коду міститься в файлах, що зберігають опис додаткових модулів, які підключаються. Кожен модуль має строго задану структуру, яка зазвичай автоматично генерується системою Delphi XE при його створенні. Модуль складається з чотирьох частин: інтерфейсної частини, частини реалізації (обов'язкова), частини ініціалізації і частини завершення (необов'язкова). Спочатку вказують заголовок модуля - ключове слово Unit, за ним довільну назву модуля (вона повинна співпадати з іменем файлу, в якому модуль зберігається) і кладуть крапку з комою: Unit Testunit; Інтерфейсна частина описує інформацію, яка доступна з інших частин програми, з інших модулів і головної частини. Частина реалізації описує інформацію, яка недоступна з інших модулів. Подібне розділення модуля на частини дозволяє створювати і розповсюджувати модулі у відкомпільованому вигляді (розширення DCU), додаючи до них тільки опис інтерфейсної частини. При цьому внести зміни в такий модуль неможливо, вихідний код, який реалізує описані в інтерфейсній частині можливості, недоступний. Такий підхід дозволяє повторно використовувати раніше написані для інших програм і вже відкоректовані модулі та розмежовує доступ до модуля декількох програмістів, а також дозволяє розбивати програму на набір логічно незалежних модулів. Інтерфейсна частина завжди йде першою і починається з ключового слова interface, а частина реалізації з - implementation.

Базовими елементами мови являються: коментарі, змінні, константи, оператори, типи даних тощо[5].

Таким чином, ви можете використовувати Delphi для створення як найпростіших додатків, на розробку яких потрібно кілька годин, так і серйозних корпоративних проектів, призначених для роботи десятків і сотень користувачів. Причому для цього можна використовувати самі останні віяння у світі комп'ютерних технологій з мінімальними витратами часу і сил.

# 2.2 Опис використаної СУБД в курсовому проекті

Було обране середовище СУБД Accessдля реалізації бази даних в курсовому проекті тому, що Access має великий спектр використання та Delphi має багато можливостей для роботи з базою даних.

Microsoft Access створена на основі реляційної моделі бази даних і призначена для створення швидких, ефективних баз даних, застосовуваних у побуті та бізнесі. Крім того, вона здатна підключатися до інших баз даних, створюючи для вас широкий фронт роботи з даними, незалежно від того, де вони знаходяться.

Для роботи з базами даних Access використовують технологію ADO. ADO (ActiveX Data Objects) - це компонент технології доступу до даних від Microsoft (MDAC - Microsoft компоненти доступу до даних). ADO є універсальним компонентом - даними для нього можуть бути як таблиці MS Access або серверні бази MS SQL, Oracle, так і Microsoft Active Directory служба, XML-файли.

Для отримання даних з бази даних Access використовується мова SQL (Structured Query Language - мова структурованих запитів). SQL - це мова програмування, який дуже нагадує англійський, але призначений для програм управління базами даних. Кожен виконуваний запит насправді заснований на мові SQL [3].

# 3 ПРОЕКТУВАННЯ ТА РЕАЛІЗАЦІЯ ПРОГРАМНОГО ПРОДУКТУ

# 3.1 Опис структури бази даних

Для реалізації бази даних було обрано Microsoft Access, в якій було створено таблиці.

Доцільно було поділити інформацію на декілька таблиць. Таким чином ми маємо таблицю, що містить перелік культур у таблиці Grain, що зображено у таблиці 3.1, дані про склади та їх характеристики містяться у таблиці Storage, що зображено у таблиці 3.2, також дані про завантаження елеватора та вивантаження містяться у таблицях Loading та Unloading, що зображено у таблицях 3.3 і 3.4 відповідно. Схема між таблицями зображена на рисунку 3.1 .

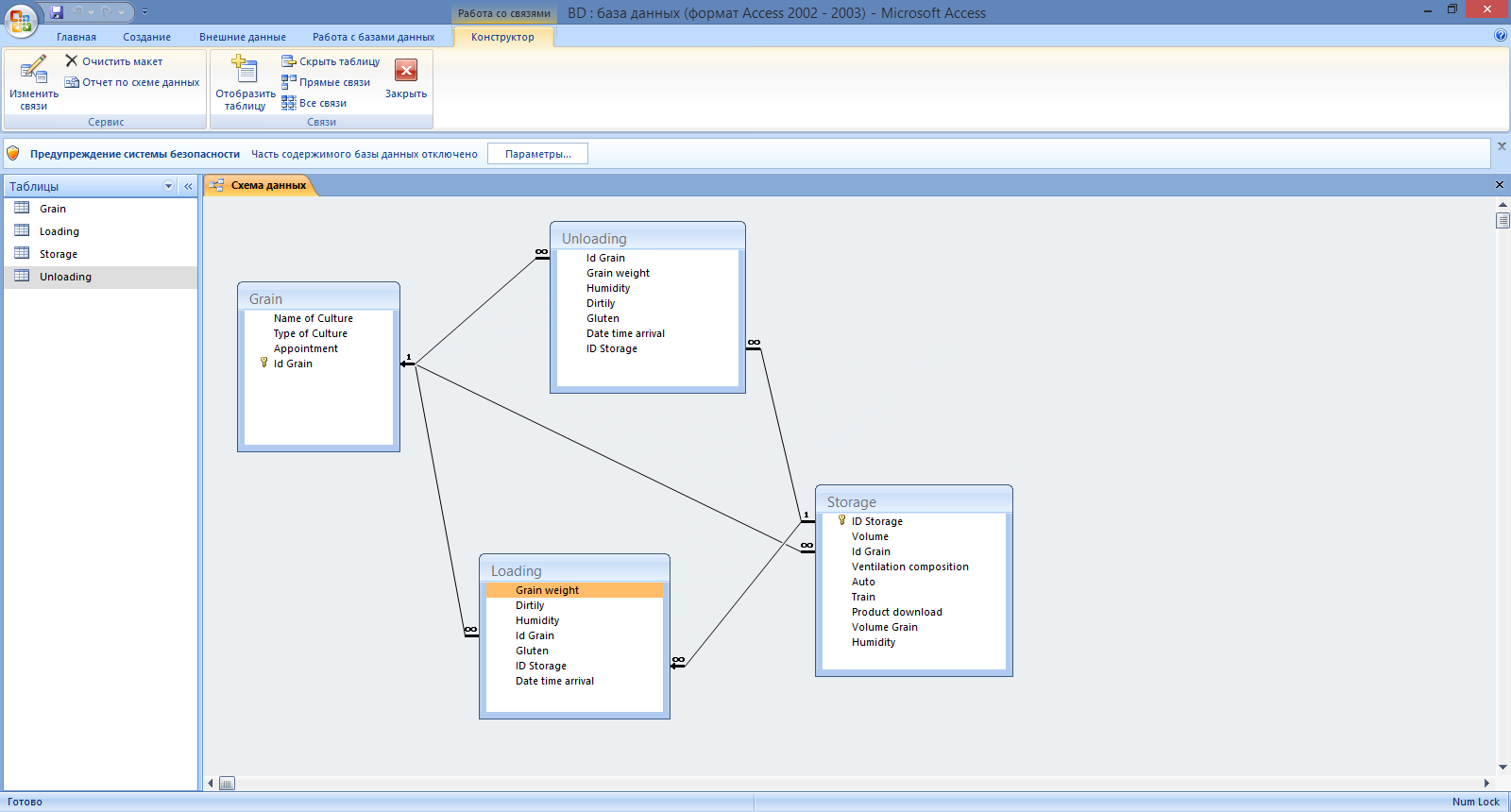


Рисунок 3.1 – Схема зв’язків між таблицями

Таблиця 3.1 – Дані про перелік зерна

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Назва поля | Опис | Тип | Розмір | Ключ |
| Name of Culture | Назва сільськогосподарських культур | Текстовий | 40 |  |
| Type of Culture | Тип культури | Текстовий | 40 |  |
| Appointment | Призначення культури | Текстовий | 40 |  |
| Id Grain | Унікальний ID-код зерна | Чисельний |  | \* |

Таблиця 3.2 – Дані про перелік та характеристики складу

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Назва поля | Опис | Тип | Розмір | Ключ |
| ID Storage | Унікальний ID-код сховищу | Чисельний |  | \* |
| Volume | Об’єм складу | Чисельний |  |  |
| Id Grain | Унікальний ID-код зерна | Чисельний |  | \* |
| Ventilation composition | Вентиляція складу | Логічний |  |  |
| Auto | Завантаження автомобільним транспортом | Логічний |  |  |
| Train | Завантаження залізничним транспортом | Логічний |  |  |
| Product download | Способи завантаження складу | Текстових |  |  |
| Volume Grain | Об’єм зерна | Числовий |  |  |
| Humidity | Вологість | Числовий |  |  |

Таблиця 3.3 – Завантаження елеватора

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Назва поля | Опис | Тип | Розмір | Ключ |
| Id Grain | Унікальний ID-код зерна | Числовий |  | \* |
| Grain weight | Вага зерна | Числовий |  |  |
| Humidity | Вологість | Числовий |  |  |
| Dirtily | Грязність | Числовий |  |  |
| Gluten | Клейковина | Числовий |  |  |
| Date time arrival | Дата прибуття | Дата |  |  |
| ID Storage | Номер сховища | Числовий |  | \* |

Таблиця 3.4 – Вивантаження елеватора

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Назва поля | Опис | Тип | Розмір | Ключ |
| Id Grain | Унікальний ID-код зерна | Числовий |  | \* |
| Grain weight | Вага зерна | Числовий |  |  |
| Humidity | Вологість | Числовий |  |  |
| Dirtily | Грязність | Числовий |  |  |
| Gluten | Клейковина | Числовий |  |  |
| Date time arrival | Дата прибуття | Дата |  |  |
| ID Storage | Номер сховища | Числовий |  | \* |

Поля таблиць, які містять ID не відображається в ході виконання програми, так як служать для зв’язування таблиць за ключовими полями та не несуть змістовного значення для користувача.

# 3.2 Проектування інтерфейсу програми

Для реалізації інтерфейсу Delphi використовує бібліотеку класів, яка містить велику кількість різноманітних класів, що підтримують форму і різні компоненти форми (командні кнопки, поля редагування і т. д.).

Під час проектування форми програми Delphi автоматично додає в текст програми необхідні об'єкти.

Коли додаю необхідні компоненти, створюється форма, Delphi формує опис класу форми. Коли створюється функція обробки події форми або її компонента, Delphi додає оголошення методу в опис класу форми програми.

Крім класів візуальних компонентів в бібліотеку класів входять і класи так званих невізуальних (невидимих) компонентів, які забезпечують створення відповідних об'єктів і доступ до їхніх методів і властивостями.

До складу інтерфейсу програми входять десять форм, призначення яких приведено в таблиці 3.4.

Таблиця 3.4 – Призначення форм

|  |  |
| --- | --- |
| Назва | Призначення |
| Menu1 | Головна форма, що містить меню всієї програми. |
| Form1 | Додавання нового сховища. Викликається з форми Grain\_and\_Storage. |
| Form5 | Додавання нового зерна. Викликається з форми Grain\_and\_Storage. |
| Grain\_and\_Storage | Перегляд інформації про сховища та зерно, також сортування їх та запити до них. Викликається з головного меню. |
| Editing | Редагування інформації про сховища та зерно. Викликається з головного меню. |
| Moduls | Не візуальна форма, призначена для розташування компонентів ADOConnection, ADOQuery та ін. |
| About | Показ вікна «Про програму». Викликається з головного меню. |
| Grain | Перегляд інформації про завантаження в сховище та вивантаження зі сховища зерна. Також сортування та запити до них. Викликається з головного меню. |
| Loading\_\_Elevator | Для завантаження та вивантаження зерна на сховища чи зі сховища. Викликається з головного меню. |
| Preview\_Grain | Побудова звітів по до таблиць. Викликається з головного меню. |

Згідно з постановкою задачі було розроблено інтерфейс програми, який описано нижче.

На формі Menu1, що зображено на рисунку 3.2 розміщено такі компоненти MainMenu1, Image1, Label1.

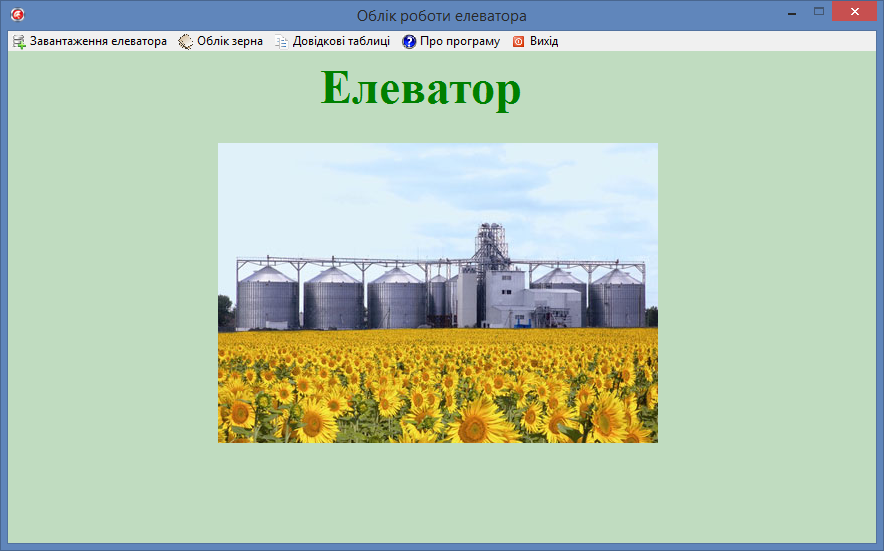


Рисунок 3.2 – Вигляд головної форми

Для форми Menu1 були встановлено властивість Position – poScreenCenter;

Створення головного меню зображено на рисунку 3.3.

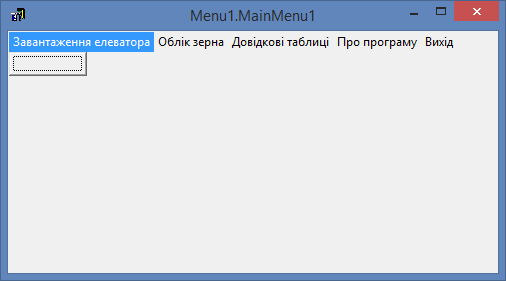


Рисунок 3.3 – Створення головного меню

Компонент MainMenu1 призначений для організації меню переходу між формами та містить такі пункти:

– «Завантаження елеватора» — викликає форму Loading\_\_Elevator;

– «Облік зерна» має підпункти «Перегляд всього списку» та «Звіти по завантаженню і вивантаженню зерна», які відповідно викликають форми Grain та Preview\_Grain;

– «Довідкові таблиці» має підпункти «Перегляд і Додавання» та «Редагування і Видалення», які відповідно викликають форму Grain\_and\_Storage та Editing;

– «Про програму» — викликає форму About;

– «Вихід» — закриває програму.

Форма Grain\_and\_Storage призначена для перегляду даних таблиць «Зерно» та «Сховища» що зображено на рисунках 3.4, 3.5 почергово, сортування та запитів. Вона містить дві сторінки для кожної таблиці відповідно.

Компонент PageControl1 призначений для перегляду даних таблиць незалежно одна від одної, сторінки мають такі назви: «Таблиця зерно», «Таблиця сховищ», які зображено на рисунку 3.4 та 3.5 відповідно.

Компонент DBGrid1 і до DBGrid2 призначенні для виведення інформації з бази даних в таблицю.

Компонент MainMenu1 має тільки один підпункт – для виходу.

Sort\_Grain

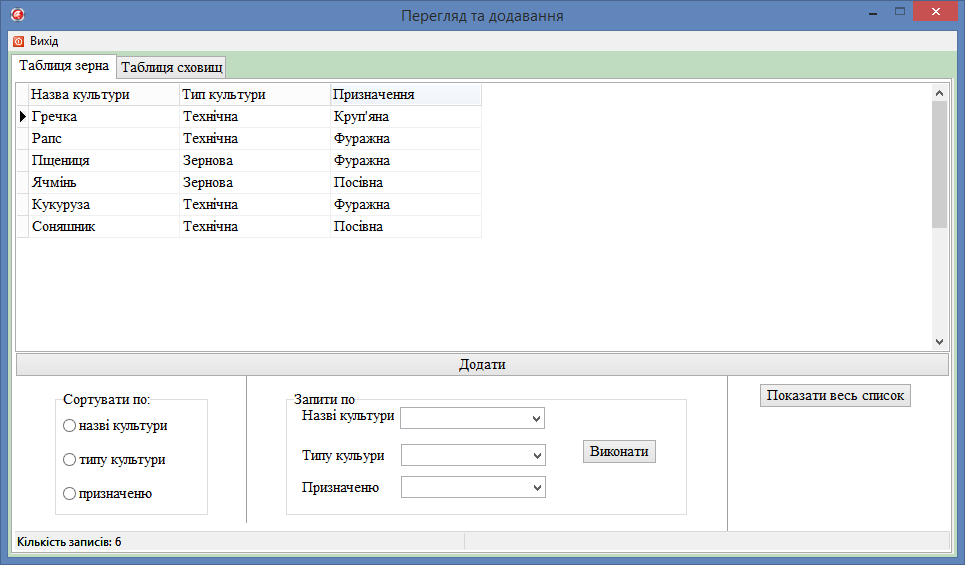
GroupBox2

DBGrid1

Button1

Button3

PageControl



StatusBar1

Рисунок 3.4 – Форма «Перегляд та додавання» сторінка 1

Компонент Button3 і Button4 призначення для виводу всієї таблиці з бази даних, а компонент Button1 та BitBtn1 призначення для виклику форм для додавання Зерна та Сховищ у відповідні таблиці.

Компонент Sort\_Grain та Sort\_Storage типу TRadioGroup призначено для вибору сортування таблиць.

Компонент StatusBar1 та StatusBar2 призначені для виведення кількості записів в таблиці та способи сортування в таблицях відповідно.

Компонент GroupBox2 потрібен для вибору типу запиту до таблиці зерно. В цьому компоненту є чотири вкладених компонента - Name\_Culture, Type\_Culture, Appointment типу TComboBox, для вибору зі списку запитів та компонент Zapros типу TButton, для виконання цих запитів.

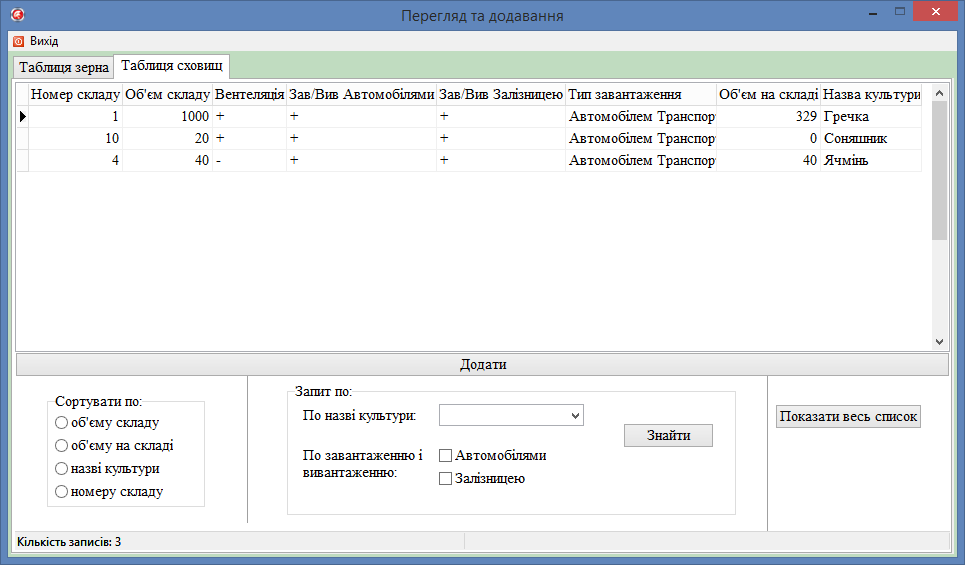
Sort\_Storage

GroupBox1

BitBtn1

Button4

DBGrid2



StatusBar1

Рисунок 3.5 – Форма «Перегляд та додавання» сторінка 2

Компонент GroupBox1 потрібен для вибору типу запиту до таблиці сховищ. В цьому компоненту є ComboBox1 для запиту по назві культури, та два елементи CheckBox1 та CheckBox2 для вибору типу завантаження складу та компонент Button2, для виконання цих запитів.

Форма «Додавання сховища»(Form1) призначена для додавання нових сховищ до таблиці бази даних, вона зображена на рисунку 3.6 .

Для вводу тексту використані компоненти ID\_Storage, Volume типу TEdit, для вибору, чи є вентиляція на складі та способи завантаження використані елементи Ventilation\_composition, Auto, Train, CheckBox1, CheckBox2 типу TCheckBox.

Компонент Name\_Culture\_Storage типу ComboBox використовується для вибору продукції, яка буде зберігатися. Компоненти Save та Cancel типу BitBtn використовуються для збереження чи відміни ввведення відповідно.

BitBtn

ComboBox

CheckBox

TEdit

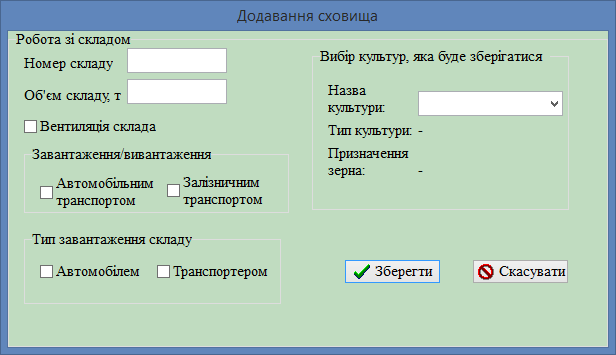


Рисунок 3.6 – Форма «Додавання сховища»

Форма «Додавання культури» призначена для додавання нових культури до таблиці бази даних, вона зображена на рисунку 3.7 .

Для ведення тексту використовується компонент Name\_Grain типу TEdit, а для вибору типу та призначення культури використовуємо компоненти Type\_Culture та Appointment типу RadioGroup.

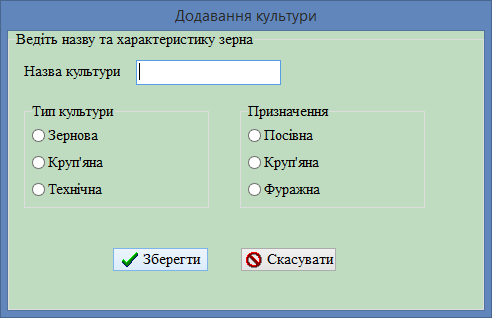


Рисунок 3.7 – Форма «Додавання культури»

Форма Moduls призначена для підключення всіх невізуальних компонентів, таких як ADOConnection, Datasource та ADOQuery, що зображені на рисунку 3.8.

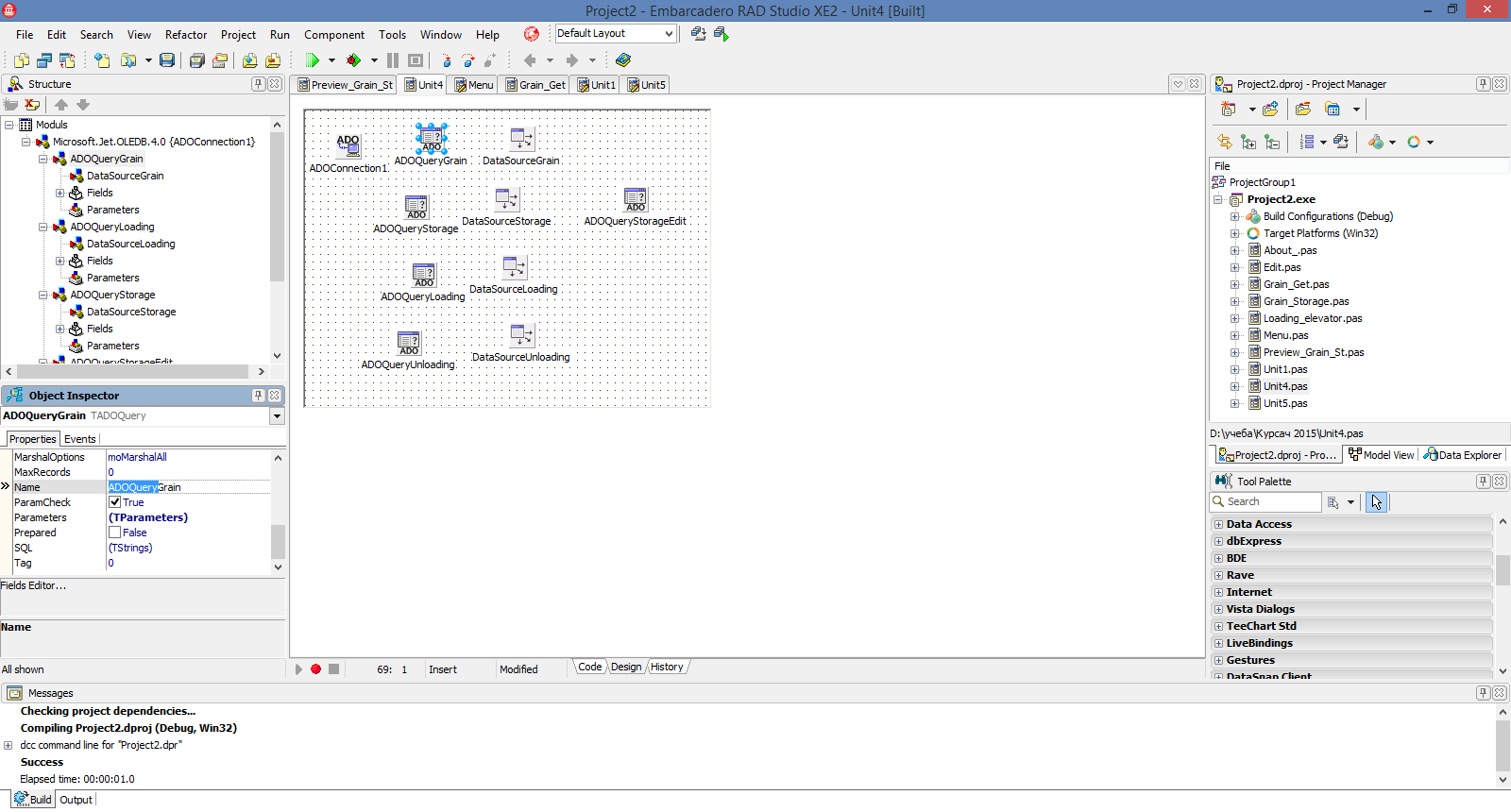


Рисунок 3.8 – Форма модуля даних

Форма «Редагування та видалення» призначена для редагування даних таблиць про культури та сховища та їх видалення, вона відображено на рисунку 3.9 та 3.10.

BitBtn

DBGrid1

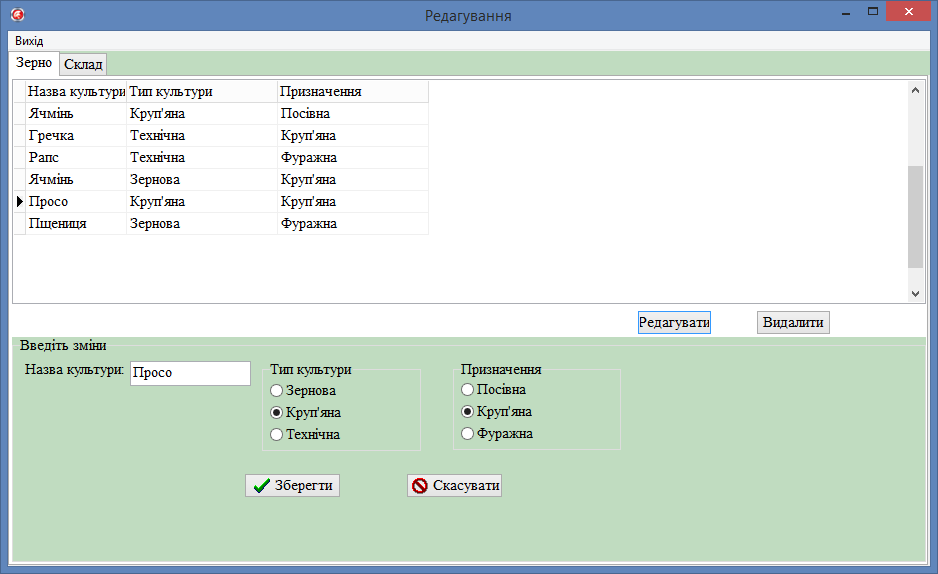


Рисунок 3.9 – Вигляд першої сторінки редагування та видалення

Компонент MainMenu1 має підпункт «Вихід» – для виходу з підпрограми та PageControl1 для переходу між таблицями.

Кнопки Button3 та Delete\_Grain типу Button призначені для редагування та видалення записів відповідно.

Компонент DBGrid1 та DBGrid2 призначений для перегляду даних про культуру.

Компоненти Save та Cancel типу BitBtn використовуються для зберігання та відміни події відповідно.

Компоненти Name\_Culture, Type\_Culture та Appointment призначені для редагування інформації про культури.

DBGrid2

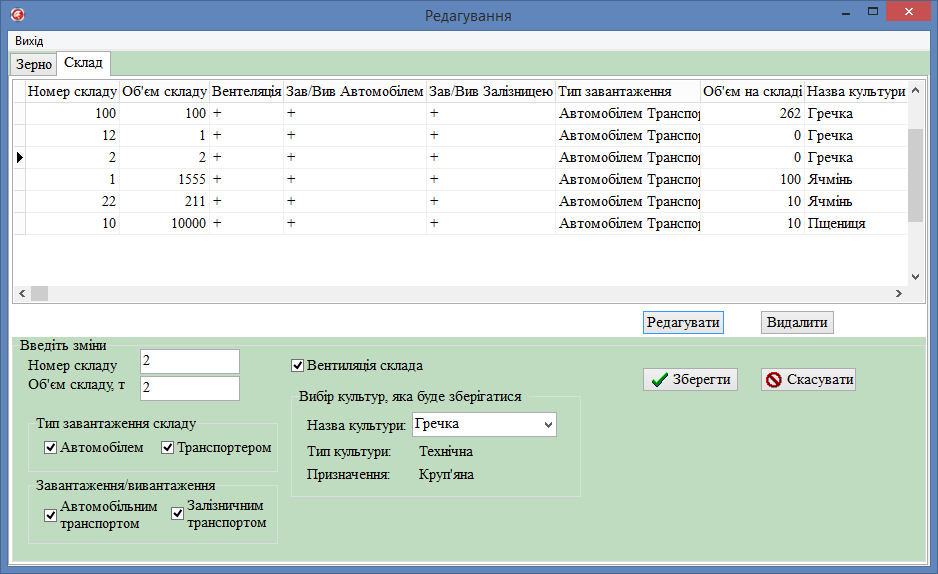


Рисунок 3.10 – Вигляд другої сторінки редагування та видалення

Для редагування використовуються компоненти ID\_Storage та Volume типу TEdit для ведення тексту, Name\_Culture\_Storage типу TComboBox для вибору культури, та для вибору чи є вентиляція на складі та способи завантаження використані елементи Ventilation\_composition, Auto, Train, CheckBox1, CheckBox2 типу TCheckBox.

Для зберігання та скасування використовується Seve\_Storage та Cencel\_Storage типу TBitBtn.

Форма «Про програму» (About) призначена для показу інформації про програму, її показано на рисунку 3.11.

MainMenu1

Image1

Lebel

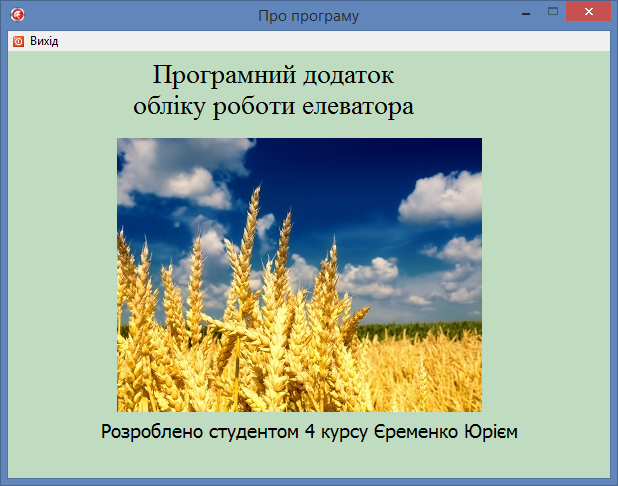


Рисунок 3.11 – Форма «Про програму»

Містить компонент MainMenu1 з підпунктом «Вихід» — для виходу з підпрограми.

Компонент Image1 для показу картинки.

Компоненти Label1, Label2 для пояснювального тексту.

Форма «Завантаження елеватора»( Loading\_\_Elevator) призначення для завантаження та вивантаження культур на елеваторі, додавання на сховище та видалення зі сховища. Має дві вкладки – «Завантаження елеватора» та «Вивантаження елеватора», зображено на рисунках 3.12 та 3.13 відповідно.

DateTimePicker

Edit

ComboBox

CheckBox

RadioGroup

GroupBox

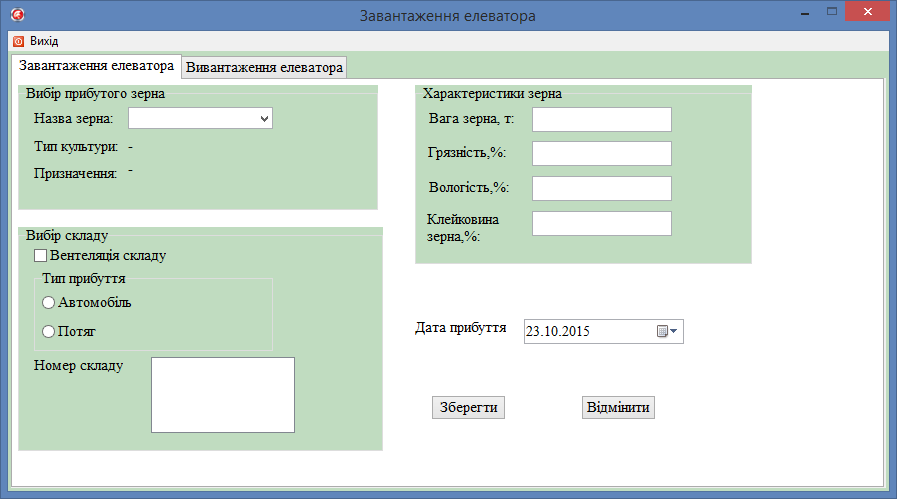


Рисунок 3.12 – Перша вкладка форми «Завантаження елеватора»

На формі міститься три GroupBox, на першому містятиться ComboBox, для вибору зерна. Другий ListBox потрібен для вибору сховища та RadioGroup для вибору типу прибуття, також CheckBox для вибору наявності вентиляції.

Третій компонент GroupBox потрібен для вводу характеристик зерна. Він містить компоненти вводу даних Edit.

Компонент DateTimePicker1 міститься на формі, він потрібен для вибору дати прибуття.

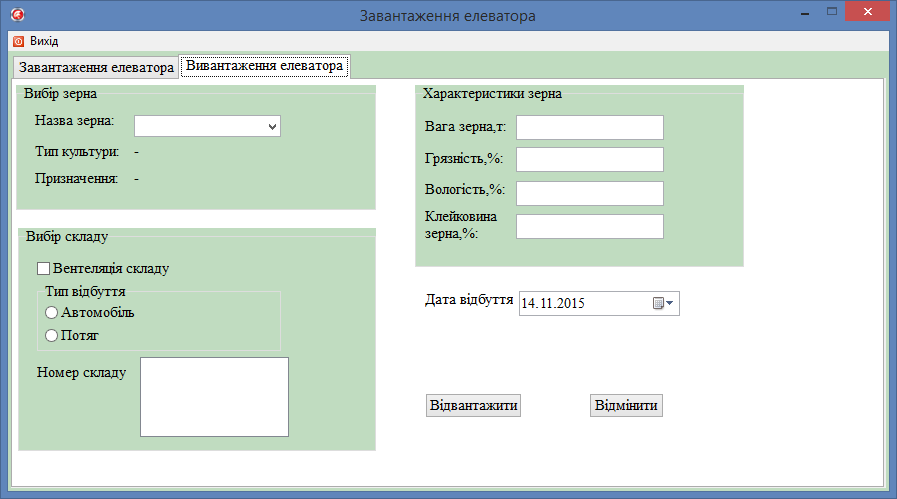


Рисунок 3.13 – Друга вкладка форми «Завантаження елеватора»

На другій вкладці міститься три GroupBox, на першому містититься ComboBox, для вибору зерна. Другий ListBox потрібен для вибору сховища та RadioGroup для вибору типу відбуття, також CheckBox для вибору наявності вентиляції.

Третій компонент GroupBox потрібен для вводу характеристик зерна. Він містить компоненти вводу даних Edit.

Компонент DateTimePicker1 міститься на формі, він потрібен для вибору дати відбуття.

Містить компонент MainMenu1 з підпунктом «Вихід» – для виходу з підпрограми.

Форма «Облік зерна»(Grain) призначена для перегляду даних таблиць про завантаження та вивантаження зерна на елеваторі, створення звітів, сортування та запитів, форма зображена на рисунках 3.14, 3.15 почергово. Вона містить дві сторінки для кожної таблиці відповідно.

DateTimePicker

Poisk

RadioGroup

DBGrid

PageControl

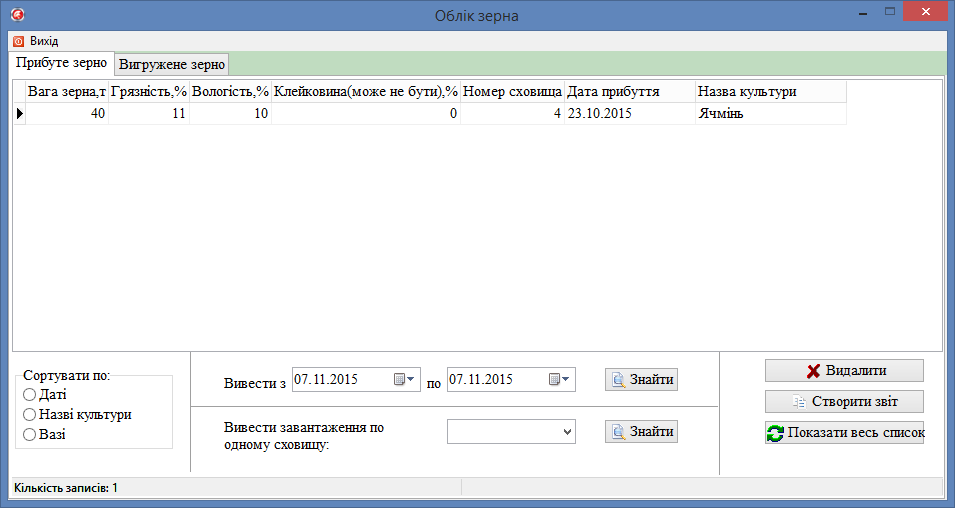


Рисунок 3.14 – Перша вкладка форми «Облік зерна»

Компонент PageControl для переходу між таблицями, що дозоляє з економити місце на формі.

На формі знаходиться DBGrid для виведення таблиці про завантажене зерно на сховища.

Компонент RadioGroup для вибору сортування таблиці.

Компонент DateTimePicker для вводу інтервалу часу та пошуку при натисканні кнопки Poisk типу BitBtn в цьому інтервалі часу завантажень елеватора.

StatusBar

BitBtn

ComboBox

DateTimePicker

RadioGroup

DBGrid

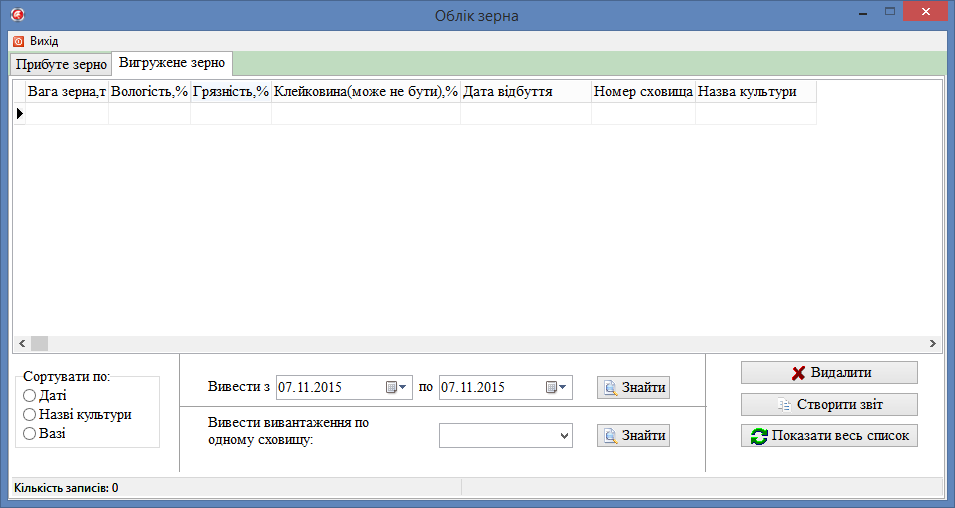


Рисунок 3.15 – Друга вкладка форми «Облік зерна»

На формі знаходиться DBGrid для виведення таблиці про завантажене зерно на сховища.

Компонент RadioGroup для вибору сортування таблиці.

Компонент DateTimePicker для вводу інтервалу часу та пошуку при натисканні кнопки Poisk типу BitBtn в цьому інтервалі часу завантажень елеватора.

Кнопка «Створити звіт» викликає форму «Створення звітів» та звіт формується уже з відсортованими чи даними по запиту.

Компонент ComboBox для виведення відвантаження по складам при натисканні кнопки Faind\_Storage типу BitBtn.

Містить компонент MainMenu1 з підпунктом «Вихід» – для виходу з підпрограми.

Компонент StatusBar відображає кількість записів в таблиці та сортування в таблиці.

Форма «Створення звітів» призначення для створення звітів про завантаження та вивантаження зерна зі сховищ, експортувати ці звіти в більш зручний формат — Word, PDF, Excel, зображено на рисунку 3.16.

PageControl

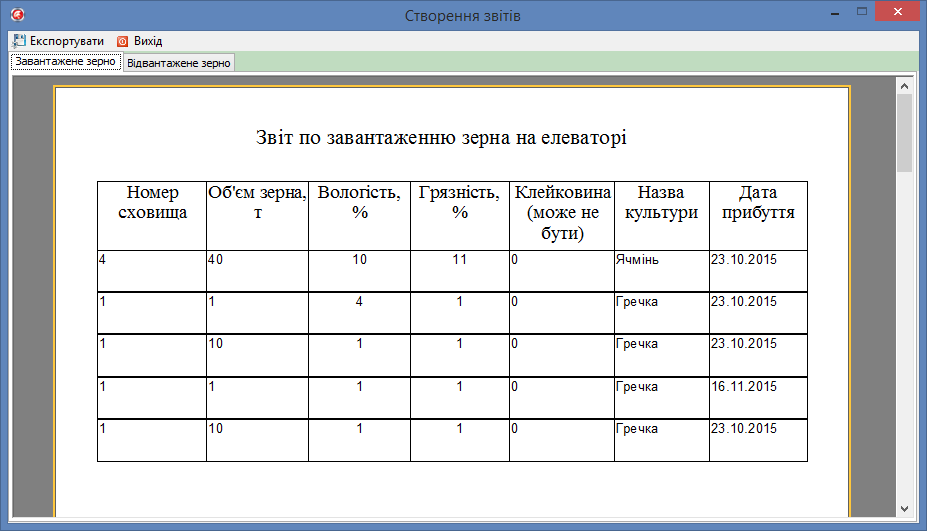


Рисунок 3.16 – Форма створення звітів

Компонент MainMenu1 з підпунктом «Імпортувати» та «Вихід» – для виходу з підпрограми.

Підпункт «Експортувати» має свої підпункти для вибору в який формат експортувати, підпункти мають такі назви «В PDF-файл», «В Word-файл» та «В Excel-файл». А ці підпункти визивають функції для експортування.

Компонент PageControl для переходу між звітами завантаженим та вивантаженим зерном.

3.3 Опис складових частин програми

Програма містить 9 модулів, їх опис приведено у таблиці 3.5.

Таблиця 3.5 – Вміст модулів проекту

|  |  |
| --- | --- |
| Назва | Перелік функцій |
| Menu | N1Click, N3Click, N5Click, N6Click, N7Click, N8Click, N9Click |
| Preview\_Grain\_St | Excel1Click, N3Click, N4Click, PDF1Click |
| Grain\_Get | N1Click, Button1Click, SortClick, FormActivate, PoiskClick, Sort\_2Click, Faind\_DateClick, Button2Clic, FormCreate, Faind\_StorageClick, Faind\_Storage\_2, Button4Click |
| About\_ | N1Click |
| Edit | Button3Click, FormCreate, SaveClick, CancelClick, Edit\_StorageClick, Cancel\_StorageClick, Seve\_StorageClick, Delete\_GrainClick, Delete\_StorageClick, Name\_Culture\_StorageChange, N1Click |
| Grain\_Storage | Button1Click, Button2Click, N1Click, BitBtn1Click, Sort1Click, FormCreate, ZaprosClick, Sort\_Storage\_Click, FormActivate, Button3Click, Button4Click, Zapros\_StorageClick |
| Loading\_elevator | Button1Click, Button2Click, Button3Click, ComboBox1, Name\_Grain2, N1Click, Name\_Grain, PageControl1, RadioGroup1Click, RadioGroup2Click. |
| Unit1 | SaveClick, CancelClick, FormCreate, Name\_Culture\_StorageChange, |
| Unit5 | SaveClick, CancelClick. |

Більш детально призначення підпрограм форм описано нижче.

Таблиця 3.6 – Призначення підпрограм форми Menu

|  |  |
| --- | --- |
| Підпрограми/функції | Призначення |
| N9Click | Закриття програми |
| N8Click | Виклик форми про програму |
| N5Click | Виклик форми перегляду та додавання довідкових таблиць |
| N6Click | Виклик форми редагування та видалення довідкових таблиць |
| N3Click | Виклик форми перегляду всього списку завантаженого та вивантаженого зерна зі складу |
| N7Click | Виклик форми звітів |
| N1Click | Виклик форми завантаження та вивантаження зерна |

Таблиця 3.7 – Призначення підпрограм форми Preview\_Grain\_St

|  |  |
| --- | --- |
| Підпрограми/функції | Призначення |
| Excel1Click | Експортування таблиць в Excel-файл |
| N3Click | Закриття форми |
| N4Click | Експортування таблиць в Word-файл |
| PDF1Click | Експортування таблиць в PDF-файл |

Таблиця 3.8 – Призначення підпрограм форми Grain\_Get

|  |  |
| --- | --- |
| Підпрограми/функції | Призначення |
| Button1Click | Видалення запису в таблиці завантажене зерно |
| N1Click | Закриття форми |
| SortClick | Сортування таблиці завантажене зерно |
| PoiskClick | Запит в таблиці завантажене зерно по даті |
| Sort\_2Click | Сортування таблиці вивантажене зерно |
| Button2Click | Виклик форми створення звіту |
| Button4Click | Виклик форми створення звіту |
| Faind\_StorageClick | Пошук завантаження по сховищу |
| Faind\_Storage\_2Click | Пошук вивантаження по сховищу |
| Faind\_DateClick | Запит в таблиці вивантажене зерно по даті |
| FormActivate | Вивід кількості записів у статусбари |
| FormCreate | Вивід записів в ComboBox |

Таблиця 3.9 – Призначення підпрограм форми About\_

|  |  |
| --- | --- |
| Підпрограми/функції | Призначення |
| N1Click | Закриття форми |

Таблиця 3.10 – Призначення підпрограм форми Edit

|  |  |
| --- | --- |
| Підпрограми/функції | Призначення |
| SaveClick | Зберігання змін при редагуванні культур |
| Seve\_StorageClick | Зберігання змін при редагуванні сховищ |
| CancelClick | Відміна змін при редагуванні культур |
| Cancel\_StorageClick | Відміна змін при редагуванні сховищ |
| Delete\_GrainClick | Видалення зернових культур |
| Delete\_StorageClick | Видалення зернових сховищ |
| Edit\_StorageClick | Редагування сховищ |
| Button3Click | Редагування культур |
| FormCreate | Не активність компонентів |
| N1Click | Закриття форми |
| Продовження таблиці 3.10 | |
| Підпрограми/функції | Призначення |
| Name\_Culture\_StorageChange | Вивід характеристик зерна |
| FormActivate | Вивід кількості записів у статусбари |

Таблиця 3.11 – Призначення підпрограм форми Grain\_Storage

|  |  |
| --- | --- |
| Підпрограми/функції | Призначення |
| BitBtn1Click | Виклик форми додавання сховищ |
| Button2Click | Виклик форми додавання назв зерна |
| Button3Click | Вивід всього списку культур |
| Button4Click | Вивід всього списку сховищ |
| FormCreate | Вивід даних в комбобокси |
| N1Click | Закриття форми |
| Sort1Click | Сортування таблиці зерно |
| Sort\_Storage\_Click | Сортування таблиці сховища |
| ZaprosClick | Запит до таблиці зерна |
| Zapros\_StorageClick | Запит до таблиці сховищ |
| FormActivate | Вивід кількості записів у статусбари |

Таблиця 3.12 – Призначення підпрограм форми Loading\_elevator

|  |  |
| --- | --- |
| Підпрограми/функції | Призначення |
| Button1Click | Завантаження(додавання) зерна на сховище |
| Button2Click | Відвантаження зерна зі сховище |
| ComboBox1Change | Очистка лістбокса |
| Name\_Grain2Change | Вибір зерна та вивід його характеристик для відвантаження |
| N1Click | Закриття форми |
| Name\_GrainChange | Вибір зерна та вивід його характеристик для завантаження |
| PageControl1Change | Вивід даних при переході між вкладками |
| RadioGroup1Click | Вибір типу прибуття та пошук сховищ відповідних |
| RadioGroup2Click | Вибір типу відвантаження та пошук сховищ відповідних |

Таблиця 3.13 – Призначення підпрограм форми Unit1

|  |  |
| --- | --- |
| Підпрограми/функції | Призначення |
| SaveClick | Збереження нового сховища |
| CancelClick | Скасування додавання |
| Продовження таблиці 3.13 | |
| Підпрограми/функції | Призначення |
| FormCreate | Встановлення властивості компонентів |
| Name\_Culture\_StorageChange | Вивід характеристик зерна |

Таблиця 3.14 – Призначення підпрограм форми Unit5

|  |  |
| --- | --- |
| Підпрограми/функції | Призначення |
| SaveClick | Збереження нової сільськогосподарської культури |
| CancelClick | Скасування додавання |

# 3.4 Структурна схема взаємодії складових програми

З головної форми через пункти головного меню викликаються наступні форми:

* «Завантаження елеватора»;
* «Перегляд всього списку»;
* «Звіти по завантаженню і вивантаженню зерна»;
* «Перегляд і Додавання»;
* «Редагування і Видалення»;
* «Про програму»

Схема взаємодії складових часин програми приведена на рисунку 3.17.

Головна форма (Menu)

Завантаження елеватора

Про програму

Вихід

Облік зерна

Перегляд всього списку

Звіти по завантаженню і вивантаженню зерна

Довідкові таблиці

Перегляд і Додавання

Редагування і Видалення

Форма завантаження (Loading\_elevator)

Форма створення звітів(Preview\_Grain\_St)

Форма перегляду всіх завантажень(Grain\_Get)

Форма редагування довідкових таблиць(Edit)

Форма додавання довідкових таблиць(Grain\_Storage)

Форма про програму (About)

Форма додавання зерна (Unit5)

Форма додавання сховищ(Unit1)

Рисунок 3.17 – Схема взаємодії складових частин програми

# 3.5 Контроль коректності вхідних та вихідних даних

Контролю коректності даних, що вводяться користувачем, необхідно приділяти чималу увагу, оскільки необроблені помилки, що виникають при неправильному введенні даних, приводять до помилок в роботі програми.

При проектування програми були усунені деякі помилки, повідомлення про які приведено на рисунках 3.18.

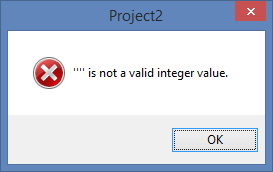


Рисунок 3.18 – Помилка при збереженні пустих полів

Для запобіганню додавання пустих полів до бази даних використано алгоритм, який зображений на лістингу 3.1.

if(Name\_Grain.text='') or(Type\_Culture.ItemIndex<0)or(Appointment.ItemIndex<0) then

ShowMessage('Вы не ввели значения!')

Else

//…

End;

Лістинг 3.1 – Алгоритм запобігання додавання пустих полів

Помилка при збережені букви замість числа зображено на рисунку 3.19.

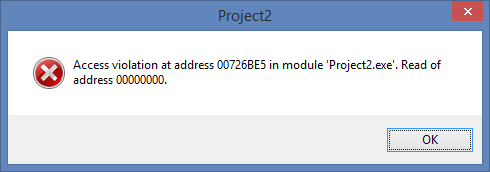


Рисунок 3.19 – Помилка збереженні букви заміть числа

Помилка виникає при ведені букви чи символу де потрібне число. Для вирішення цієї помилки в властивостях елементу заборонив ведення букв.

Для коректності введених даних я використовую перевірку для виявлення некоректний даних, так як неможливо завантажити зерно на сховище більше, чим його об’єм, та відповідно відвантажити більше зерна, чим є на сховищі. Для цього я використовую, який зображений на лістингу 3.2.

Moduls.ADOQueryStorage.first;

for i:=1 to Moduls.ADOQueryStorage.recordcount do

begin

if(Moduls.ADOQueryStorage.fieldbyname('ID Storage').AsInteger=strtoint(ListBox2.Items.Strings[ListBox2.ItemIndex])) then

begin

if((Moduls.ADOQueryStorage.fieldbyname('Volume').AsInteger - Moduls.ADOQueryStorage.fieldbyname('Volume Grain').AsInteger)<strtoint(Grain\_weight.text) ) then

begin Vol:=Moduls.ADOQueryStorage.fieldbyname('Volume').AsInteger - Moduls.ADOQueryStorage.fieldbyname('Volume Grain').AsInteger;

flag:=true;

end;

end;

Moduls.ADOQueryStorage.Next;

end;

if(flag=false) then

begin

...

end;

else ShowMessage('Не хватає місця на сховищі, можливо завантажити на склад '+inttostr(Vol)+',Т');

end;

Лістинг 3.2 – Алгоритм запобігання некоректних даних

На екран виводиться відповідне повідомлення, яке зображене на рисунку 3.20.

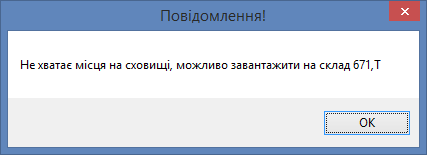


Рисунок 3.20 – Виведення повідомлення

Для коректного виведення даних в зв’язаних таблицях, щоб не виводило ID зернової культури, вивожу її назву. Для цього зробив SQL­­ – запит, який приведений в лістингу 3.3.

SELECT Storage.\* , Grain.[Name of Culture]

FROM Grain, Storage

WHERE Grain.[Id Grain]= Storage.[Id Grain];

Лістинг 3.3 – SQL­­ – запит коректного виведення даних в таблицях

На рисунку 3.21 зображено уже нормальне виведення зв’язаної таблиці.

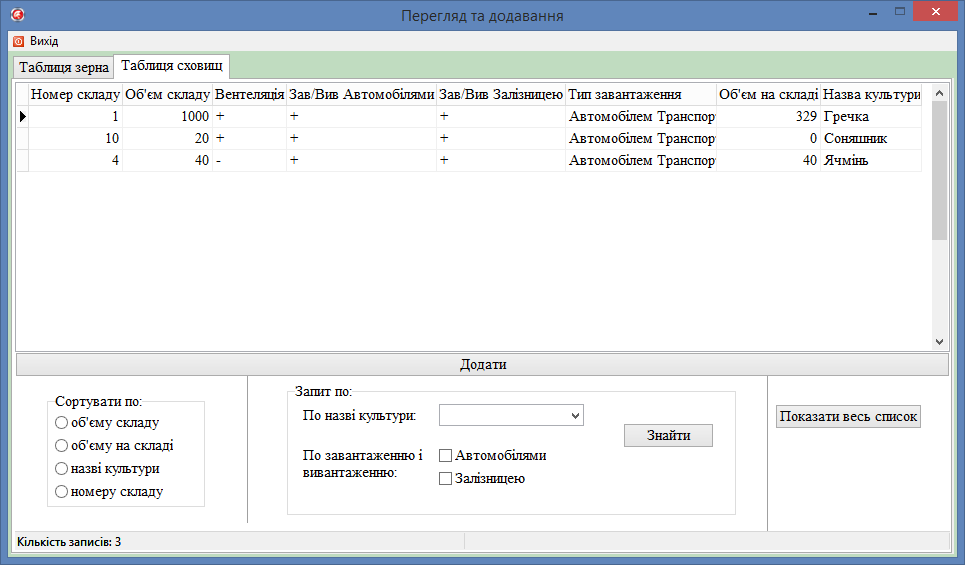


Рисунок 3.21 – Виведення таблиці сховищ с назвою культур заміть ID

# 4 РОЗРОБКА ІНСТРУКЦІЇ КОРИСТУВАЧА

Для запуску програми потрібно двічі клацнути на файл Курсовий проект.exe. Після цього з’явиться головне вікно програми, що зображено на рисунку 4.1.

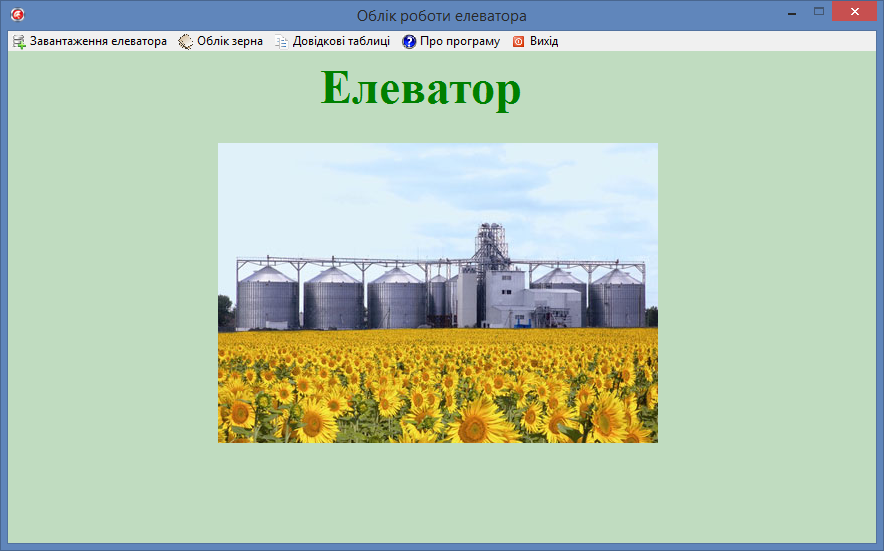


Рисунок 4.1 – Головне вікно програми

Для початку роботи с програмою потрібно вибрати пункт «Довідкові таблиці» та відкрити форму «Перегляд та додавання», що зображена на рисунку 4.2.

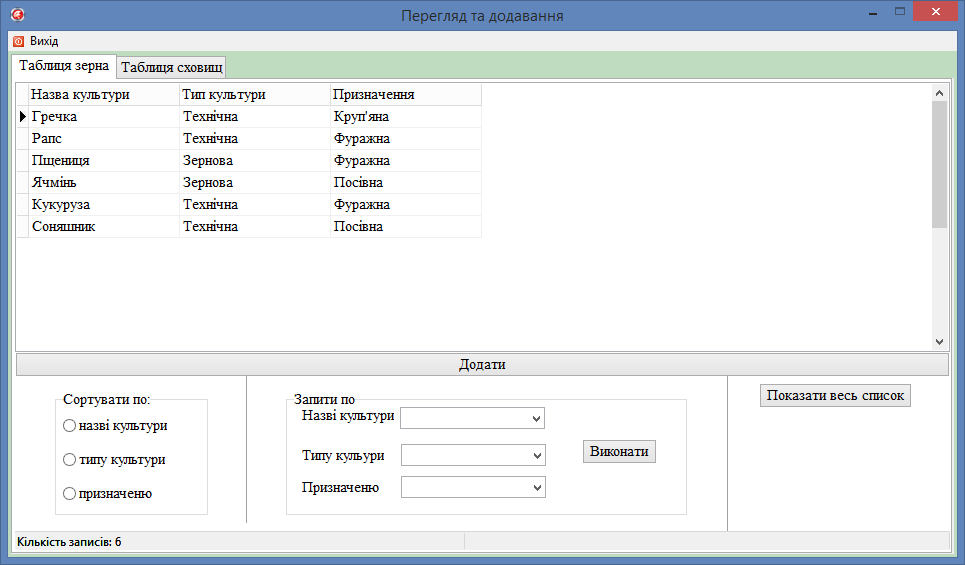


Рисунок 4.2 – Вікно додавання та перегляду довідкових таблиць

На цій формі міститься дві вкладки для переходу між таблицями. На першій вкладці назви культур та їх характеристик та другій вкладці таблиця списку всіх сховищ та їх характеристика.

На першій вкладці, яка зображена на рисунку 4.2, можна сортувати по різним признакам, та виводити записи по запитам. А для початку потрібно додати нову сільськогосподарську культуру, для цього потрібно нажати кнопку «Додати». Відкриється вікно для додавання культур, воно зображене на рисунку 4.3 . На цьому вікні потрібно ввести назву культури, тип та призначення. Після збереження нова культура додається до таблиці.

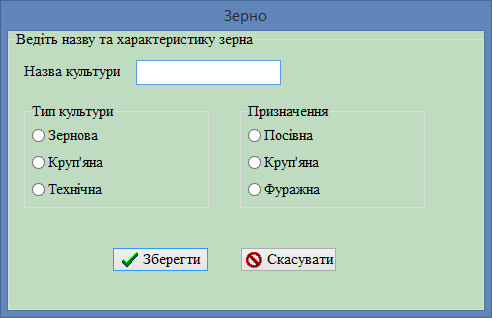


Рисунок 4.3 – Вікно додавання нової культури

На другій вкладці вікна «Перегляд та додавання» знаходиться таблиця зі списком сховищ та характеристиками їх, ця вкладка зображена на рисунку 4.4.

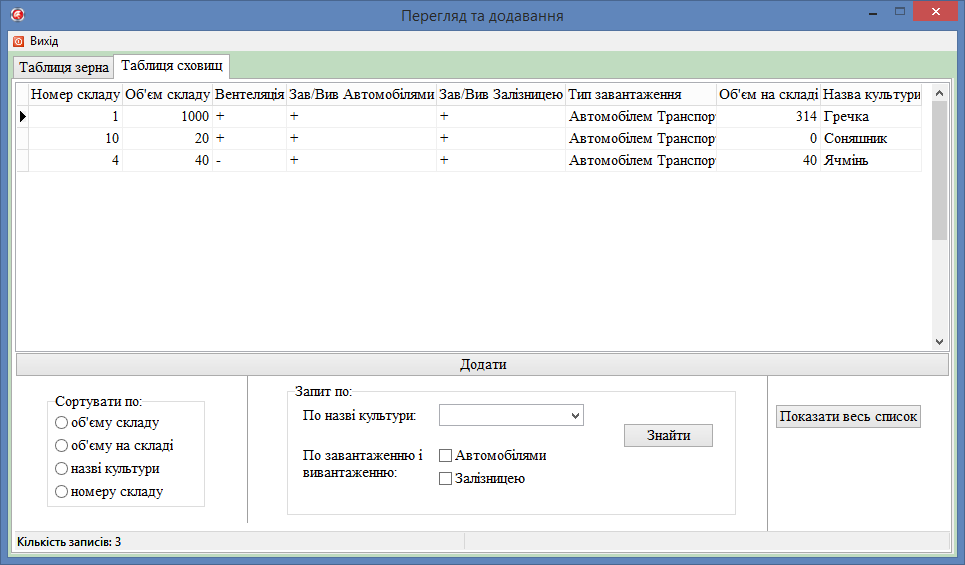


Рисунок 4.4 – Друга вкладка вікна «Перегляд та додавання»

На цій вкладці знаходиться таблиця зі сховищами та с характеристиками до кожного. Можна сортувати цю таблицю по: об'єму складу, об'єму на складі, назві культури, номеру складу. Також можливо запити по культурі та способі завантаження на сховище. А для того щоб відмінити сортування та запити потрібно натиснути кнопку «Показати весь список».

Для того, щоб додати нове сховище до таблиці потрібно натиснути кнопку «Додати» та відкриється вікно додавання сховищ, яке зображене на рисунку 4.5.

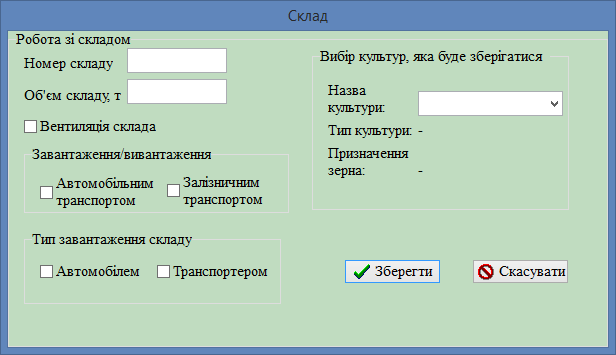


Рисунок 4.5 – Вікно додавання нового сховища

На формі додавання нового сховища потрібно вести номер складу та його об’єм, вибрати чи є вентиляція. Вибрати тип завантаження сховища: автомобільним чи залізничним транспортом. Вибрати тип завантаження сховища уже в самому сховищу: автомобілем чи транспортером. Вибрати зі списку культуру яка буде зберігатися на цьому сховищу. Після всіх ведених даних зберегти запис та він буде відображений в таблиці зі всіма сховищами.

Для редагування та видалення довідкових таблиць виберіть пункт меню «Редагування та видалення» на головній формі. З’явиться вікно з двома вкладками, на першій вкладці редагування та видалення зерна, а на другій вкладці редагування та видалення сховищ, зображені вкладки на рисунках 4.6 та 4.7 відповідно.

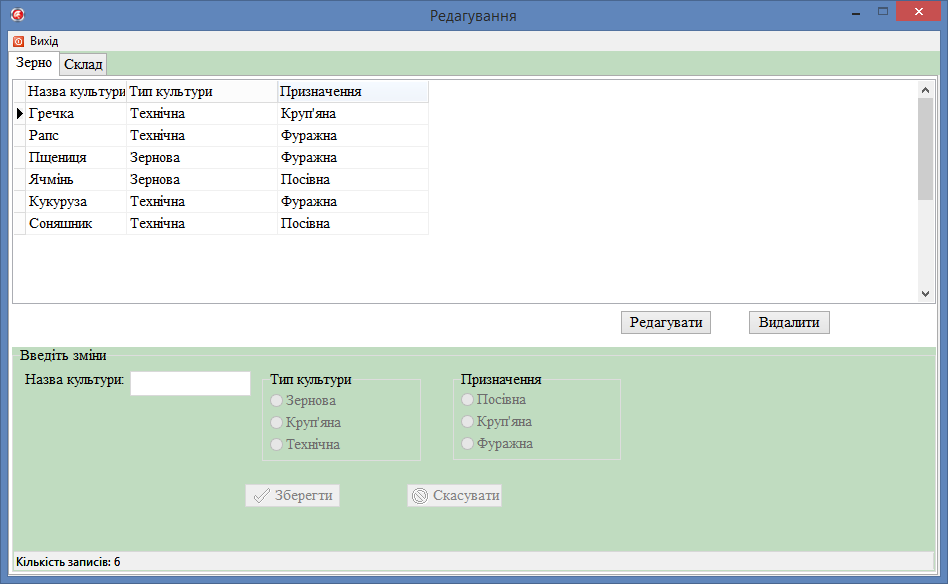


Рисунок 4.7 – Вікно редагування та видалення сільськогосподарських культур

Для редагування виберіть культуру та натисніть редагування, але не можливо редагувати сільськогосподарську культуру, якщо уже є на сховищі. Після ведення змін натисніть «Зберегти» та всі зміни збережуться. А для того,

щоб скасувати дію натисніть «Скасувати».   
 Для видалення культури виберіть потрібний запис, та натисніть «Видалити», після цього запис видалиться з таблиці. Але не можливо видалити запис, якщо зерно уже є на сховищах.

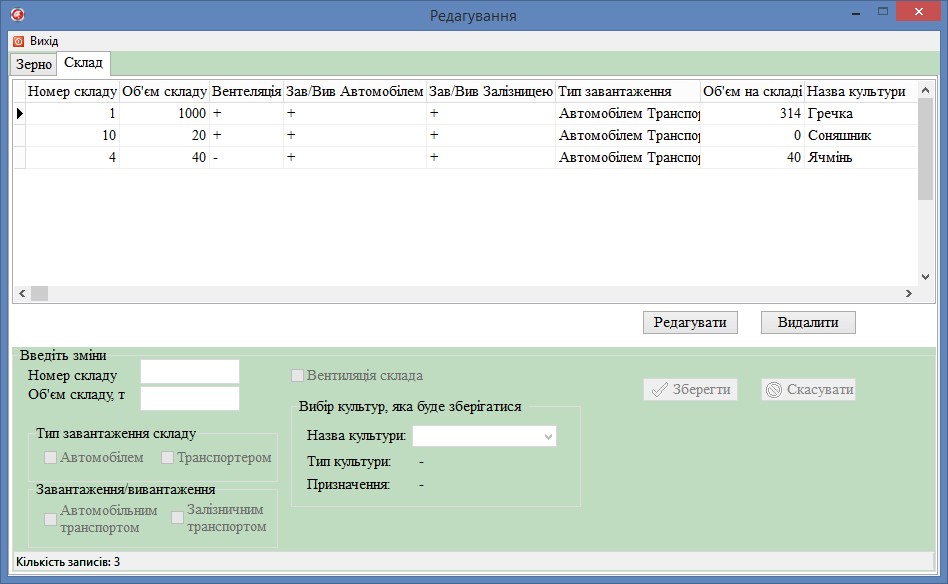


Рисунок 4.8 – Вікно редагування та видалення сховищ

Для редагування сховищ виберіть сховище та натисніть кнопку «Редагувати», після цього введіть зміни. Щоб зберегти зміни натисніть «Зберегти». Для відміни збереження натисніть «Скасувати». Редагувати запис можна тільки, якщо в сховищі не має зерна.

Для видалення сховищ натисніть кнопку «Видалити», після цього запис видалиться з таблиці. Але не можливе видалення, якщо на цьому сховищу є зерно.

При завантаження елеватора потрібно зайти на головне вікно та вибрати пункт меню «Завантаження елеватора». Відкриється форма з двома вкладками, на першій вкладці вивантаження зерна зі сховища, а на другій вкладці вивантаження зерна зі сховищ елеватора, вкладки зображені на рисунках 4.9 та 4.10.

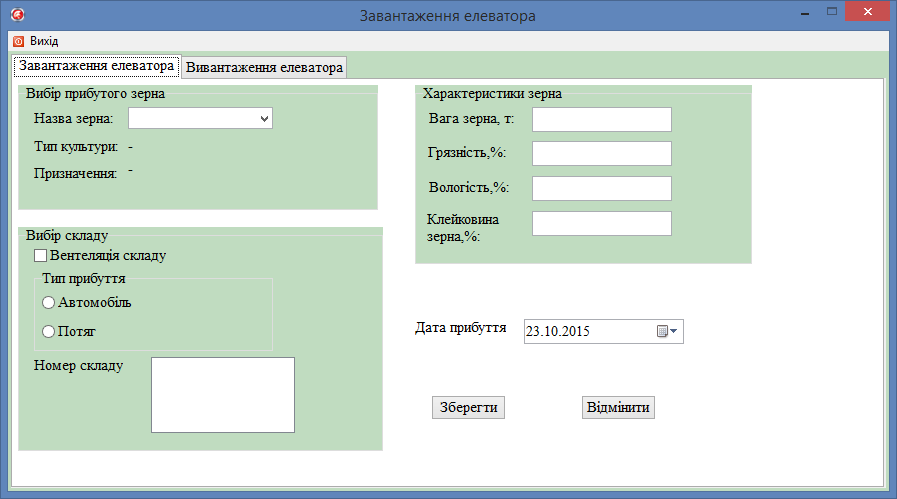


Рисунок 4.9 – Вікно завантаження елеватора

Для того, щоб завантажити зерно на елеватор потрібно вибрати назву зерна зі списку, кожного зерна при виборі відображається його характеристика, потім вибрати чи потрібна вентиляція на сховищі та вибрати спосіб прибуття. Після цих шагів буде запропоновано вибрати номер складу, який підходе, а якщо жодне сховище не підходить для завантажень то буде виведене відповідне повідомлення. Потім потрібно вести характеристики уже прибутого зерна: вагу, забрудненість(не більше 12%), вологість(для технічних культур 14% максимальна, а для всіх інших культур 16) та клейковину, але клейковина тільки для Пшениці потрібна. Після цих шагів вибираємо дату і натискаємо кнопку «Зберегти».

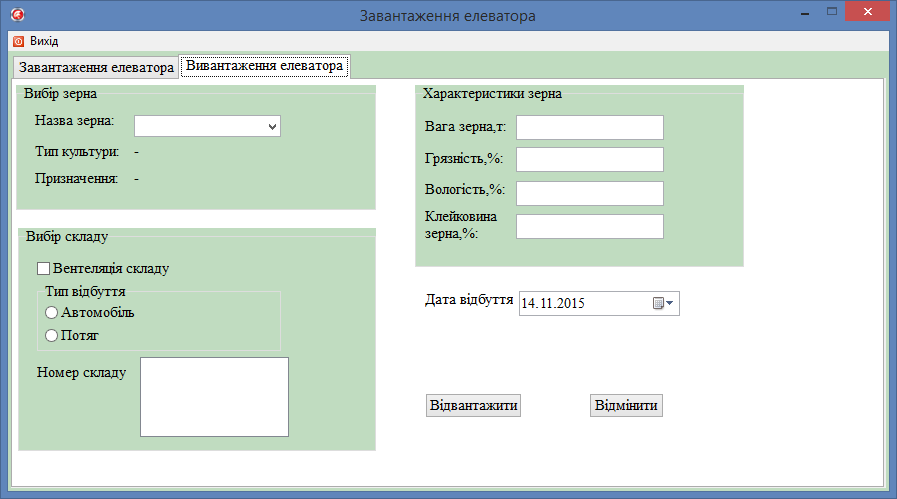


Рисунок 4.10 – Вікно вивантаження елеватора

Для того, щоб вивантажити зерно на елеваторі потрібно вибрати назву зерна зі списку, кожного зерна при виборі відображається його характеристика, потім вибрати чи потрібна вентиляція на сховищі та вибрати спосіб відбуття. Після цих шагів буде запропоновано вибрати номер складу, який підходе, а якщо жодне сховище не підходить для завантажень, то буде виведене відповідне повідомлення. Потім потрібно вести характеристики вже відбутого зерна: вагу, забрудненість(не більше 12%), вологість(для технічних культур 14% максимальна, а для всіх інших культур 16%), вологість та клейковину, але клейковина тільки для Пшениці потрібна. Після цих шагів вибираємо дату і натискаємо кнопку «Зберегти».

Для того, щоб вивести дані про завантаження та вивантаження елеватора потрібно натиснути в головному меню «Облік зерна» та натиснути «Перегляд всього списку». Появиться вікно з двома вкладками, на першій вкладці знаходиться таблиця с прибутим зерном на сховище, а на другій вкладці знаходиться вивантажене зерно, вкладки зображені на рисунках 4.11 та 4.12.

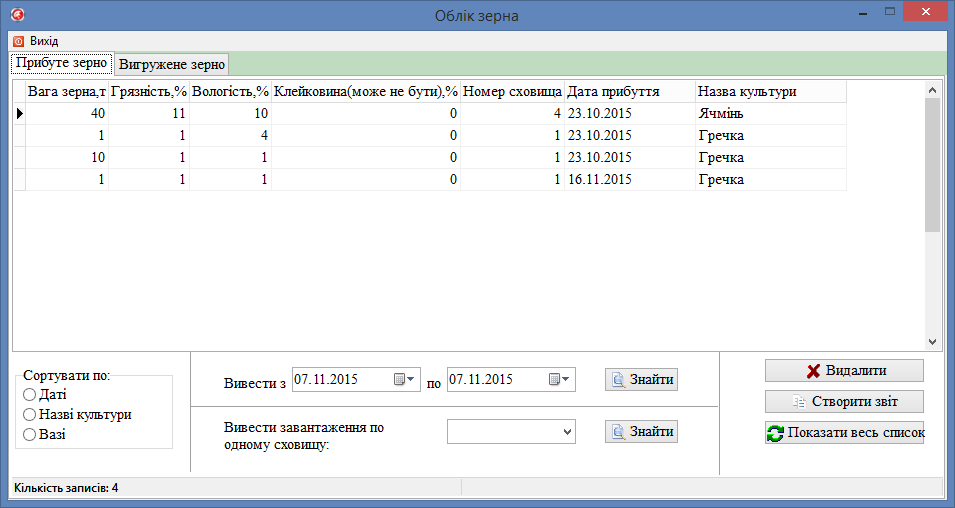


Рисунок 4.11 – Вікно виведення всього прибутого зерна на сховища

На цьому вікні зображена таблиця с інформацією про прибуте зерно на елеватор. Можна таблицю з цими записами сортувати по даті, назві культури, вазі. Можливий пошук по даті прибуття в інтервалі та пошук по одному сховищу,який виведе всі завантаження по одному сховищу. Для того, щоб видалити запис про завантаження елеватора то натисніть кнопку «Видалити». З відсортованими записами та при пошуку в інтервалі часу можливо створити звіт, для цього потрібно викликати форму для створення звітів, яка зображена на рисунку 4.13.

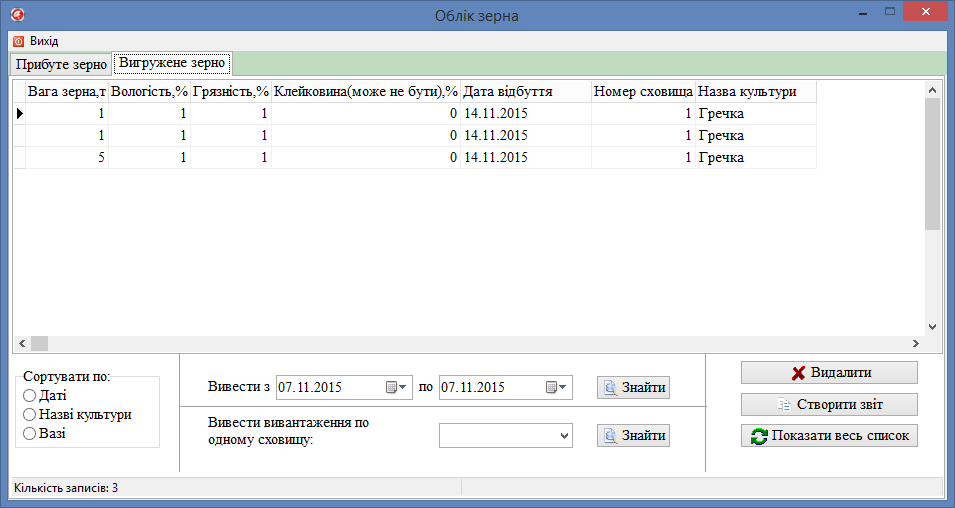


Рисунок 4.12 – Вікно виведення всього вантаженого зерна зі сховища

На цьому вікні зображена таблиця с інформацією про вивантажене зерно з елеватора. Можна таблицю з цими записами сортувати по даті, назві культури, вазі. Можливий пошук по даті прибуття в інтервалі та пошук по одному сховищу,який виведе всі завантаження по одному сховищу. Для того, щоб видалити запис про вивантаження елеватора то натисніть кнопку «Видалити». З відсортованими записами та при пошуку в інтервалі часу можливо створити звіт, для цього потрібно викликати форму для створення звітів.

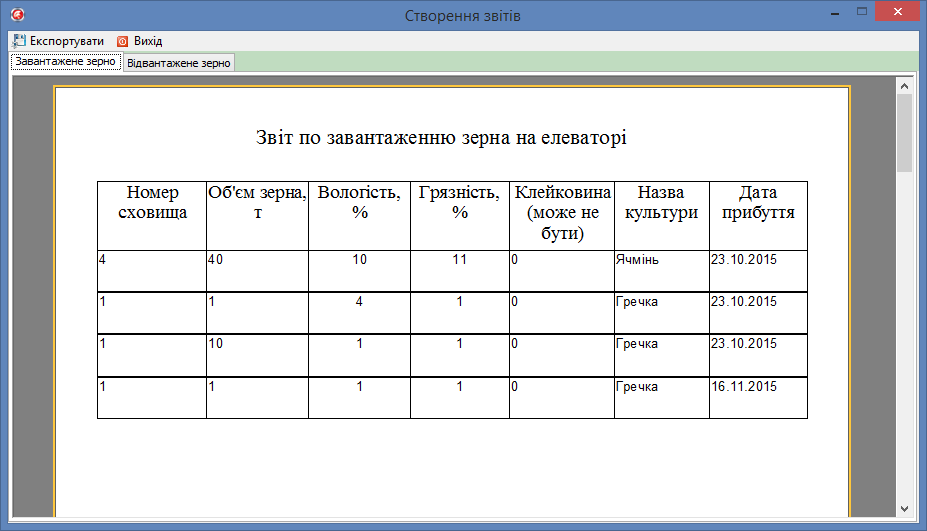


Рисунок 4.13 – Вікно створення звіту по завантаженню елеватора

Форма для звітів зображена на рисунку 4.14.

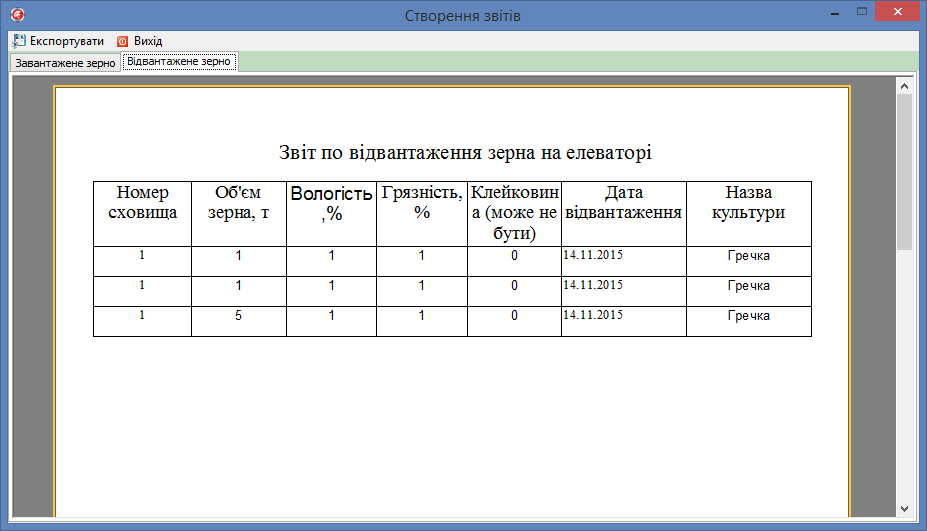


Рисунок 4.14 – Вікно створення звіту по вивантаження елеватора

Створений звіт можна експортувати в Word,PDF та Excel файли.

Для перегляду інформації про програму потрібно обрати пункт меню «Про програму», тоді з’явиться відповідне вікно, яке зображено на рисунку 4.15.

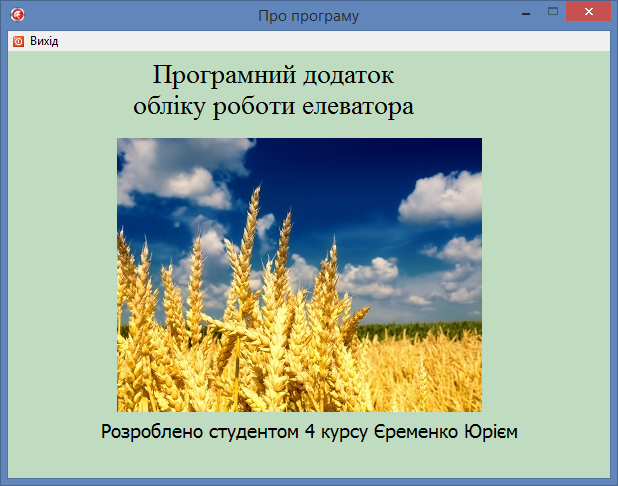


Рисунок 4.15 – Вікно «Про програму»

Для виходу з програми обрати в головному меню пункт «Вихід».

# ВИСНОВКИ

В ході виконання курсового проекту був розроблений програмний додаток, що дозволяє виконувати перегляд бази даних, додавання інформації, редагування та видалення, сортування даних, фільтрування за датою прибуття і відбуття, за сховищами, а також перегляд звіту завантаження та вивантаження елеватора, також експортування звіту у різні формати.

Програмний додаток має простий інтуїтивно-зрозумілий інтерфейс та містить підказки та перевірки коректності введених даних, що дозволяє працювати з ним людям з мінімальними навиками володіння комп’ютером. Отже цей програмний додаток можна використовувати на елеваторах та у малих фермерських господарств.

У подальшому цей проект можна буде вдосконалити додавши клієнт-серверний інтерфейс, об’єднавши всі елеватори підприємств об’єднати в одну базу даних.

# СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Баас Р. Delphi 5: для пользователя[Текст]/ Р. Баас , М.Фервай, Х. Гюнтер; Пер. с нем. - К.: BHVб, 2000 - 496 с.
2. Бобровский С. Delphi 6 и Kylix: Библиотека программиста[Текст]/ С.Бобровский - СПб.: Питер, 2002. - 560 с.
3. Федоров А. Н. Введення в бази даних Частина 2. Настільні СУБД[Текст]/ А.Н., Федоров Н. С. Єлманова: КомпьютерПресс - 2000. - № 3,4.
4. Бобровский С. Delphi 7. Учебный курс[Текст]/ С. Бобровский - СПб.: Питер, 2004. - 735 с.
5. Архангельський, А.Я. Програмування в Delphi 7[Текст] / А.Я.Архангельський. -М: Бином, 2003. - 1152 с.
6. Дарахвелидазе П.Г., Марков Е.П. Программирование в Delphi 7[Текст] / П.Г.Дарахвелидазе., Е.П.Марков - СПб.: БХВ-Петербург, 2003. - 784 с.