

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ОДЕСЬКИЙ ДЕРЖАВНИЙ ЕКОЛОГІЧНИЙ УНІВЕРСИТЕТ

Факультет комп'ютерних наук управ-
ління та адміністрування
Кафедра інформаційних технологій

Бакалаврська кваліфікаційна робота

на тему: Створення застосування для роботи з ГІС «Поводження з
сіськогосподарськими відходами Одеської області»

Виконав студент 4 курсу групи К-42
Напрямок 6.05.01.01 Комп'ютерні науки

Юшанов Денис Олексійович

Керівник ст.викл.

Рольщиков Вадим Борисович

Консультант к.геогр.н., доцент

Кузніченко Світлана Дмитрівна

Рецензент к.геогр.н., доцент

Лужбін Анатолій Михайлович

ЗМІСТ

Перелік скорочень, умовних позначень і термінів	5
Вступ	6
1 Аналітична частина	7
1.1 Аналіз предметної області	7
1.2 Аналіз інструментів розробки	8
1.3 Вибір платформи розробки	8
1.3.1 Опис QT	8
1.3.2 Опис .NET Framework	10
1.4 Вибір СУБД	13
1.4.1 Опис Microsoft SQL Server	13
1.4.2 Опис MySQL	16
1.4.3 Опис PostgreSQL	18
1.5 Вибір IDE	22
1.5.1 Розгляд Qt Creator	22
1.5.2 Розгляд Eclipse	24
1.5.3 Розгляд Microsoft Visual Studio	26
2 Проектна частина	29
2.1 Постановка завдань	29
2.2 Проектування структури БД програмного продукту	29
2.3 Проектування елементів маніпулювання з даними БД	41
Висновки	55
Перелік джерел посилання	56

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ І ТЕРМІНІВ

БД – база даних

ПЗ – програмне забезпечення

ПП – програмний продукт

СУБД – система управління базами даних

ТЗ – технічне завдання

CLR – Common Language Runtime

IDE – Integrated Development Environment, інтегроване середовище розробки

OpenGL – Open Graphics Library, відкрита графічна бібліотека

SVG – Scalable Vector Graphics, масштабована векторна графіка

XML – Extensible Markup Language, розширювана мова розмітки

WYSIWYG – What You See Is What You Get, що бачиш, то і отримаєш

ВСТУП

Актуальність даної теми в першу чергу виходить з причини розвитку сільського господарства Одеської області, яке має тенденцію до постійного зростання. Внаслідок збільшення виробництва сільськогосподарської продукції постає питання щодо зростання кількості відходів, які є результатом як утворення залишків при збиранні врожаю сільськогосподарських культур, так і побічним продуктом при утриманні сільськогосподарських тварин. Утворення великих обсягів відходів рослинного та тваринного походження у підприємствах сільського господарства та у господарствах населення Одеської області становить значну проблему з їх переробки та утилізації. Між тим, використання таких відходів для одержання альтернативного палива методом анаеробного зброджування у біогазових установках є сучасним способом ха-зйнування, насамперед за умови оцінки об'ємів створюваних органічних відходів та планування видобутку біогазу у господарствах районів Одеської області. [1]¹⁾.

Метою даної дипломної роботи є моделювання і розробка системи, за допомогою якої виконується систематизування статистичних даних, проведення розрахунків з утворення відходів сільського господарства по районах Одеської області, упорядкування отриманої інформації, її відображення у наочному вигляді та отримання інформації за результатами обчислення у вигляді графічного контенту.

Представлений дипломний проект містить 57 сторінок, 25 рисунків, 13 посилань, 10 таблиць.

(дата звернення 11.05.2019).

1 АНАЛІТИЧНА ЧАСТИНА

1.1 Аналіз предметної області

Кінцевий програмний продукт – це сукупність програмних засобів, які в поєднанні повинні вміти працювати з статистичними даними отриманими з кожного району Одеської області, які в свою чергу являють собою дані з валового збору, з того чи іншого напрямку сільського господарства.

Кінцевий продукт включатиме в себе місце збереження всіх необхідних статистичних даних, а також програми-настільного застосування, яка дозволить обробляти ці дані в залежності від вимог користувача.

Перелік вимог які повинні виконуватися під час розробки програмного продукту:

- програмний продукт повинен розраховувати кількість відходів, виходячи з показників валового збору кожного напрямку сільського господарства, для кожного з районів Одеської області;
- програмний продукт повинен вміти розраховувати кількість біогазу, ґрунтуючись на отриманих показниках з розрахунків відходів;
- програмний продукт повинен мати БД, необхідну для комфортного користування і маніпулювання статистичними даними по кожному регіону;
- програмний продукт повинен мати графічний інтерфейс користувача, який дозволяв би проводити необхідні операції;

Актуальність цієї роботи полягає в тому, що за відсутності системного підходу до визначення кількісних показників сільськогосподарських відходів у районах Одеської області виникають певні складнощі з виконанням розрахунків потенціального виробництва біогазу з органічних залишків тваринного та рослинного походження, що є досить актуальною проблемою у напрямку економного енергоспоживання та використання альтернативних джерел енергії з метою скорочення об'ємів витрат природного газу а інших видів вичисленого пального.

Необхідність систематизації вихідних та розрахованих даних, отримання результатів розрахунків постає основним чинником щодо створення логічної, яка б задовольняла поставленим вимогам. [1]¹⁾

1.2 Аналіз інструментів розробки

Основними завданнями при розробці програмного продукту є:

- огляд платформи, на якій буде розроблятися програмний продукт;
- СУБД для створення БД і внесення статистичних даних по районам;
- середовище розробки (IDE), у якому безпосередньо буде проводитися розробка програмного продукту.

1.3 Вибір платформи розробки

При виборі платформи були розглянуті такі кросплатформові програмні забезпечення, як QT і .NET.

1.3.1 Опис QT

QT – кросплатформовий інструментарій розробки програмного забезпечення мовою програмування C++. З часу своєї появи в 1996 році бібліотека лягла в основу багатьох програмних проектів. Крім того, Qt є фундаментом популярного робочого середовища KDE, що входить до складу багатьох дистрибутивів Linux. Qt дозволяє запускати написане за його допомогою програмне забезпечення на більшості сучасних операційних систем, просто компілюючи текст програми для кожної операційної системи без зміни сирцевого коду. Містить всі основні класи, які можуть бути потрібні для розробки прикладного програмного забезпечення, починаючи з елементів графічного інтерфейсу й закінчуючи класами для роботи з мережею, базами даних, OpenGL,

(дата звернення 11.05.2019).

SVG і XML. Бібліотека дозволяє керувати нитками, працювати з мережею та забезпечує кросплатформовий доступ до файлів. Є повністю об'єктно-орієнтованим, розширюваним і підтримує техніку компонентного програмування.

Відмітна особливість – використання метаоб'єктного компілятора – попередньої системи обробки вихідного коду. Розширення можливостей забезпечується системою модулів, які можливо розміщувати безпосередньо на панелі візуального редактора. Також існує можливість розширення звичної функціональності віджетів, пов'язаної з розміщенням їх на екрані, відображенням, перемальовуванням при зміні розмірів вікна.

Комплектується візуальним середовищем розробки графічного інтерфейсу Qt Designer, що дозволяє створювати діалоги і форми в режимі WYSIWYG. У постачанні Qt є Qt linguist – графічна утиліта, що дозволяє спростити локалізацію і переклад програми на багато мов; і Qt Assistant – довідкова система Qt, що спрощує роботу з документацією по бібліотеці, а також дозволяє створювати кросплатформову довідку для розроблювального на основі Qt програмного забезпечення. Починаючи з версії 4.5.0 в комплект включено середовище розробки Qt Creator, яке включає редактор коду, довідку, графічні засоби Qt Designer і можливість налагодження застосувань. Qt Creator може використовувати GCC або Microsoft VC++ у якості компілятора і GDB у якості відладчика. Для Windows-версій бібліотека комплектується компілятором, заголовними і об'єктними файлами MinGW.

Існують версії бібліотеки для Microsoft Windows, систем класу UNIX з графічною підсистемою X11, Android, iOS, Mac OS X, Microsoft Windows CE, QNX, вбудовуємих Linux-систем і платформи S60. Йде портування на Windows Phone і Windows RT. Також йде портування на Naiku і Tizen.

Деякий час бібліотека також поширювалася ще в версії Qt/Embedded, призначеної для застосування на вбудовуємих і мобільних пристроях, але починаючи з середини 2000-х років вона виділена в самостійний продукт Qtopia.

Починаючи з версії 4.5 Qt поширюється по трьом ліцензіям:

- Qt Commercial – для розробки програмного забезпечення з власницькою ліцензією, яка припускає модифікацію самої Qt без розкриття змін;
- GNU GPL – для розробки з відкритим кодом, що розповсюджується на умовах GNU GPL, а також для модифікації Qt;
- GNU LGPL – для розробки програмного забезпечення з власницькою ліцензією.

Вихідний код, єдиний для усіх варіантів ліцензій, знаходиться у вільному доступі в Git-сховищі, розташованому на Github. Крім самого вихідного коду Qt, там же розташовані сховища супутніх бібліотек, що розробляються авторами бібліотеки і співтовариством.

До версії 4.0.0 під вільною ліцензією поширювалися лише Qt/Mac, Qt/X11, Qt/Embedded, але, починаючи з 4.0.0 (випущеної в кінці червня 2005), Qt Software “звільнили” і Qt/Windows. При цьому існували сторонні вільні версії Qt/Windows раніше 4.0.0, зроблені на основі Qt/X11.[2]¹⁾

1.3.2 Опис .NET Framework

Платформа .NET Framework – програмна технологія, запропонована фірмою Microsoft як платформа для створення як звичайних програм, так і веб-застосунків. Багато в чому є продовженням ідей та принципів, покладених в технологію Java. Однією з ідей .NET є сумісність служб, написаних різними мовами. Хоча ця можливість рекламується Microsoft як перевага .NET, платформа Java має таку саму можливість.

Кожна бібліотека (збірка) в .NET має свідчення про свою версію, що дозволяє усунути можливі конфлікти між різними версіями збірок.

¹⁾ [2] Qt – Википедія. URL: <https://ru.wikipedia.org/wiki/Qt> (дата звернення 12.05.2019).

.NET — кросплатформова технологія, в цей час існує реалізація для платформи Microsoft Windows, FreeBSD (від Microsoft) і варіант технології для ОС Linux в проєкті Mono (в рамках угоди між Microsoft з Novell), DotGNU.

Захист авторських прав відноситься до створення середовищ виконання (CLR — Common Language Runtime) для програм .NET. Компілятори для .NET випускаються багатьма фірмами для різних мов вільно.

.NET поділяється на дві основні частини — середовище виконання (по суті віртуальна машина) та інструментарій розробки.

Середовища розробки .NET-програм: Visual Studio .NET (C++, C#, J#), SharpDevelop, Borland Developer Studio (Delphi, C#) тощо. Середовище Eclipse має додаток для розробки .NET-програм. Застосовні програми також можна розроблювати в текстовому редакторі та використовувати консольний компілятор.

Як і технологія Java, середовище розробки .NET створює байт-код, призначений для виконання віртуальною машиною. Вхідна мова цієї машини в .NET називається CIL (Common Intermediate Language), також відома як MSIL (Microsoft Intermediate Language), або просто IL. Застосування байт-коду дозволяє отримати кросплатформність на рівні скомпільованого проєкту (в термінах .NET: збірка), а не на рівні сирцевого тексту, як, наприклад, в C. Перед запуском збірки в середовищі виконання (CLR) байт-код перетворюється вбудованим в середовище JIT-компілятором (just in time, компіляція на льоту) в машинні коди цільового процесора.

Слід зазначити, що один з перших JIT-компіляторів для Java був також розроблений фірмою Microsoft (тепер в Java використовується досконаліша багаторівнева компіляція — Sun HotSpot). Сучасна технологія динамічної компіляції дозволяє досягнути аналогічного рівня швидкодії з традиційними «статичними» компіляторами (наприклад, C++) і питання швидкодії часто залежить від якості того чи іншого компілятора.

Основною ідеєю при розробці .NET Framework було забезпечення свободи розробника за рахунок надання йому можливості створювати додатки різних типів, здатні виконуватися на різних типах пристроїв і в різних середовищах.

Другим принципом стала орієнтація на системи, що працюють під управлінням сімейства операційних систем Microsoft Windows[3]¹⁾.

Зазвичай застосування, розроблені на основі конкретної версії платформи .NET Framework, можуть виконуватися без доробок і на більш пізніх версіях платформи.

Якщо ви не розробляєте застосування .NET Framework, але використовуєте їх, вам не потрібно володіти спеціальними знаннями про платформу .NET Framework або її роботі. У більшості випадків платформа .NET Framework абсолютно прозора для користувачів.

Платформа .NET Framework допомагає вирішувати конфлікти версій, коли на комп'ютері встановлені декілька версій середовища CLR. Це означає, що різні версії застосувань можуть співіснувати і що застосування може виконуватися на версії платформи .NET Framework, для якої воно було створено. Паралельне виконання застосовується до груп версій .NET Framework 1.0 / 1.1, 2.0 / 3.0 / 3.5 і 4 / 4.5.x / 4.6.x / 4.7.x / 4.8.

Під час налаштування відповідно до стандарту .NET розробники створюють бібліотеки класів, які працюють на різних платформах .NET Framework, підтримуваних відповідної версією стандарту. Наприклад, бібліотеки, розроблені відповідно до стандарту .NET 2.0, можуть використовуватися додатками, орієнтованими на платформи .NET Framework 4.6.1, .NET Core 2.0 і UWP 10.0.16299. [4]²⁾.

Врешті-решт, було все ж таки прийнято рішення, розробляти програмний продукт використовуючи платформу .NET Framework і мову C#. Вибір

¹⁾ [3] .NET Framework – Википедия. URL: https://ru.wikipedia.org/wiki/.NET_Framework (дата звернення 13.05.2019).

²⁾ [4] Начало работы с .NET Framework | Microsoft Docs. URL: <https://docs.microsoft.com/ru-ru/dotnet/framework/get-started/> (дата звернення 13.05.2019).

платформи обґрунтовано тим, що .NET Framework надає більшу кількість служб, які йдуть одразу з платформою, і розробку можливо продовжувати на різних версіях платформи, без встановлення додаткових програмних застосунків, які необхідно встановлювати разом із Qt для отримання еквівалентного результату.

1.4 Вибір СУБД

1.4.1 Опис Microsoft SQL Server

Microsoft SQL Server – комерційна система керування базами даних, що розповсюджується корпорацією Microsoft. Мова, що використовується для запитів – Transact-SQL, створена спільно Microsoft та Sybase. Transact-SQL є реалізацією стандарту ANSI / ISO щодо структурованої мови запитів SQL із розширеннями. Використовується як для невеликих і середніх за розміром баз даних, так і для великих баз даних масштабу підприємства. Багато років вдало конкурує з іншими системами керування базами даних.

Microsoft робить SQL Server доступним у різноманітних варіантах, які різняться наборами властивостей в залежності від цілей кінцевого користувача. Це такі редакції як:

- SQL Server Compact Edition (SQL CE) – компактне видання – вкладений механізм бази даних. Завдяки малому обсягу (2 Мб для DLL) має зменшені властивості у порівнянні з іншими варіантами. Розмір бази даних обмежений 4 Гб і не може використовуватися як служба Windows;

- SQL Server Express Edition – Раніше відомий під назвою MSDE (Microsoft SQL Server Desktop Engine), Microsoft SQL Server Express є вільно поширюваною версією SQL Server. Дана версія має деякі технічні обмеження, також відсутні графічні інструменти адміністрування. Такі обмеження роблять її непридатною для розгортання великих баз даних. В основному вона використовується у застосунках, при проектуванні, або для

самостійного вивчення. Розмір бази даних обмежений 4 Гб, розмір пам'яті, що може бути адресованою – 1 Гб, підтримує лише один процесор;

- SQL Server Workgroup Edition – SQL Server Workgroup Edition включає функціональність ядра бази даних, але не включає додаткові сервіси;

- SQL Server Standard Edition – SQL Server Standard edition включає механізми ядра бази даних, а також автономні сервіси. Відрізняється від варіанту Enterprise Edition тим, що підтримує менше активних вузлів та не включає деякі функції збільшення продуктивності;

- SQL Server Enterprise Edition – SQL Server Enterprise Edition – це повнофункціональна версія SQL Server;

- SQL Server Developer Edition – SQL Server Developer Edition включає ті самі функції, що й SQL Server Enterprise Edition, але містить обмеження щодо використання його лише для розробки та тестування. Його ліцензія не дозволяє використання як виробничого сервера.

Microsoft SQL Server як мову запитів використовує версію SQL, що отримала назву Transact-SQL (скорочено T-SQL), яка є реалізацією SQL-92 (стандарт ISO для SQL) з багатьма розширеннями. T-SQL дозволяє використовувати додатковий синтаксис процедур, що зберігаються і забезпечує підтримку транзакцій (взаємодія бази даних з керуючим застосунком). Microsoft SQL Server та Sybase ASE для взаємодії з мережею використовують протокол рівня застосунка під назвою Tabular Data Stream (TDS, протокол передачі табличних даних).

Microsoft SQL Server також підтримує Open Database Connectivity (ODBC) – інтерфейс взаємодії застосунків з СУБД. Версія SQL Server 2005 надає можливість підключення користувачів через веб-сервер-сервіси, що використовують протокол SOAP. Це дозволяє клієнтським програмам, не призначеним для Windows, кроссплатформенно з'єднуватися з SQL Server. Microsoft також випустила сертифікований драйвер JDBC, що дозволяє

застосункам під керування Java (таким як BEA і IBM Websphere) з'єднуватися з Microsoft SQL Server 2000 і 2005.

SQL Server підтримує дзеркалювання та кластеризацію баз даних. Кластер серверу SQL – це сукупність однаково конфігурованих серверів; така схема допомагає розподілити робоче навантаження між декількома серверами. Усі сервери мають одне віртуальне ім'я, а дані розподіляються за IP-адресами машин кластеру протягом робочого циклу. Також у разі відмови або збою на одному з серверів кластеру доступне автоматичне перенесення навантаження на інший сервер.

SQL Server підтримує надлишкове дублювання даних за трьома сценаріями:

- знімок – виконується «знімок» бази даних, який сервер відправляє одержувачам;
- історія змін – всі зміни бази даних безперервно передаються користувачам;
- синхронізація з іншими серверами – бази даних декількох серверів синхронізуються між собою. Зміни усіх баз даних відбуваються незалежно на кожному сервері, а під час синхронізації відбувається звірка даних. Дублювання такого типу передбачає можливість вирішення протиріч між базами даних.

SQL Server 2005 має вбудовану підтримку .NET Framework. Завдяки цьому, процедури бази даних, що зберігаються, можуть бути написані на будь-якій мові платформи .NET з використанням повного набору бібліотек, доступних для .NET Framework. На відміну від інших процесів, .NET Framework виділяє додаткову пам'ять і будує засоби керування SQL Server, не використовуючи вбудовані засоби Windows. Це підвищує продуктивність порівняно із загальними алгоритмами Windows, оскільки алгоритми розподілу ресурсів спеціально налагоджені для використання у структурах SQL Server.

Microsoft та інші компанії пропонують велику кількість програмних засобів розробки, які дозволяють розробляти застосунки для бізнесу з використанням баз даних Microsoft SQL Server. Microsoft SQL Server 2005 включає також Common Language Runtime (CLR) Microsoft .NET, що дозволяє застосункам, розробленим на мовах платформи .NET (наприклад, VB.NET або C#), реалізовувати процедури, що зберігаються та різні функції. Попередні версії засобів розробки Microsoft використовували лише API для надання функціонального доступу до Microsoft SQL Server[14]¹⁾.

1.4.2 Опис MySQL

MySQL — вільна система керування реляційними базами даних. MySQL був розроблений компанією «ТсХ» для підвищення швидкодії обробки великих баз даних. Ця система керування базами даних (СКБД) з відкритим кодом була створена як альтернатива комерційним системам. MySQL з самого початку була дуже схожою на mSQL, проте з часом вона все розширювалася і зараз MySQL — одна з найпоширеніших систем керування базами даних. Вона використовується, в першу чергу, для створення динамічних веб-сторінок, оскільки має чудову підтримку з боку різноманітних мов програмування.

MySQL має подвійне ліцензування. MySQL може розповсюджуватися відповідно до умов ліцензії GPL. Але за умовами GPL, якщо якась програма використовує бібліотеки MySQL, то вона теж повинна розповсюджуватися за ліцензією GPL. Проте це може розходитися з планами розробників, які не бажають відкривати сирцеві тексти своїх програм. Для таких випадків передбачена комерційна ліцензія компанії Oracle, яка також забезпечує якісну сервісну підтримку. В разі використання та розповсюдження програмного забезпечення з іншими вільними ліцензіями, такими як BSD,

¹⁾ [5] Microsoft SQL Server – Вікіпедія. URL: https://uk.wikipedia.org/wiki/Microsoft_SQL_Server (дата звернення 14.05.2019)

Apache, MIT та інші, MySQL дозволяє використання бібліотек MySQL за ліцензією GPL.

MySQL виникла як спроба застосувати mSQL до власних розробок компанії: таблиць, для яких використовувалися ISAM — підпрограми низького рівня для індексного доступу до даних. У результаті був вироблений новий SQL-інтерфейс, але API-інтерфейс залишився в спадок від mSQL. Звідки походить назва «MySQL» — достеменно не відомо. Розробники дають два варіанти: або тому, що практично всі напрацювання компанії починалися з префікса My, або на честь дівчинки на ім'я My, дочки Майкла Монті Віденіуса, одного з розробників системи.

Логотип MySQL у вигляді дельфіна носить ім'я «Sakila». Він був обраний з великого списку запропонованих користувачами «імен дельфіна». Ім'я «Sakila» було відправлено Open Source-розробником Ambrose Twebaze.

В січні-лютому 2008 Sun Microsystems придбала розробника системи керування базами даних MySQL за \$1 млрд. Після поглинання у 2009 році Sun Microsystems компанією Oracle Corporation MySQL стала власністю Oracle.

За час розвитку під орудою Oracle дедалі більше відокремлює MySQL від спільноти і робить процес розробки все менш прозорим. Наприклад, повернута практика поставки власницьких розширених функцій в Enterprise-версії MySQL, спостерігається приховування інформації про вразливості, зі складу виключений тестовий набір, закритий доступ до більшої частини системи відстеження помилок та припинено публікація згрупованого логу змін, що дозволяє судити про прив'язку патчів до конкретних змін.

MySQL вважається гарним рішенням для малих і середніх застосувань. Сирцеві коди сервера компілюються на багатьох платформах. Найповніше можливості сервера виявляються в UNIX-системах, де є підтримка багатопотоковості, що підвищує продуктивність системи в цілому.

MySQL — компактний багатопотоковий сервер баз даних. Характеризується високою швидкістю, стійкістю і простотою використання.

Можливості сервера MySQL:

- простота у встановленні та використанні;
- кількість рядків у таблицях може досягати 50 млн;
- висока швидкість виконання команд;
- наявність простої і ефективної системи безпеки;
- підтримується необмежена кількість користувачів, що одночасно працюють із БД;

MySQL версії 5.1 має 20 відомих серйозних помилок в додаток до 35 дефектів версії 5.0. Критичні помилки іноді не виправляються протягом тривалих періодів часу. Одним із прикладів є критична помилка, що відома з 2003 року.

MySQL показує низьку ефективність при використанні її як сховища даних, це частково пов'язано з нездатністю використовувати декілька процесорів для обробки одного запиту. До того ж, MySQL часто критикують за те, що ця СКБД має розходження зі стандартом SQL щодо трактування NULL значень і значення за замовчуванням[6]¹⁾.

1.4.3 Опис PostgreSQL

PostgreSQL – об'єктно-реляційна система керування базами даних (СКБД). Є альтернативою як комерційним СКБД (Oracle Database, Microsoft SQL Server, IBM DB2 та інші), так і СКБД з відкритим кодом (MySQL, Firebird, SQLite).

Порівняно з іншими проектами з відкритим кодом, такими як Apache, FreeBSD або MySQL, PostgreSQL не контролюється якоюсь однією компанією, її розробка можлива завдяки співпраці багатьох людей та компаній, які хочуть використовувати цю СКБД та впроваджувати у неї найновіші досягнення.

¹⁾ [6] MySql – Вікіпедія. URL: <https://uk.wikipedia.org/wiki/MySQL> (дата звернення 14.05.2019).

Сервер PostgreSQL написаний на мові C. Зазвичай розповсюджується у вигляді набору текстових файлів із сирцевим кодом. Для інсталяції необхідно відкомпілювати файли на своєму комп'ютері і скопіювати в деякий каталог. Весь процес детально описаний в документації.

PostgreSQL — широко розповсюджена система керування базами даних з відкритим сирцевим кодом. Прототип був розроблений в Каліфорнійському університеті Берклі в 1987 році під назвою POSTGRES, після чого активно розвивався і доповнювався. В червні 1990 року з'явилась друга версія із переробленою системою правил маніпулювання та роботи з таблицями, у 1991 році — третя версія, із доданою підтримкою одночасної роботи кількох менеджерів збереження, покращеним механізмом запитів і доповненою системою внутрішніх правил. В цей час POSTGRES використовувався для реалізації великих систем, таких як: система аналізу фінансових даних, пакет моніторингу функціональності потоків, база даних відстеження астероїдів, система медичної інформації, кілька географічних систем. POSTGRES також використовувався як навчальний інструмент в кількох університетах. 1992 року POSTGRES став головною СКБД наукового комп'ютерного проекту Sequoia 2000. 1993 року кількість користувачів подвоїлась. Стало зрозуміло, що для підтримки й подальшого розвитку необхідні великі витрати часу на дослідження баз даних, тому офіційно проект Берклі було зупинено на версії 4.2. 1994 року Andrew Yu і Jolly Chen додали інтерпретатор мови SQL, вдосконалили сирцевий код і виклали в Інтернеті свою реалізацію під назвою Postgres95. 1996 року програмний продукт було перейменовано на PostgreSQL із початковою версією 6.0. Подальшою підтримкою й розробкою займається група спеціалістів у галузі баз даних, які добровільно приєдналися до цього проекту.

PostgreSQL має функції, які дозволяють виконувати деякий код безпосередньо сервером бази даних. Ці функції можуть бути написані на SQL, який має деякі примітивні програмні оператори, такі як галуження та

цикли. Але гнучкішою буде функція написана на одній із мов програмування, з якими PostgreSQL може працювати. До таких мов належать:

- вбудована мова, яка зветься PL/pgSQL, подібна до процедурної мови PL/SQL компанії Oracle;

- мови розробки сценаріїв: PL/Perl, PL/Python, PL/Tcl, PL/Ruby, PL/sh;

- класичні мови програмування C, C++, Java (за допомогою PL/Java).

Функції можуть виконуватись із привілеями користувача, який її викликав, або із привілеями користувача, який її написав.

У PostgreSQL є підтримка індексів наступних типів: B-дерево, хеш, R-дерево, GiST, GIN. При необхідності можна створити нові типи індексів.

PostgreSQL підтримує одночасну модифікацію БД декількома користувачами за допомогою механізму Multiversion Concurrency Control (MVCC). Завдяки цьому виконуються вимоги ACID, і практично відпадає потреба в блокуванні зчитування.

PostgreSQL підтримує великий набір вбудованих типів даних:

- числові типи;

- цілі;

- з фіксованою крапкою;

- з нефіксованою крапкою;

- грошовий тип;

- символні типи довільної довжини;

- двійкові типи (включаючи BLOB);

- типи «дата/час»;

- булевий тип;

- перерахування;

- геометричні примітиви;

- мережеві типи;

- UUID-ідентификатор;

- XML-дані;

- JSON-дані;

- масиви;
- OID-типи;
- псевдотипи.

Крім того, користувач може самостійно створювати нові необхідні йому типи та програмувати для них механізми індексування за допомогою GiST.

PostgreSQL може бути розширено користувачем для власних потреб практично в будь-якому аспекті. Є можливість додавати власні:

- перетворення типів;
- типи даних;
- домени (для користувача типи з самого початку з накладеними обмеженнями);
- функції (включаючи агрегатні);
- індекси;
- оператори (включаючи перевизначення вже існуючих);
- процедурні мови.

В PostgreSQL таблиці можуть успадковувати характеристики та набори полів від інших таблиць (батьківських). При цьому дані, які додаються до породженої таблиці, автоматично будуть брати участь (якщо це не вказано окремо) в запитах до батьківської таблиці. Цей функціонал в поточний час не є повністю завершеним. Однак він достатній для практичного використання.

В PostgreSQL тригери визначаються як функції, що ініціюються DML-операціями. Наприклад, операція INSERT може запускати тригер, що перевіряє доданий запис на відповідність певним умовам. Тригери можна писати різними мовами програмування. Вони пов'язані з визначеною таблицею. Множинні тригери виконуються в алфавітному порядку.

Згідно з результатами автоматизованого дослідження різного ПЗ на предмет помилок, у вихідному коді PostgreSQL було знайдено 20 проблемних місць на 775 000 рядків вихідного коду (в середньому, одна помилка на 39 000 рядків коду).

Для порівняння:

- MySQL – 97 проблем, одна помилка на 4 000 рядків коду;
- FreeBSD (цілком) – 306 проблем, одна помилка на 4 000 рядків коду;
- Linux (тільки ядро) – 950 проблем, одна помилка на 10 000 рядків коду[7]¹⁾.

Після порівняння декількох СУБД було зроблено висновок, що для розробки нашого застосування краще за все підійде PostgreSQL ніж MS SQL Server, тому що база PostgreSQL має дуже велику перевагу, завдяки своїм модулям, наприклад завдяки модулю PostGIS, який можливо встановити в нашу базу даних і доповнювати її не тільки статистичними даними, а й створювати елементи, які будуть прив'язані до фізичних величин (географічних координат), що відкриває для нас у майбутньому дуже великі перспективи, адже на підставі цих даних, можливо прив'язати статистичні дані до географічних величин.

1.5 Вибір IDE

1.5.1 Розгляд Qt Creator

Qt Creator — інтегроване середовище розробки, призначене для створення крос-платформових застосунків з використанням бібліотеки Qt. Підтримується розробка як класичних програм мовою C++, так і використання мови QML, для визначення сценаріїв, в якій використовується JavaScript, а структура і параметри елементів інтерфейсу задаються CSS-подібними блоками. Qt Creator може використовувати GCC або Microsoft VC++ як компілятор і GDB як зневаджувач. Для Windows версій бібліотека комплектується компілятором, заголовними і об'єктними файлами MinGW. Сирцеві тексти Qt Creator поширюються в рамках ліцензії LGPL.

¹⁾ [7] PostgreSQL – Вікіпедія. URL: <https://uk.wikipedia.org/wiki/PostgreSQL> (дата звернення 14.05.2019).

Qt Creator розроблена норвезькою компанією Trolltech, яку у 2008 році поглинула Nokia. Анонс проекту відбувся на Qt Developer Days в жовтні 2008 року. Публічна бета-версія проекту була опублікована 30 жовтня 2008. Фінальний реліз відбувся 3 березня 2009 року (разом з виходом Qt 4.5).

Після укладення стратегічного союзу з Microsoft Nokia втратила інтерес до розвитку технологій Qt. У березні 2011 фінська компанія Digia, постачальник ERP-систем, послуг і рішень в області мобільних систем і користувацьких інтерфейсів, оголосила про укладення угоди з Nokia про викуп у тої прав на комерційне ліцензування та надання послуг з підтримки розробки з використанням бібліотеки Qt.[5] У вересні 2012 Nokia повністю відмовилася від Qt і Digia купує у Nokia весь бізнес і програмні технології, пов'язані з Qt.

До особливостей Qt Creator можна віднести:

- зроблений спеціально для розробки на Qt;
- вбудований редактор форм (Qt Designer) і довідкова система (Qt Assistant);
- контекстно-залежна система допомоги;
- розширюваність плагінами;
- є графічний фронтенд для GDB;
- підтримка зневадження за допомогою CDB;
- для створення проектів використовується qmake (планується підтримка Makefile та тестується підтримка CMake);
- узагальнене підсвічування синтаксису, підтримується велика кількість мов програмування і розмітки. Є можливість створення своїх стилів підсвічування;
- можливість редагувати етапи складання проекту;
- підтримка розробки на мовах C/C++, JavaScript, QML;
- QML-дизайнер;
- можливість розробки під Symbian і Maemo зі зневадженням в симуляторі або на пристрої

Якщо подивитись на архітектуру Qt Creator, функція main по суті своїй – виклик завантажувача плагінів, які й реалізують всю функціональність. Тому всі компоненти QtCreator так чи інакше пов'язані з плагінами [15]¹⁾.

1.5.2 Розгляд Eclipse

Eclipse – вільне модульне інтегроване середовище розробки програмного забезпечення. Розробляється і підтримується Eclipse Foundation і включає проекти, такі як платформа Eclipse, набір інструментів для програмістів на мові Java, системи контролю версій, конструктори GUI тощо. Написаний в основному на Java, може бути використаний для розробки застосунків на Java і, за допомогою різних плагінів, на інших мовах програмування, включаючи Ada, C, C++, COBOL, Fortran, Perl, PHP, Python, R, Ruby, Scala, Clojure та Scheme. Середовища розробки зокрема включають Eclipse ADT (Ada Development Toolkit) для Ada, Eclipse CDT для C/C++, Eclipse JDT для Java, Eclipse PDT для PHP.

Початок коду йде від IBM VisualAge, він був розрахований на розробників Java, складаючи Java Development Tools (JDT). Але користувачі могли розширяти можливості, встановлюючи написані для програмного каркасу Eclipse плагіни, такі як інструменти розробки під інші мови програмування, і могли писати і вносити свої власні плагіни і модулі.

Випущена на умовах Eclipse Public License, Eclipse є вільним програмним забезпеченням. Він став одним з перших IDE під GNU Classpath і без проблем працює під IcedTea.

Eclipse це фреймворк для розробки модульних платформонезалежних застосунків із низкою особливостей:

- можливість розробки ПЗ на багатьох мовах програмування (рідною є Java);

¹⁾ [8] Qt Creator - Вікіпедія. URL: https://uk.wikipedia.org/wiki/Qt_Creator (дата звернення 15.05.2019).

- платформонезалежна;
- модульна, призначена для подальшого розширення незалежним розробниками;
- з відкритим сирцевим кодом;
- розробляється і підтримується фондом Eclipse, куди входять такі постачальники ПЗ, як IBM, Oracle, Borland.

Спочатку проект розроблявся в IBM як корпоративний стандарт IDE для розробки на багатьох мовах під платформи IBM. Потім проект було перейменовано на Eclipse і надано для подальшого розвитку спільноті.

Eclipse насамперед повноцінна Java IDE, націлена на групову розробку, має засоби роботи з системами контролю версій (підтримка CVS входить у поставку Eclipse, активно розвиваються кілька варіантів SVN модулів, існує підтримка VSS та інших). З огляду на безкоштовність, у багатьох організаціях Eclipse – корпоративний стандарт для розробки ПЗ на Java.

Друге призначення Eclipse – служити платформою для нових розширень. Такими стали C/C++ Development Tools (CDT), розроблювані інженерами QNX разом із IBM, засоби для підтримки інших мов різних розробників. Безліч розширень доповнює Eclipse менеджерами для роботи з базами даних, серверами застосунків та інших.

З версії 3.0 Eclipse став не монолітною IDE, яка підтримує розширення, а набором розширень. У основі лежать фреймворки OSGi, і SWT/JFace, на основі яких розроблений наступний шар – платформа і засоби розробки повноцінних клієнтських застосунків RCP (Rich Client Platform). Платформа RCP є базою для розробки різних RCP програм як торент-клієнт Azareus чи File Arranger.

Eclipse написана на Java, тому є платформонезалежним продуктом, крім бібліотеки графічного інтерфейсу SWT, яка розробляється окремо для більшості поширених платформ. Бібліотека SWT використовує графічні засоби платформи (ОС), що забезпечує швидкість і звичний зовнішній вигляд інтерфейсу користувача.

Відповідно до IDC, із Eclipse працюють 2,3 мільйона розробників[9]¹⁾.

1.5.3 Розгляд Microsoft Visual Studio

Microsoft Visual Studio - лінійка продуктів компанії Microsoft, що включають інтегроване середовище розробки програмного забезпечення і ряд інших інструментальних засобів. Дані продукти дозволяють розробляти як консольні додатки, так і додатки з графічним інтерфейсом, в тому числі з підтримкою технології Windows Forms, а також веб-сайти, веб-додатки, веб-служби як в рідному, так і в керованому кодах для всіх платформ, підтримуваних Windows, Windows Mobile, Windows CE, .NET Framework, Xbox, Windows Phone .NET Compact Framework і Silverlight. Visual Studio використовує платформи розробки програмного забезпечення Microsoft, такі як Windows API, Windows Forms, Windows Presentation Foundation, Магазин Windows і Microsoft Silverlight. Він може виробляти як власний код, так і керований код.

Visual Studio включає в себе редактор вихідного коду з підтримкою технології IntelliSense і можливістю найпростішого рефакторінга коду. Вбудований відладчик може працювати як відладчик рівня вихідного коду, так і відладчик машинного рівня. Решта вбудовуються інструменти включають в себе редактор форм для спрощення створення графічного інтерфейсу додатку, веб-редактор, дизайнер класів і дизайнер схеми бази даних. Visual Studio дозволяє створювати і підключати сторонні додатки (плагіни) для розширення функціональності практично на кожному рівні, включаючи додавання підтримки систем контролю версій вихідного коду (як, наприклад, Subversion і Visual SourceSafe), додавання нових наборів інструментів (наприклад, для редагування і візуального проектування коду на предметно-орієнтованих мовах програмування) або інструментів для інших

¹⁾ [9] Eclipse – Вікіпедія. URL: <https://uk.wikipedia.org/wiki/Eclipse> (дата звернення 15.05.2019).

аспектів процесу розробки програмного забезпечення (наприклад, клієнт Team Explorer для роботи з Team Foundation Server)[10]¹⁾.

Дизайнер форм Visual Studio незамінний при розробці програм з графічним інтерфейсом, допомагаючи спроектувати зовнішній вигляд майбутнього програми і роботу кожного елемента інтерфейсу.

Нарешті, Visual Studio надає комплекс інструментів для автоматизації тестування додатків в частині перевірки роботи інтерфейсів, модульного і навантажувального тестування.

Для командних проектів Visual Studio пропонує підтримку групової роботи, дозволяючи виконувати спільне редагування і налагодження будь-якій частині коду в реальному часі, а в якості системи управління версіями використовувати Team Foundation або Git[11]²⁾.

Visual Studio підтримує 36 різних мов програмування і дозволяє редактору коду і відладчику підтримувати (в різному ступені) практично будь-яку мову програмування, за умови наявності спеціальної мовної служби. Вбудовані мови включають C, C++, C++/CLI, Visual Basic .NET, C#, F#, JavaScript, TypeScript, XML, XSLT, HTML і CSS. Підтримка інших мов, таких як Python, Ruby, Node.js і M серед інших, доступна через плагіни. Java (і J#) підтримувалися в минулому.

Найбільш базове видання Visual Studio, видання Community, доступне безкоштовно. Гаслом для Visual Studio Community edition є "Безкоштовна, повнофункціональна IDE для студентів, відкритих джерел та окремих розробників".

В даний час підтримується версія Visual Studio – 2019[12]³⁾.

¹⁾ [10] Microsoft Visual Studio – Википедия. URL: https://ru.wikipedia.org/wiki/Microsoft_Visual_Studio (дата звернення 15.05.2019).

²⁾ [11] Microsoft Visual Studio – описание, ссылка для скачивания, расширения файлов. URL: <https://open-file.ru/programs/microsoft-visual-studio> (дата звернення 15.05.2019).

³⁾ [12] Microsoft Visual Studio – Wikipedia. URL: https://en.wikipedia.org/wiki/Microsoft_Visual_Studio (дата звернення 15.05.2019).

Після розгляду і порівняння декількох IDE, для розробки програмного застосування було обрано середовище Microsoft Visual Studio, адже вона повністю задовольняє нашим потребам у реалізації нашого програмного продукту, а також надає увесь необхідний інструментарій під час розробки, без встановлення додаткових модулів, а так же являє собою кросплатформове середовище розробки.

2 ПРОЕКТНА ЧАСТИНА

2.1 Постановка завдань

Після огляду і аналізу предметної області та вимог до ПП, було виділено ряд завдань, які необхідно вирішити під час створення застосування:

- спроектувати і розробити БД програмного продукту;
- спроектувати і розробити структуру програмного продукту;
- спроектувати і розробити графічний інтерфейс користувача;

2.2 Проектування структури БД програмного продукту

Бази даних - це сукупність відомостей (про реальні об'єкти, процеси, події чи явища), що відносяться до певної теми або завдання, організована таким чином, щоб забезпечити зручне представлення цієї сукупності як в цілому, так і її частини. Реляційна база даних являє собою безліч взаємопов'язаних таблиць, кожна з яких містить інформацію про об'єкти певного типу. Кожен рядок таблиці включає дані про один об'єкт (наприклад, клієнта, автомобілі, документи), а стовпці таблиці містять різні характеристики цих об'єктів - атрибути (наприклад, найменування та адреси клієнтів, марки і ціни автомобілів). Рядки таблиці називаються записами; всі записи мають однакову структуру - вони складаються з полів, в яких зберігаються атрибути об'єкта. Кожне поле запису містить одну характеристику об'єкта і має строго певний тип даних (наприклад, текстовий рядок, число, дата). Всі записи мають одні і ті ж поля, тільки в них містяться різні значення атрибутів. В БД, яку ми проектуємо, можна виділити сутності, які наведені в таблицях з 2.1-2.10. Зв'язки сутностей показані на рис. 2.1.

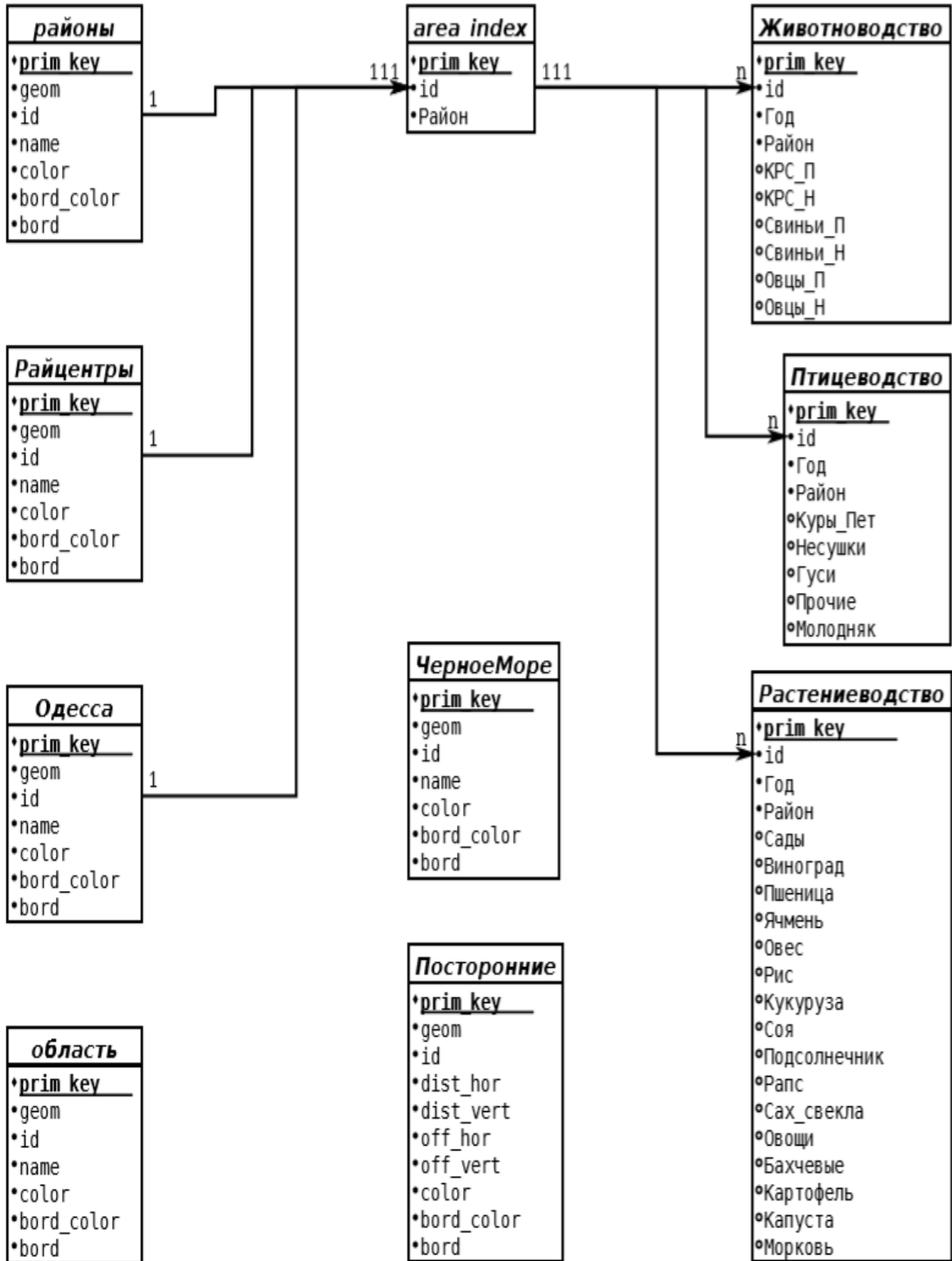


Рисунок 2.1 – Схема БД застосування

Структура таблиць бази даних postgis приведена в нижче перерахованих таблицях:

Розглянемо структуру таблиць бази даних postgis детальніше.

Дані про райони зберігаються у сутності “районы”. Опис даної сутності приведено в таблиці 2.1.

Таблиця 2.1 – Опис сутності “районы”

Атрибут	Тип даних	Опис
prim_key	int	Первинний ключ
geom	geometry	Значення об'єкту в евклідовій системі координат
id	int	Ідентифікатор району
name	varchar(100)	Назва району
color	varchar(30)	Колір району
bord_color	varchar(30)	Колір кордону
bord	double	Ширина кордону

Дані про райцентри зберігаються в сутності “Райцентри”. Опис даної сутності приведено в таблиці 2.2.

Таблиця 2.2 – Опис сутності “Райцентри”

Атрибут	Тип даних	Опис
prim_key	int	Первинний ключ
geom	geometry	Значення об’єкту в евклідовій системі координат
id	int	Ідентифікатор райцентра
name	varchar(100)	Назва райцентра
color	varchar(30)	Колір райцентра
bord_color	varchar(30)	Колір кордону
bord	double	Ширина кордону

Дані про місто Одеса зберігаються в сутності “Одесса”. Опис даної сутності приведено в таблиці 2.3.

Таблиця 2.3 – Опис сутності “Одесса”

Атрибут	Тип даних	Опис
prim_key	int	Первинний ключ
geom	geometry	Значення об’єкту в евклідовій системі координат
id	int	Ідентифікатор міста
name	varchar(30)	Назва міста
color	varchar(30)	Колір міста
bord_color	varchar(30)	Колір кордону
bord	double	Ширина кордону

Дані про область зберігаються в сутності “область”. Опис даної сутності приведено в таблиці 2.4.

Таблиця 2.4 – Опис сутності “область”

Атрибут	Тип даних	Опис
prim_key	int	Первинний ключ
geom	geometry	Значення об'єкту в евклідовій системі координат
id	int	Ідентифікатор області
name	varchar(100)	Назва області
color	varchar(30)	Колір області
bord_color	varchar(30)	Колір кордону
bord	double	Ширина кордону

Дані про Чорне море зберігаються в сутності “ЧерноеМоре”. Опис даної сутності приведено в таблиці 2.5.

Таблиця 2.5 – Опис сутності “ЧерноеМоре”

Атрибут	Тип даних	Опис
prim_key	int	Первинний ключ
geom	geometry	Значення об'єкту в евклідовій системі координат
id	int	Ідентифікатор області
name	varchar(50)	Назва моря
color	varchar(30)	Колір моря
bord_color	varchar(30)	Колір кордону
bord	double	Ширина кордону

Дані про сусідні об'єкти з Одеською областю зберігаються в сутності “Посторонние”. Опис даної сутності приведено в таблиці 2.6.

Таблиця 2.6 – Опис сутності “Посторонние”

Атрибут	Тип даних	Опис
prim_key	int	Первинний ключ
geom	geometry	Значення об'єкту в евклідовій системі координат
id	int	Ідентифікатор області
dist_hor	double	Дистанція по горизонталі
dist_vert	double	Дистанція по вертикалі
off_hor	double	Поза горизонтальної осі
off_vert	double	Поза вертикальної осі
color	varchar(30)	Колір сусідніх об'єктів
bord_color	varchar(30)	Колір кордону
bord	double	Ширина кордону

Дані про індекси районів в області зберігаються в сутності “area_index”.

Опис даної сутності приведено в таблиці 2.7.

Таблиця 2.7 – Опис сутності “area_index”

Атрибут	Тип даних	Опис
prim_key	int	Первинний ключ
id	int	Ідентифікатор району
Район	varchar(100)	Назва району

Дані по тваринництву зберігаються в сутності “Животноводство”. Опис даної сутності приведено в таблиці 2.8.

Таблиця 2.8 – Опис сутності “Животноводство”

Атрибут	Тип даних	Опис
prim_key	int	Первинний ключ
id	int	Ідентифікатор виду с-х тварин
Год	date	Рік збору даних

Продовження таблиці 2.8

Атрибут	Тип даних	Опис
Район	varchar(100)	Назва району
КРС_П	double	К-сть поголів'я великої рогатої худоби в с-х підприємствах
КРС_Н	double	К-сть поголів'я великої рогатої худоби в господарствах населення
Свиньи_П	double	К-сть поголів'я свиней в с-х підприємствах
Свиньи_Н	double	К-сть поголів'я свиней в господарствах населення
Овцы_П	double	К-сть поголів'я овець в с-х підприємствах
Овцы_Н	double	К-сть поголів'я овець в господарствах населення

Дані по птахівництву зберігаються в сутності “Птицеводство”. Опис даної сутності приведено в таблиці 2.9.

Таблиця 2.9 – Опис сутності “Птицеводство”

Атрибут	Тип даних	Опис
prim_key	int	Первинний ключ

Продовження таблиці 2.9

id	int	Ідентифікатор виду птахів
Год	date	Рік збору даних
Район	varchar(100)	Назва району
Куры_Пет	double	К-сть курей і півнів
Несушки	double	К-сть несучок
Гуси	double	К-сть гусей
Прочие	double	К-сть інших птахів
Молодняк	double	К-сть молодняка

Дані по рослинництву зберігаються в сутності “Растениеводство”. Опис даної сутності приведено в таблиці 2.10.

Таблиця 2.10 – Опис сутності “Растениеводство”

Атрибут	Тип даних	Опис
prim_key	int	Первинний ключ
id	int	Ідентифікатор виду культур
Год	date	Рік валового збору

Продовження таблиці 2.10

Атрибут	Тип даних	Опис
Район	varchar(100)	
Сады	double	Первинний ключ
Виноград	double	Ідентифікатор виду культур
Пшеница	double	К-сть пшениці
Ячень	double	К-сть ячменю
Овес	double	К-сть вівса
Рис	double	К-сть рису
Кукуруза	double	К-сть кукурудзи
Соя	double	К-сть сої
Подсолнечник	double	К-сть соняшників
Рапс	double	К-сть ріпаку
Сах_свекла	double	К-сть цукрових буряків
Овоци	double	К-сть овочів
Бахчевые	double	К-сть баштанних
Картофель	double	К-сть картоплі

Продовження таблиці 2.10

Атрибут	Тип даних	Опис
Капуста	double	К-сть капусти
Морковь	double	К-сть моркви

База даних нашого програмного засобу буде зберігати дані сутностей, більша частина усіх уявлень і процедур буде реалізована у функціях програми.

2.3 Проектування елементів маніпулювання з даними БД

Представлення - це збережений результатний набір запити. Він доступний як віртуальна таблиця, що складається з результатів запити. На відміну від звичайних таблиць в реляційній БД, розріз даних не є частиною схеми даних: це динамічна, віртуальна таблиця що є результатом обробки даних з бази. Зміна даних в таблицях бази даних змінює їх у відповідних розрізах. Є два види розрізів: віртуальні та матеріалізовані.

Віртуальні представлення насправді не зберігають результат виконання запити в базі даних, а щоразу динамічно обчислюють. Матеріалізовані представлення – зберігаються, як і звичайні таблиці. Матеріалізовані представлення мають одну проблему: вони повинні б були оновлюватись при кожному оновленні таблиць від яких вони залежать, проте це не так. Відповідно оновлення табличок, від яких вони залежать, буде проходити повільніше. Як

вирішення цієї проблеми можна вказати, що матеріалізовані представлення оновлюватимуться щодня (щогодини)[13]¹⁾.

У нашому застосуванні під час ініціалізації буде відображено головне вікно, в якому можна буде отримати необхідну інформацію.

У головному вікні відображаються таблиці з даними по відповідним районам, а також по відповідним ресурсам, а також відповідні елементи інтерфейсу, завдяки яким можна проводити маніпуляції з базою даних, а саме:

- додавати нові дані в базу даних;
- видаляти дані з бази даних;
- змінювати дані у базі даних;
- розраховувати відповідні значення і будувати графіки по ним.

На рис.2.2 можна побачити головне вікно та обрану вкладку “Животноводство”

id	Год	Район	КРС_П	КРС_Н	Свиньи_П	Свиньи_Н	Овцы_П	Овцы_Н
42	31.12.2007	Ивановский	1822	11015	1007	10881	1024	4495
41	31.12.2006	Ивановский	2848	12778	1869	12069	669	4556
44	31.12.2009	Ивановский	1243	10191	1644	11927	970	6174
45	31.12.2010	Ивановский	1316	10025	2144	11999	61	6194
43	31.12.2008	Ивановский	1269	9969	1150	10888	1015	5683

Рисунок 2.2 – Вкладка “Животноводство”

На рис.2.3 можна побачити, як при натисканні на відповідну комірку в таблиці, яку ми виводимо, заповнюються поля, для подальшої обробки.

¹⁾ [13] Розріз даних – Вікіпедія. URL: https://uk.wikipedia.org/wiki/Розріз_даних (дата звернення 16.05.2019).

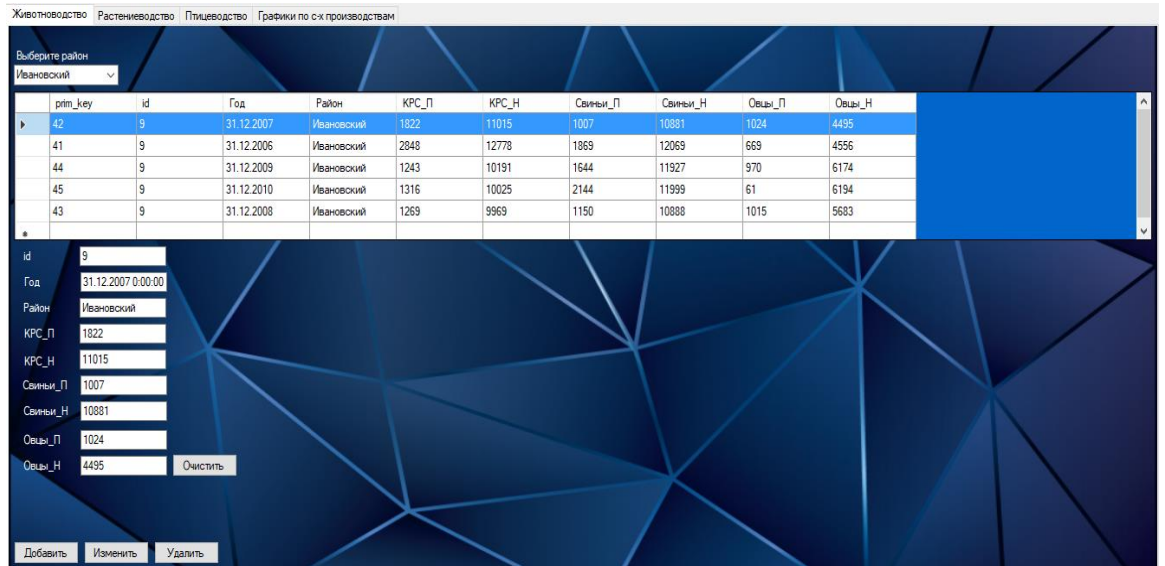


Рисунок 2.3 – Внесения данных у поля

На рис.2.4 можна побачити, як при натисканні на відповідний елемент інтерфейсу(кнопку), в таблиці, яку ми виводимо, додається нове значення, дані для якого беруться з відповідних полів.

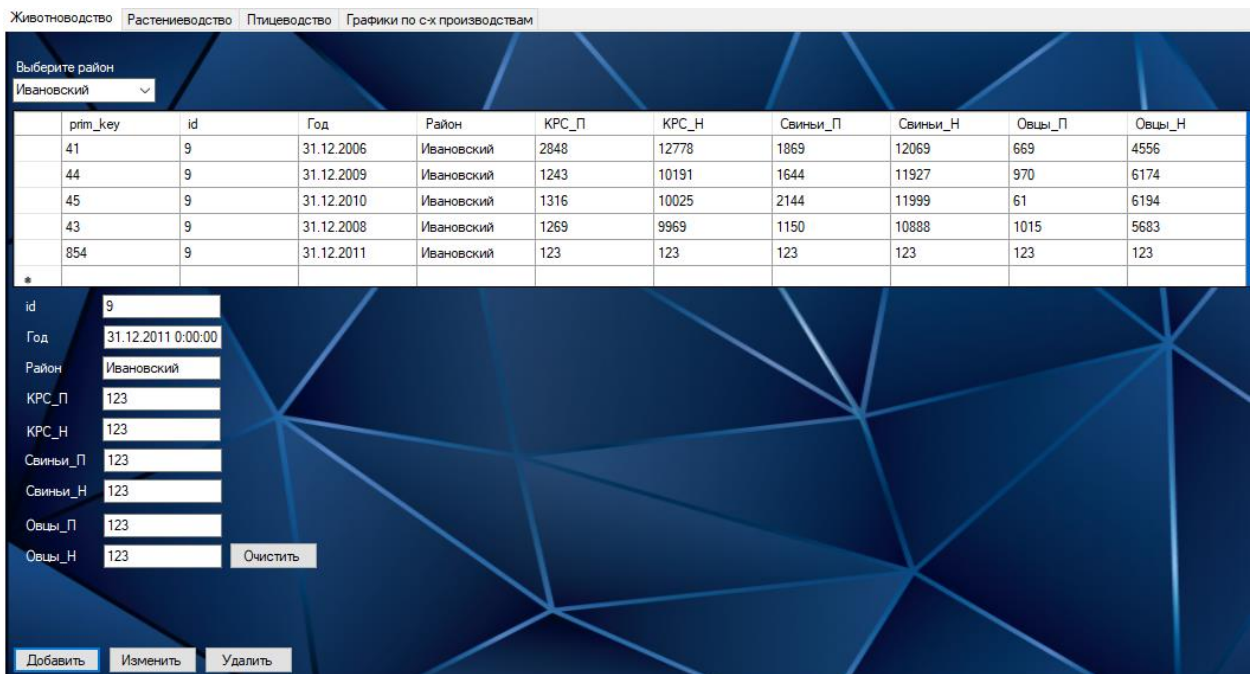


Рисунок 2.4 – Додавання нового елементу

На рис.2.5 можна побачити, як при натисканні на відповідний елемент інтерфейсу(кнопку), в таблиці, яку ми виводимо, змінюється елемент, котрий ми обрали, дані змінюються на відповідні дані з полів.

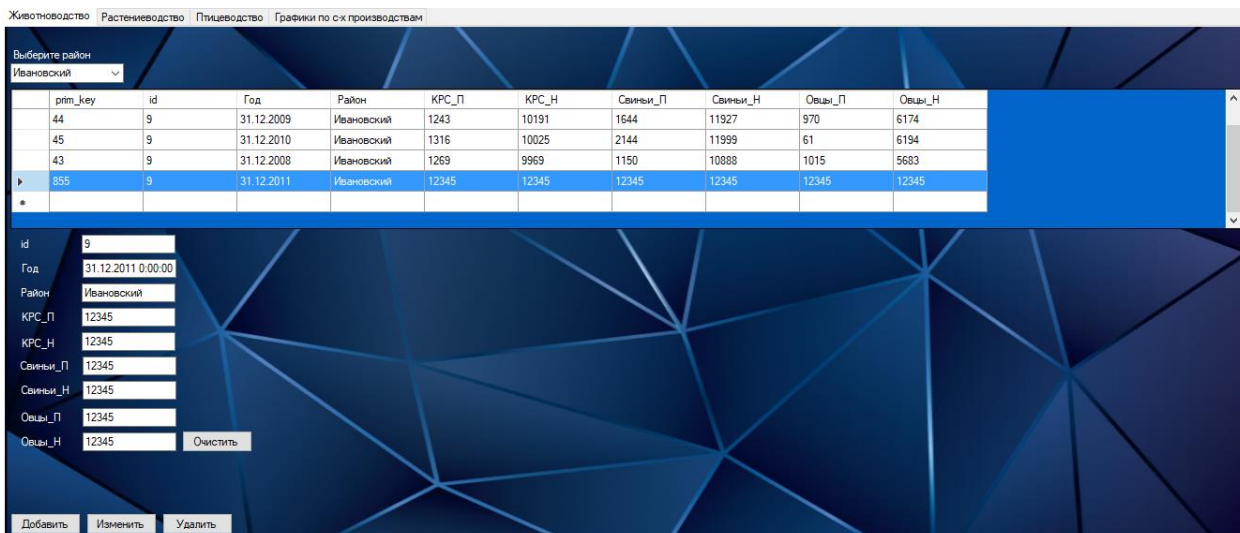


Рисунок 2.5 – Зміна значень обраного елемента на нові з полів

На рис.2.6 можна побачити, як при натисканні на відповідний елемент інтерфейсу(кнопку), в таблиці, яку ми виводимо, видаляється значення, яке ми обрали при натисканні на відповідну комірку в таблиці.

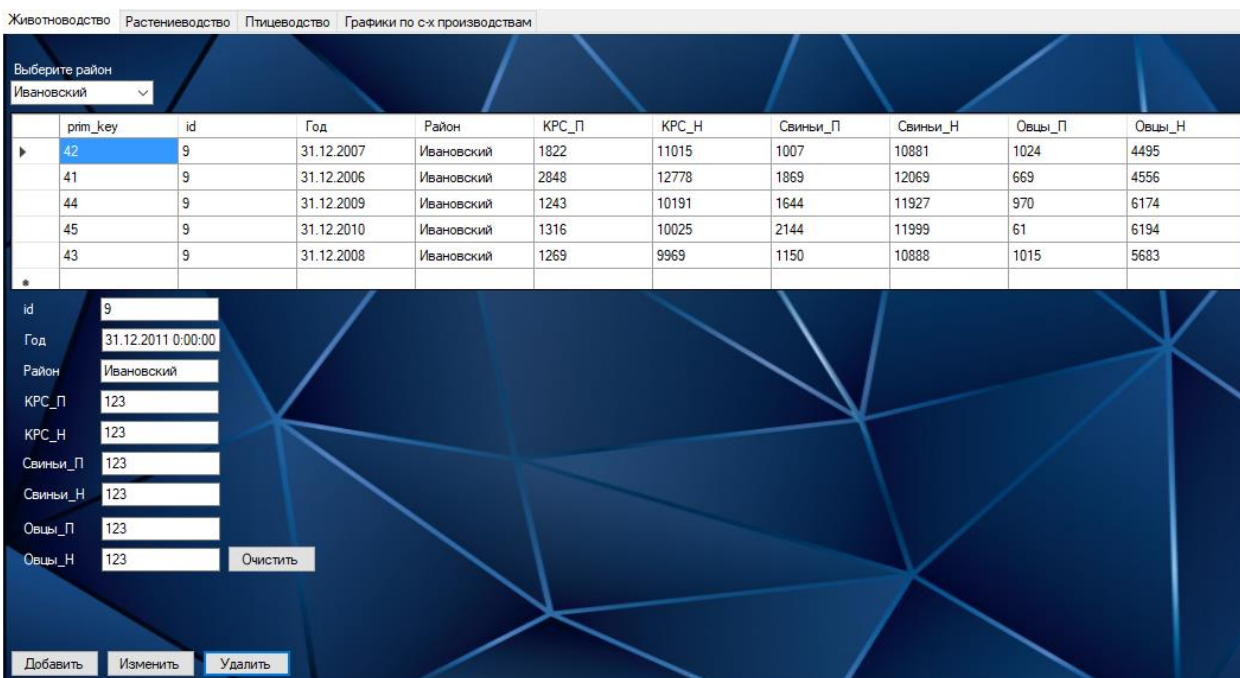


Рисунок 2.6 – Оновлене вікно, і видалене значення з таблиці

На рис.2.6 можна побачити, головне вікно та обрану вкладку “Растениеводство”

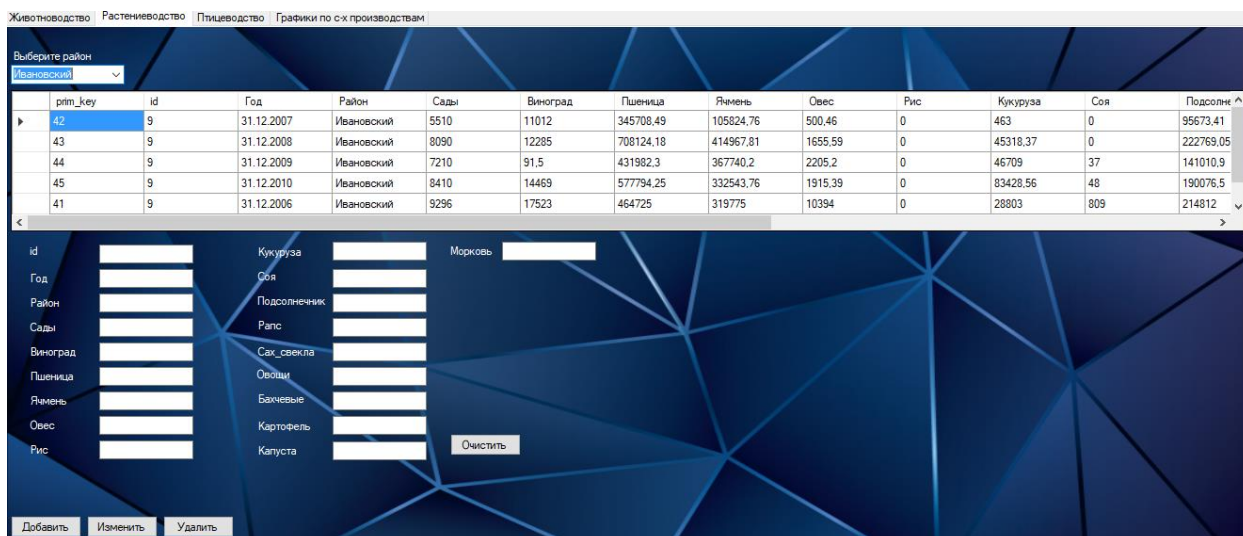


Рисунок 2.7 – Вкладка “Растениеводство”

На рис.2.8 можна побачити, як при натисканні на відповідну комірку в таблиці, яку ми виводимо, заповнюються поля, для подальшої обробки.

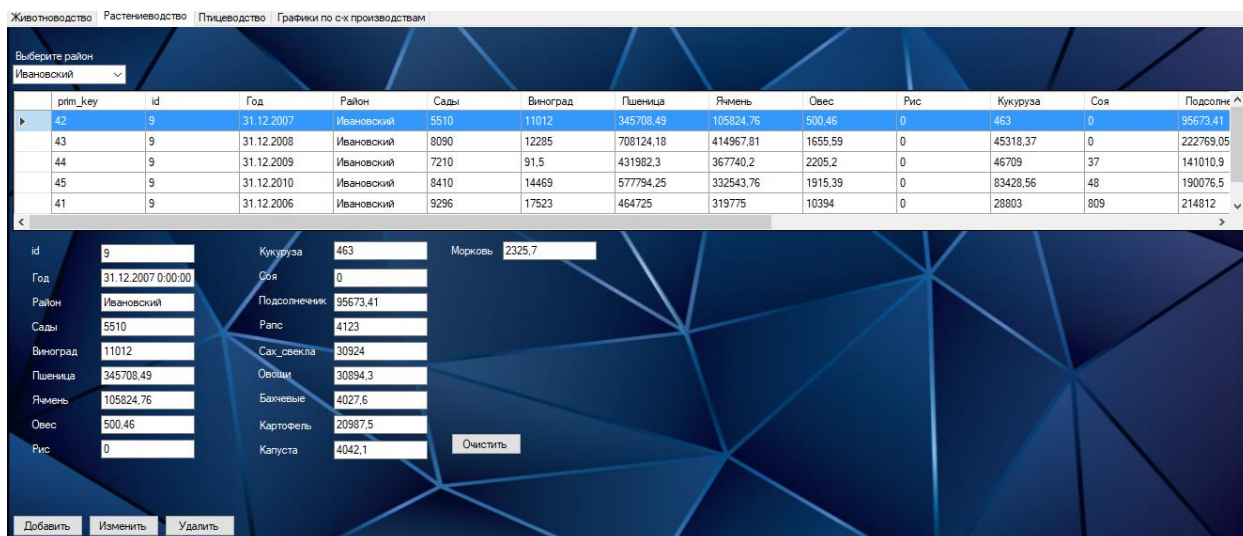


Рисунок 2.8 - Внесения данных у поля

На рис.2.9 можна побачити, як при натисканні на відповідний елемент інтерфейсу(кнопку), в таблиці, яку ми виводимо, додається нове значення, дані для якого беруться з відповідних полів.

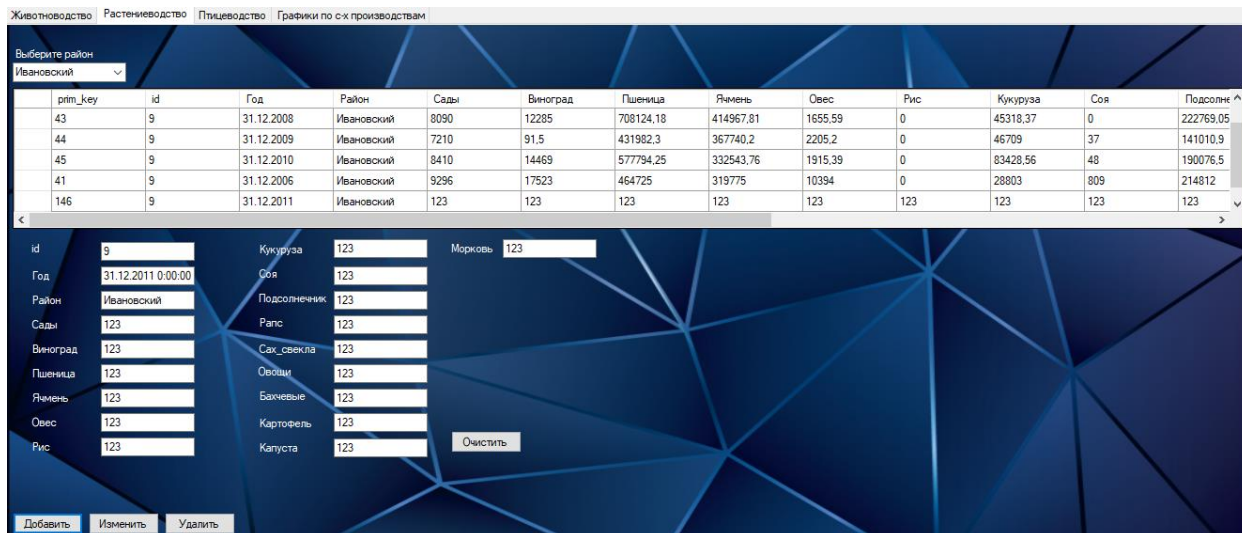


Рисунок 2.9 - Додавання нового елемента

На рис.2.10 можна побачити, як при натисканні на відповідний елемент інтерфейсу(кнопку), в таблиці, яку ми виводимо, змінюється елемент, котрий ми обрали, дані змінюються на відповідні дані з полів.

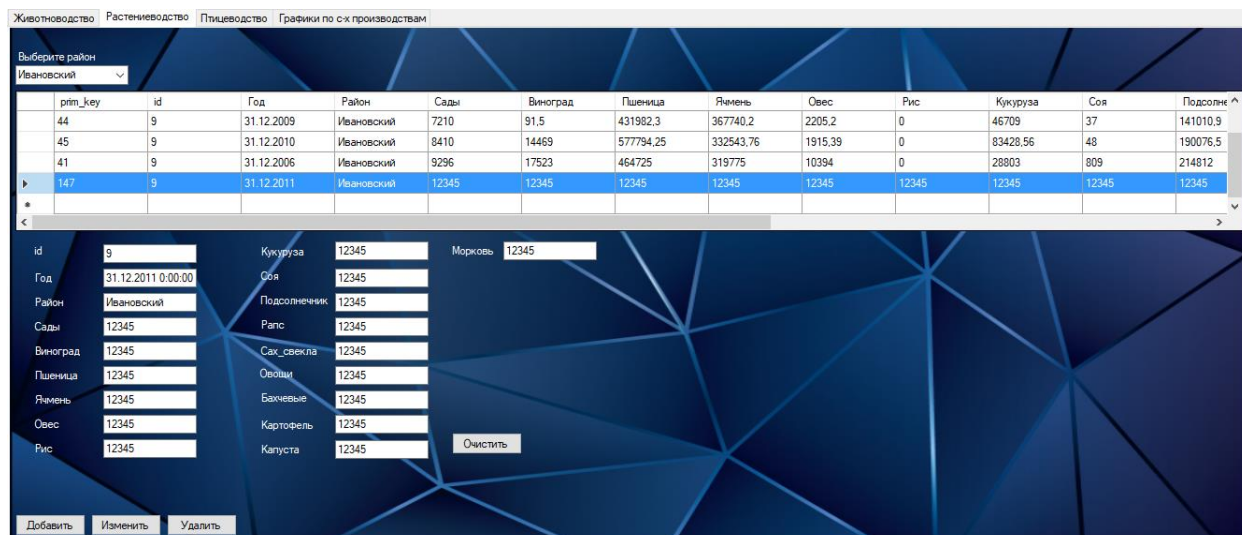


Рисунок 2.10 - Зміна значень обраного елемента на нові з полів

На рис.2.11 можна побачити, як при натисканні на відповідний елемент інтерфейсу(кнопку), в таблиці, яку ми виводимо, видаляється значення, яке ми обрали при натисканні на відповідну комірку в таблиці.

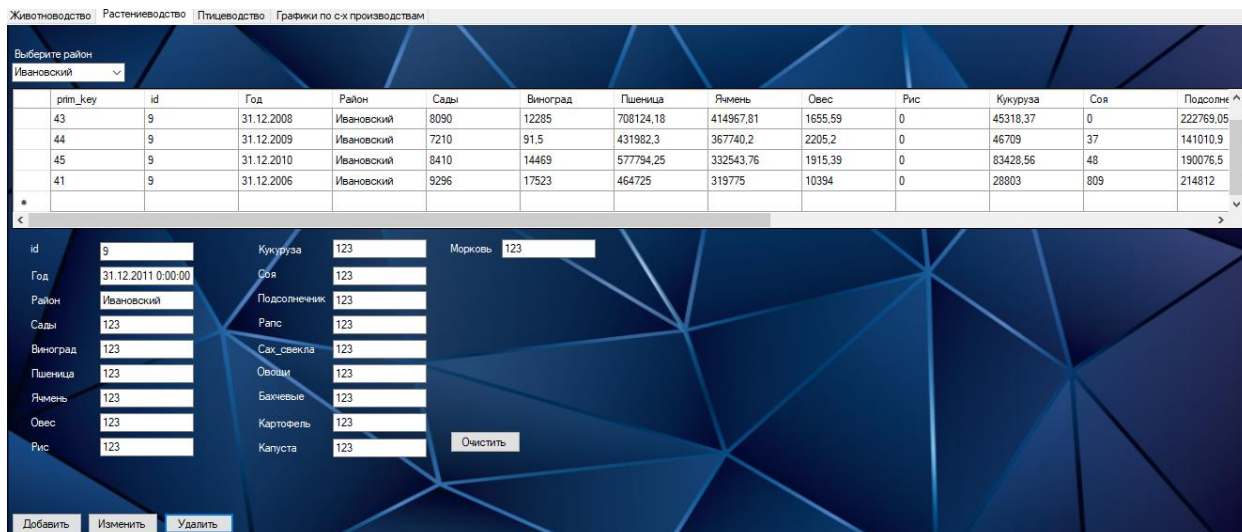


Рисунок 2.11 - Оновлене вікно, і видалене значення з таблиці

На рис.2.12 можна побачити головне вікно та обрану вкладку “Птицеводство”

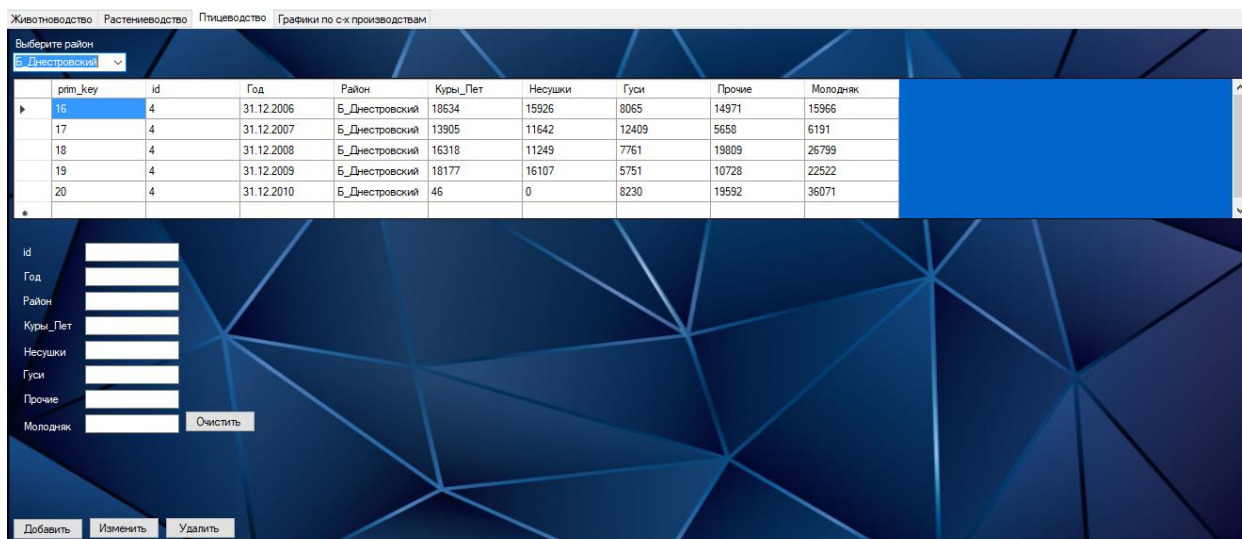


Рисунок 2.12 – Вкладка “Птицеводство”

На рис.2.13 можна побачити, як при натисканні на відповідну комірку в таблиці, яку ми виводимо, заповнюються поля, для подальшої обробки.

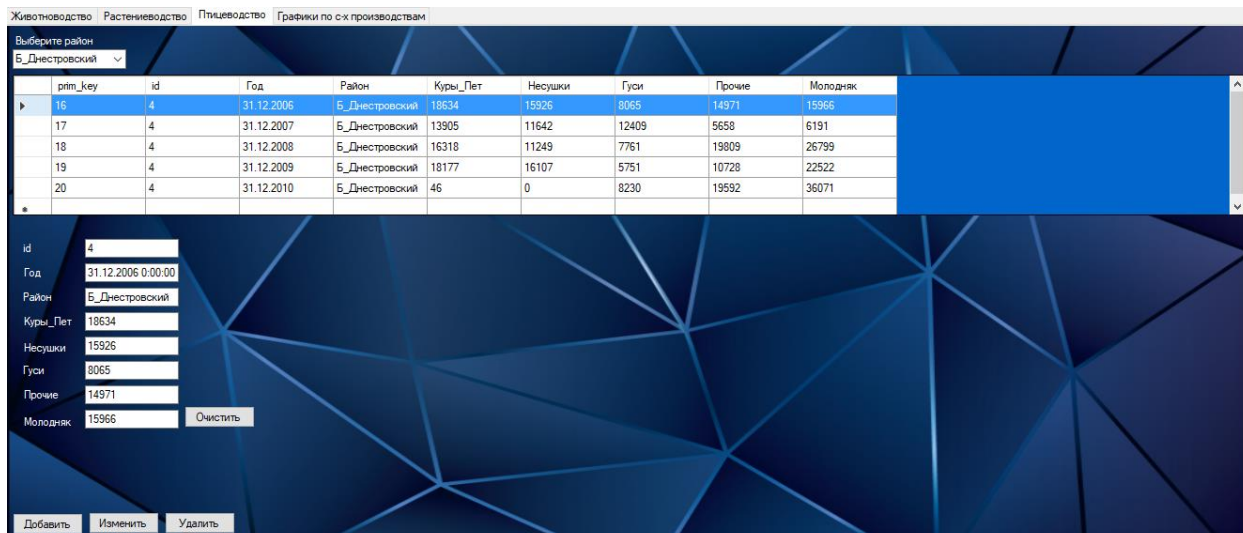


Рисунок 2.13 - Внесения даних у поля

На рис.2.14 можна побачити, як при натисканні на відповідний елемент інтерфейсу(кнопку), в таблиці, яку ми виводимо, додається нове значення, дані для якого беруться з відповідних полів.

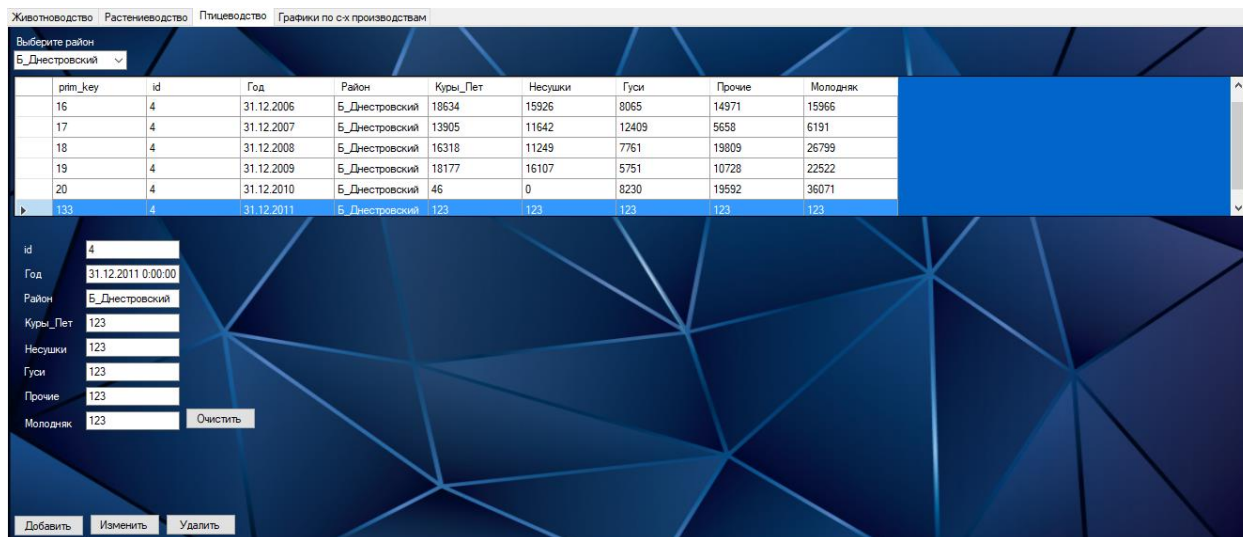


Рисунок 2.14 - Додавання нового елемента

На рис.2.15 можна побачити, як при натисканні на відповідний елемент інтерфейсу(кнопку), в таблиці, яку ми виводимо, змінюється елемент, котрий ми обрали, дані змінюються на відповідні дані з полів.

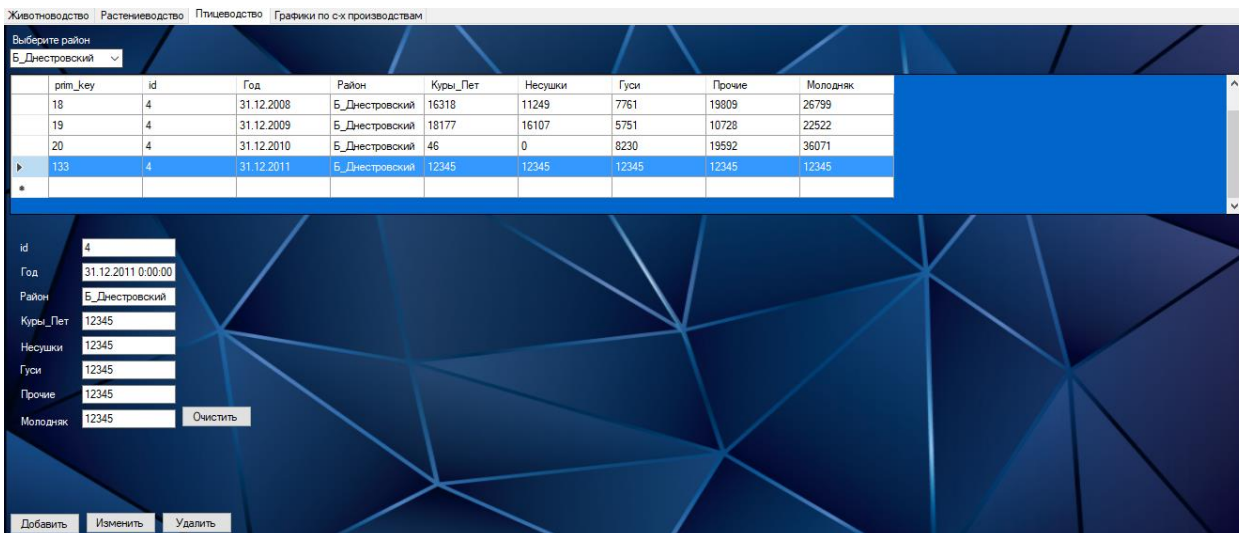


Рисунок 2.15 - Зміна значень обраного елемента на нові з полів

На рис.2.16 можна побачити, як при натисканні на відповідний елемент інтерфейсу(кнопку), в таблиці, яку ми виводимо, видаляється значення, яке ми обрали при натисканні на відповідну комірку в таблиці.

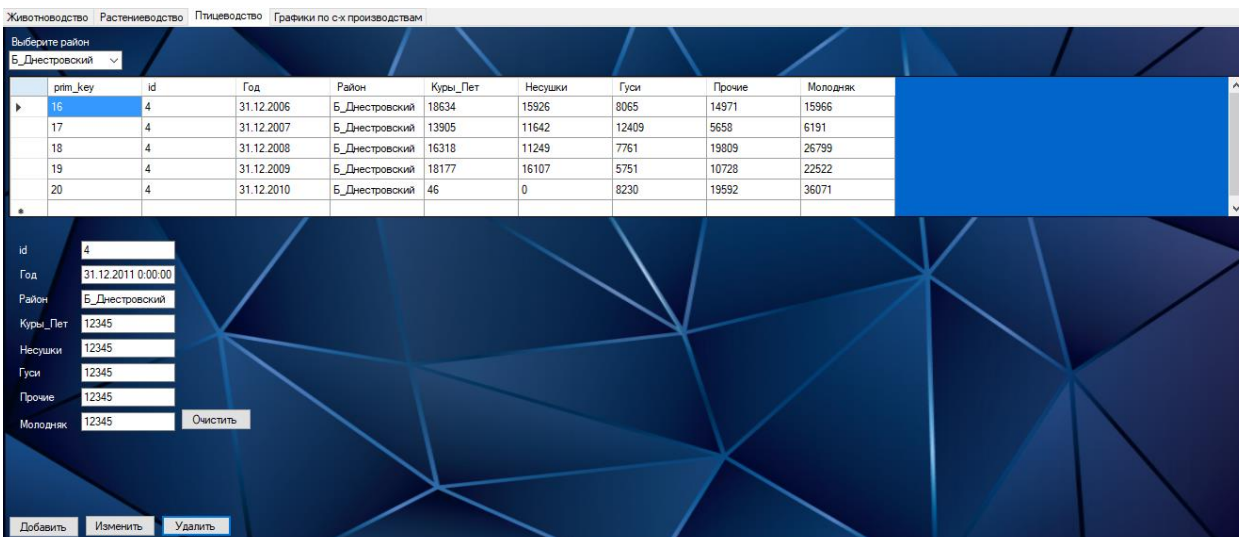


Рисунок 2.16 - Оновлене вікно, і видалене значення з таблиці

На рис.2.17 можна побачити головне вікно та обрану вкладку “Графіки по с-х производствам”

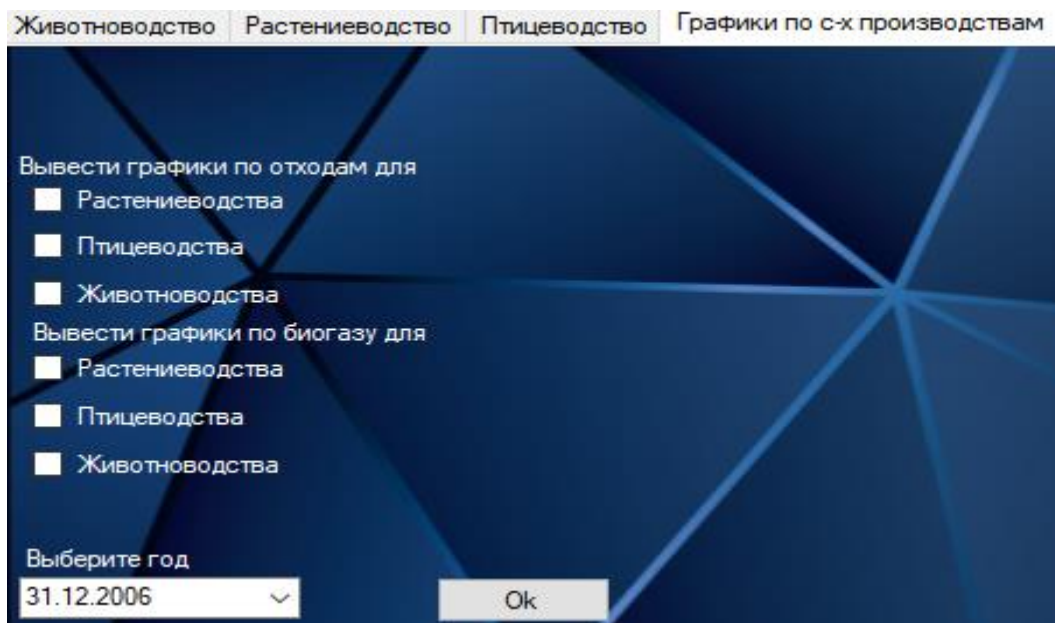


Рисунок 2.17 – Вкладка “Графики по с-х производствам”

На рис.2.18 можна побачити головне вікно, обрану вкладку “Графики по с-х производствам” та обрані елементи для побудови графіків.

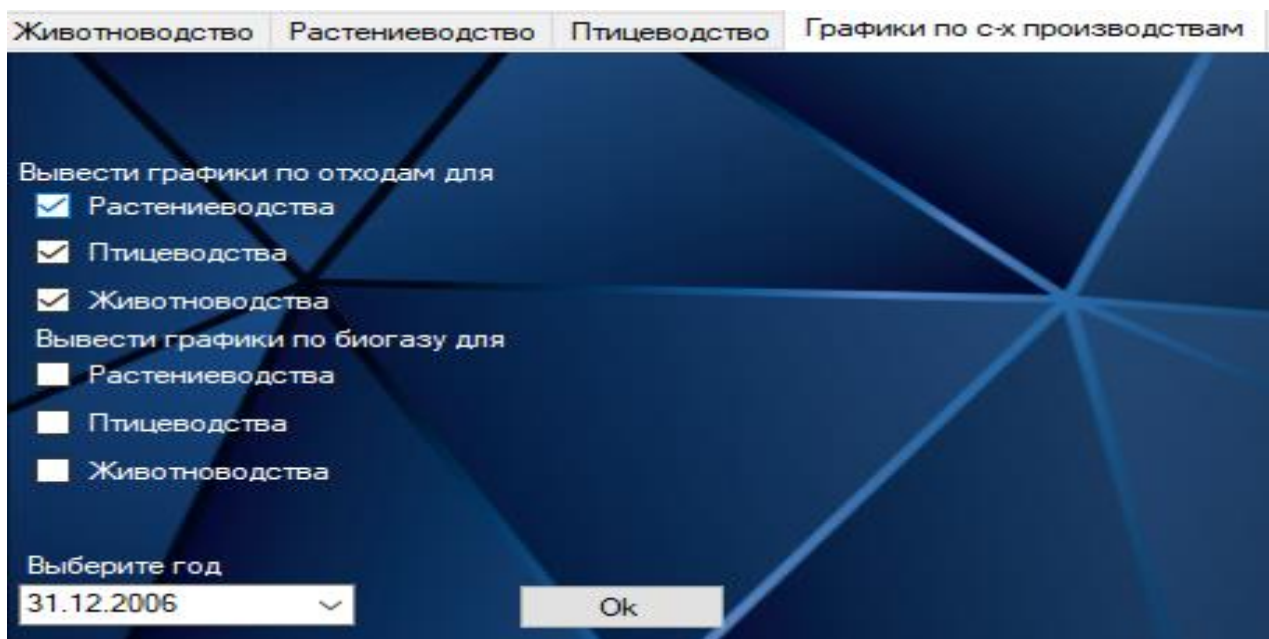


Рисунок 2.18 – Вкладка с обраними элементами для побудови графіків

На рис.2.19-2.21 можна побачити побудовані графіки, за обраними нами елементами у вкладці “Графики по с-х производствам”.

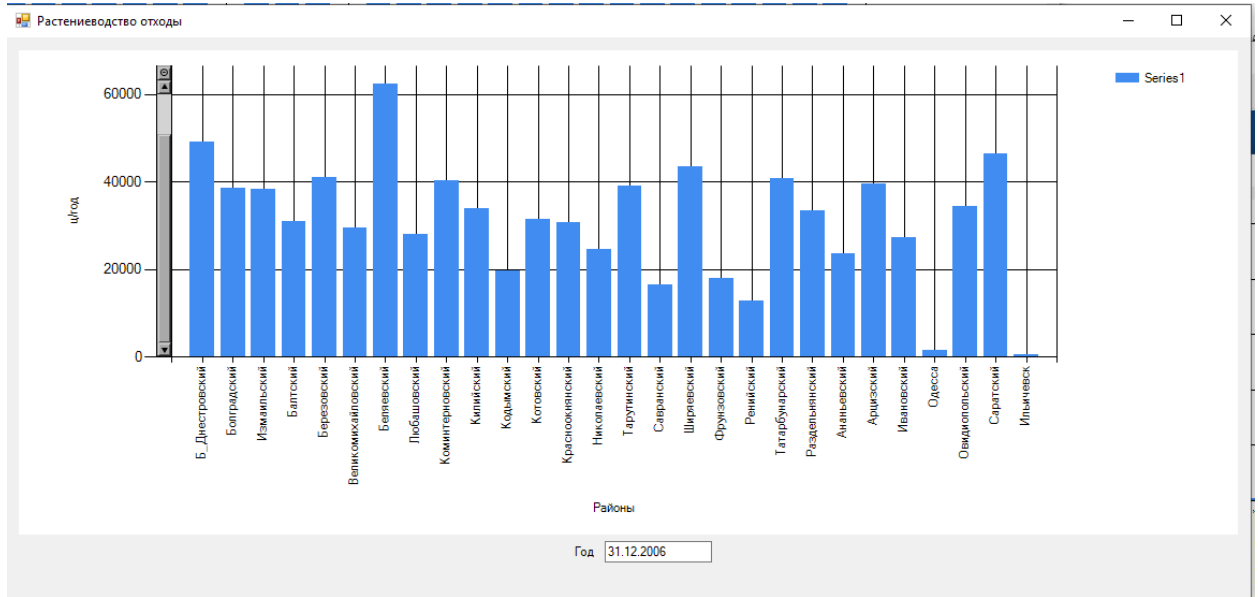


Рисунок 2.19 – Графік побудований з даних по рослинництву

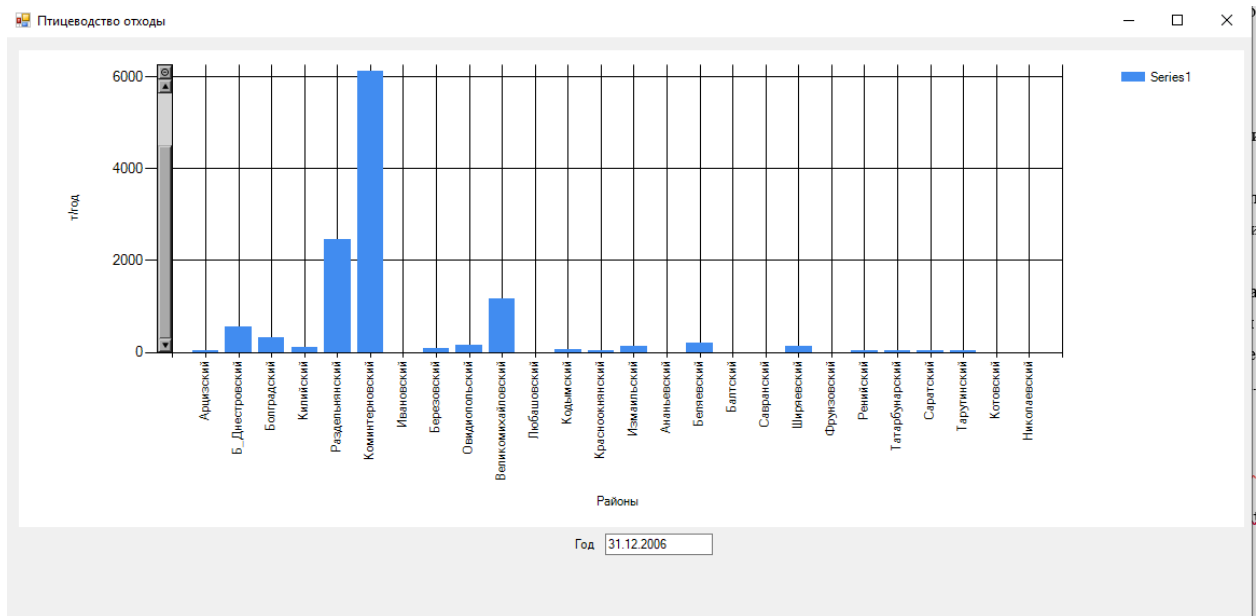


Рисунок 2.20 – Графік побудований з даних по птахівництву

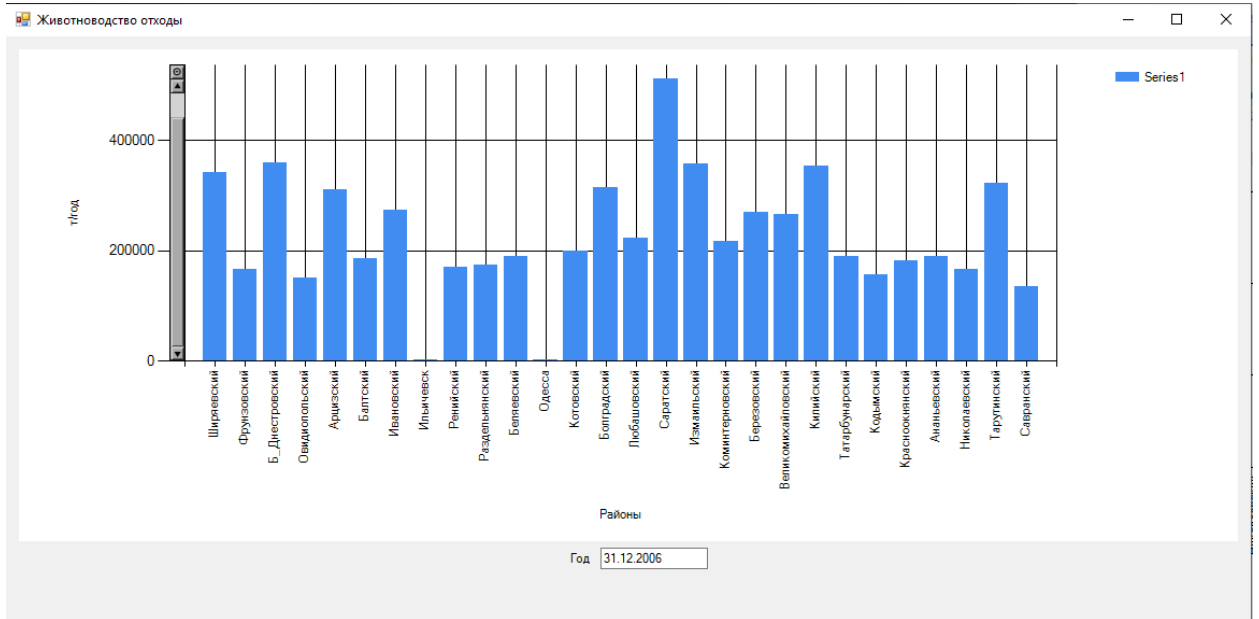


Рисунок 2.21 – Графік побудований з даних по тваринництву

На рис.2.22 можна побачити головне вікно, обрану вкладку “Графики по с-х производствам” та обрані елементи для побудови графіків по розрахунку біогазу.

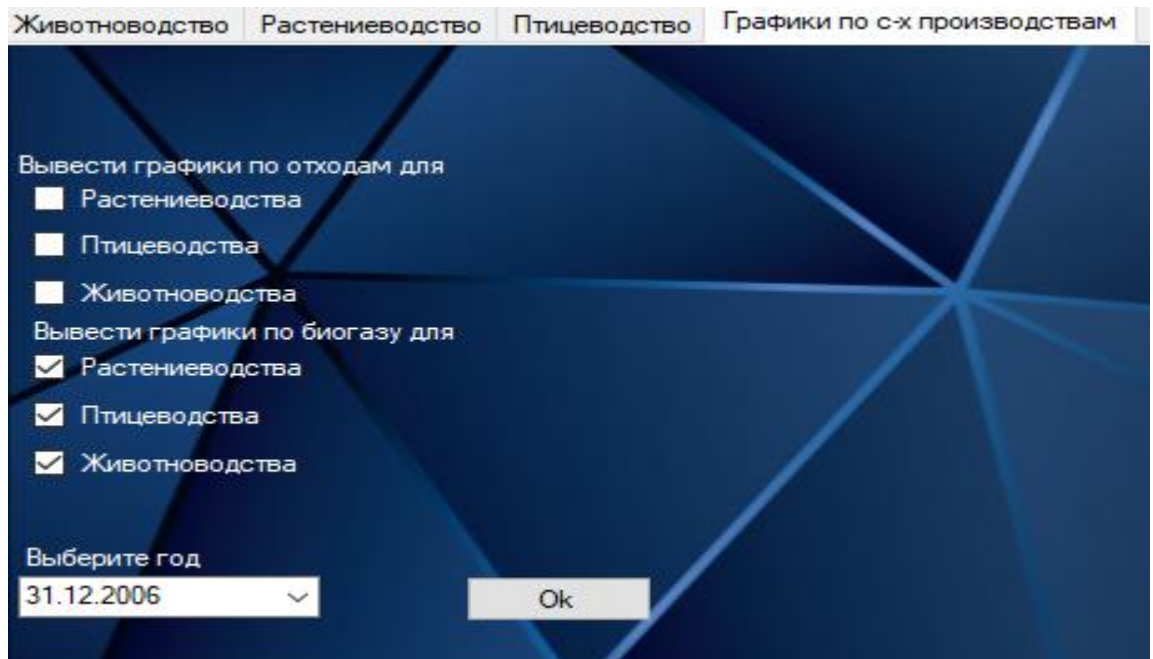


Рисунок 2.22 - Вкладка с обранными элементами для побудови графіків по біогазу

На рис.2.23-2.25 побачити побудовані графіки, за обраними нами елементами у вкладці “Графіки по с-х производствам”, по розрахунку біогазу.

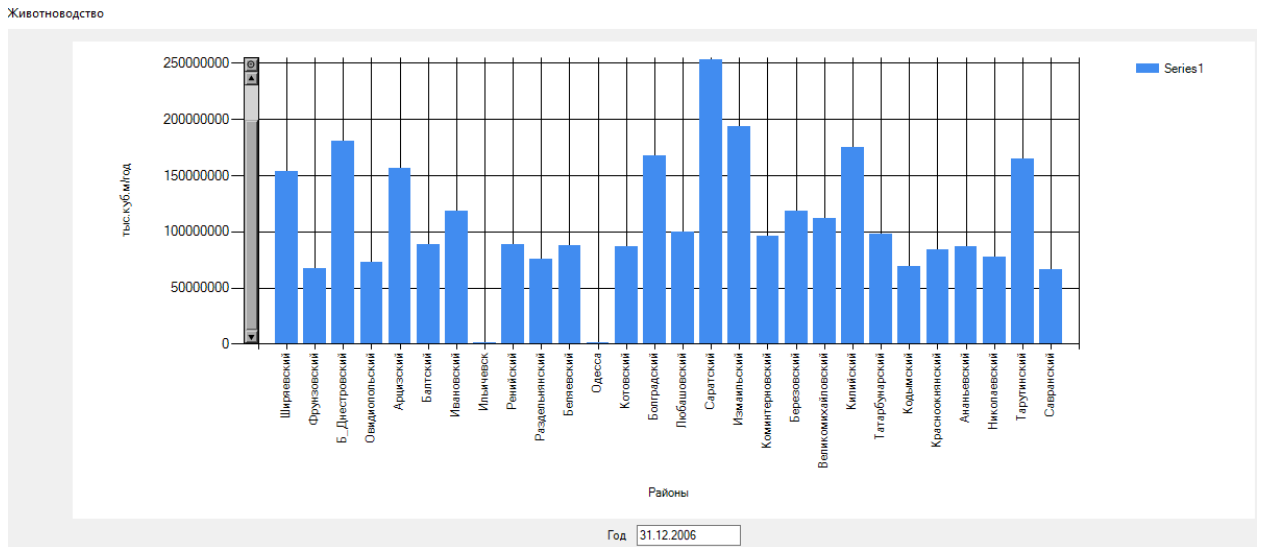


Рисунок 2.23 – Розрахунок і побудування графіку з даних по відходам по тваринництву (тис.куб.м)

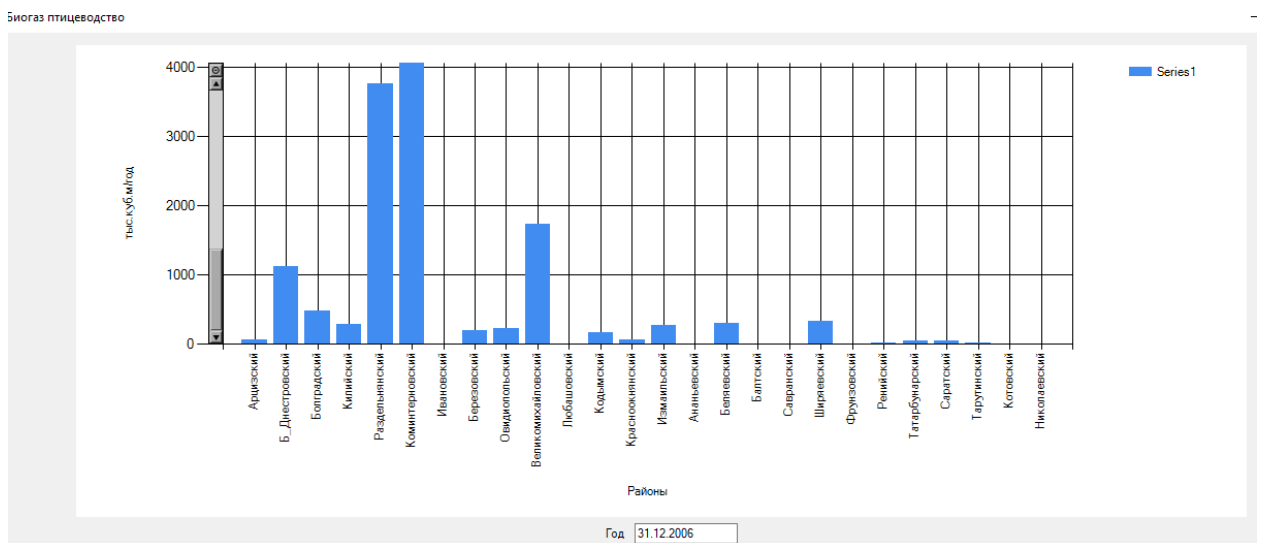


Рисунок 2.24 - Розрахунок і побудування графіку з даних по відходам по птахівництву (тис.куб.м)

Биогаз растениеводство

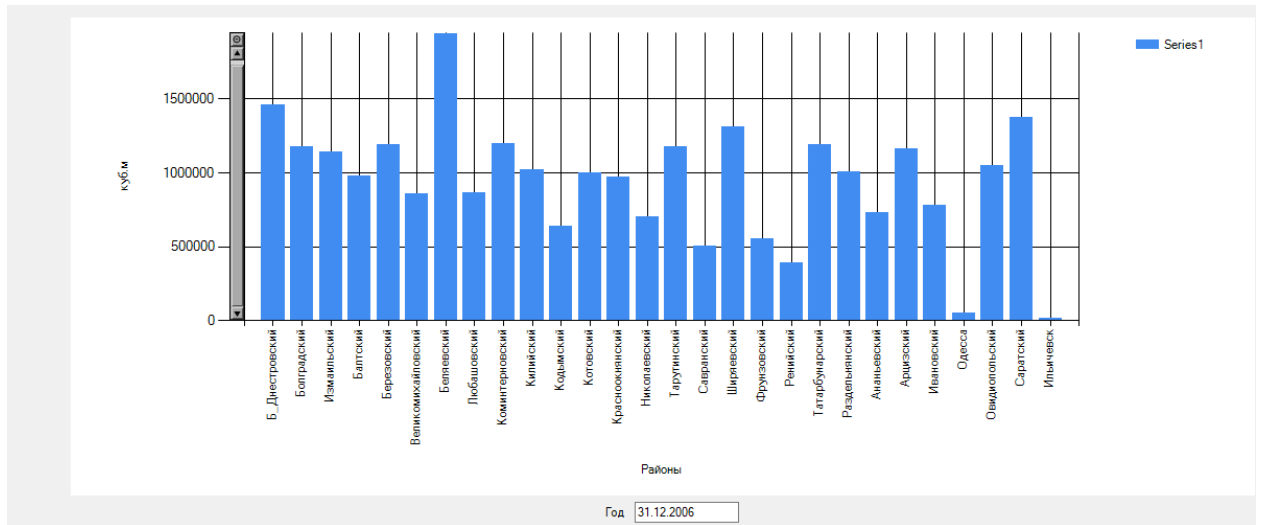


Рисунок 2.25 - Розрахунок і побудування графіку з даних по відходам по рослинництву (куб.м)

ВИСНОВКИ

Згідно з технічним завданням, був проведений аналіз предметної галузі, під час цього аналізу, були вибрані напрямки по розробці програмного застосування, для вирішення питань, щодо поводження з сільськогосподарськими відходами в Одеській області.

Була змодельована база даних, були обрані мова програмування та середовище розробки для нашого застосування, а саме Microsoft Visual Studio і C#.

В результаті розробки, був одержан програмний продукт, який дозволяє певною мірою працювати з даними по сільськогосподарським відходам в Одеській області, маніпулювати ними і наглядно виводити необхідні дані, в якості графіків, для проведення аналізу по ним.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

- (дата звернення 11.05.2019).
2. Qt – Википедия. URL: <https://ru.wikipedia.org/wiki/Qt> (дата звернення 12.05.2019).
 3. .NET Framework – Википедия. URL: https://ru.wikipedia.org/wiki/.NET_Framework (дата звернення 13.05.2019).
 4. Начало работы с .NET Framework | Microsoft Docs. URL: <https://docs.microsoft.com/ru-ru/dotnet/framework/get-started/> (дата звернення 13.05.2019).
 5. Microsoft SQL Server – Вікіпедія. URL: https://uk.wikipedia.org/wiki/Microsoft_SQL_Server (дата звернення 14.05.2019).
 6. MySql – Вікіпедія. URL: <https://uk.wikipedia.org/wiki/MySQL> (дата звернення 14.05.2019).
 7. PostgreSQL – Вікіпедія. URL: <https://uk.wikipedia.org/wiki/PostgreSQL> (дата звернення 14.05.2019).
 8. Qt Creator - Вікіпедія. URL: https://uk.wikipedia.org/wiki/Qt_Creator (дата звернення 15.05.2019).
 9. Eclipse – Вікіпедія. URL: <https://uk.wikipedia.org/wiki/Eclipse> (дата звернення 15.05.2019).
 10. Microsoft Visual Studio – Википедия. URL: https://ru.wikipedia.org/wiki/Microsoft_Visual_Studio (дата звернення 15.05.2019).
 11. Microsoft Visual Studio – описание, ссылка для скачивания, расширения файлов. URL: <https://open-file.ru/programs/microsoft-visual-studio> (дата звернення 15.05.2019).
 12. Microsoft Visual Studio – Wikipedia. URL: https://en.wikipedia.org/wiki/Microsoft_Visual_Studio (дата звернення 15.05.2019).
 13. Розріз даних – Вікіпедія. URL: https://uk.wikipedia.org/wiki/Розріз_даних (дата звернення 16.05.2019).