

Міністерство освіти і науки України

Луцький національний технічний університет

(повне найменування закладу вищої освіти)

Факультет комп'ютерних та інформаційних технологій

(повне найменування факультету)

Кафедра комп'ютерної інженерії та кібербезпеки

(повне найменування кафедри)

**КВАЛІФІКАЦІЙНА РОБОТА
ЗА СТУПЕНЕМ ВИЩОЇ ОСВІТИ «БАКАЛАВР»**

**ІНФОРМАЦІЙНА СИСТЕМА ПУНКТУ ОБМІНУ ВАЛЮТ
ЗАСОБАМИ C# WINFORMS**

**CURRENCY EXCHANGE POINT INFORMATION SYSTEM USING C#
WINFORMS**

спеціальність 123 Комп'ютерна інженерія

(шифр і назва спеціальності)

освітня програма Комп'ютерна інженерія

(назва освітньої програми)

Виконав: здобувач вищої освіти

групи КІ-42

Коць Богдан Сергійович

(підпис)

Керівник:

к.т.н., доцент

Пех Петро Антонович

(підпис)

Кваліфікаційну роботу

допущено до захисту

« 07 » червня 2024 р.

Гарант освітньої програми:

к.т.н., доцент

Лавренчук Світлана Василівна

(підпис)

Луцьк – 2024 року

ЛУЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

Факультет комп'ютерних та інформаційних технологій

Кафедра комп'ютерної інженерії та кібербезпеки

Ступінь вищої освіти: бакалавр

Галузь знань: 12 Інформаційні технології

Спеціальність: 123 Комп'ютерна інженерія

Освітня програма: «Комп'ютерна інженерія»

ЗАТВЕРДЖУЮ

Завідувач кафедри

проф. Н.Черняшук

« 10 » 01 2024 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧУ ВИЩОЇ ОСВІТИ

Коцю Богдану Сергійовичу

(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи Інформаційна система пункту обміну валют засобами C# WinForms

Керівник роботи к.т.н., доцент Пех Петро Антонович

затвержені наказом закладу вищої освіти від «30» грудня 2023 року № 459/01-02

2. Строк подання здобувачем вищої освіти кваліфікаційної роботи 11.06.2024р.

3. Вихідні дані до роботи Джерелом розробки є науково-технічна література та публікації в періодичних виданнях з даного питання, опубліковані зарубіжні та вітчизняні роботи в даній області та різні інтернет-ресурси технічного спрямування

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити):

Вступ

Аналіз проблеми роботи та постановка завдань дослідження розробки

Інформаційної системи пункту обміну валют засобами C#

Інструменти для розробки інформаційної системи пункту обміну валют засобами C#

WinForms

Розробка інформаційної системи пункту обміну валют

Висновки

5. Перелік графічного (ілюстративного) матеріалу:

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис	
		завдання видав	завдання прийняв
<i>Аналіз проблеми за темою роботи та постановка завдань дослідження</i>	<i>Пех П.А., доцент</i>		
<i>Інструменти для розробки інформаційної системи пункту обміну валют засобами C# WinForms</i>	<i>Пех П.А., доцент</i>		
<i>Розробка інформаційної системи пункту обміну валют</i>	<i>Пех П.А., доцент</i>		
<i>Нормоконтроль</i>	<i>Багнюк Н.В., доцент</i>		
<i>Гарант ОП</i>	<i>Лавренчук С.В., доцент</i>		
<i>Показник запозичень тексту</i>		_____%	
<i>Академічна доброчесність</i>	<i>Міскевич О.І., асистент</i>		

7. Дата видачі завдання 10.01.2024 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	<i>Розділ 1. Огляд програмних засобів Розробки інформаційної системи пункту обміну валют засобами C# WinForms</i>	до 15.02.2024 р.	Виконано
2.	<i>Розділ 2. Інструменти для розробки інформаційної системи пункту обміну валют засобами C# WinForms</i>	до 15.03.2024 р.	Виконано
3.	<i>Розділ 3. Розробка інформаційної системи пункту обміну валют</i>	до 04.05.2024 р.	Виконано
4.	<i>Формування висновків</i>	до 07.05.2025 р.	Виконано
5.	<i>Формування списку використаних джерел</i>	до 10.05.2024 р.	Виконано
6.	<i>Формування додатків</i>	до 15.05.2024 р.	Виконано
7.	<i>Оформлення ілюстративного матеріалу</i>	до 20.05.2024 р.	Виконано
8.	<i>Нормоконтроль</i>	до 06.06.2024 р.	Виконано
9.	<i>Інструментальна перевірка на академічний плагіат</i>	до 07.06.2024 р.	Виконано
10.	<i>Представлення кваліфікаційної роботи бакалавра до захисту</i>	до 11.06.2024 р.	Виконано

Здобувач вищої освіти

(підпис)

Коць Б.С.

(прізвище, ініціали)

Керівник кваліфікаційної роботи

(підпис)

Пех П.А.

(прізвище, ініціали)

АНОТАЦІЯ

Коць Б.С. Інформаційна система пункту обміну валют засобами C# WinForms.

Кваліфікаційна робота бакалавра ОП «Комп'ютерна інженерія» спеціальності 123 Комп'ютерна інженерія. Луцький національний технічний університет. Луцьк, 2024.

Кваліфікаційна робота бакалавра складається зі вступу, трьох розділів, висновків та списку використаних джерел та додатків.

У роботі обгрунтовано вибір мови та середовища програмування для розробки інформаційної системи пункту обміну валют у вигляді програмного проекту засобами C# WinForms. Дана детальна характеристика інструментів середовища C# WinForms для розробки програмного проекту та технологія роботи з ними. Розглянута технологія створення бази даних з інформацією про клієнтів пункту обміну. Розроблено програмний проект у складі: форми з інформацією про курси валют у банківських установах м. Луцька; форми з інформацією про курси валют в обмінниках м. Луцька; форми «Обмін валют» для прогнозування результатів виконання операцій щодо обміну валют; форми «Робота з базою даних»; форми «Обмін валют з підключенням бази даних».

Ключові слова: мова програмування C#, середовище C# winforms, курси валют, обмінний пункт, програмний проект

ANNOTATION

Kots B.S. Information system of the currency exchange point using C# WinForms.

Bachelor's qualifying thesis of the OP «Computer Engineering» specialty 123 Computer Engineering. Lutsk National Technical University. Lutsk, 2024.

The bachelor's qualification work consists of an introduction, three sections, conclusions and a list of used sources and appendices.

The work substantiates the choice of language and programming environment for the development of an information system of a currency exchange point in the form of a software project using C#WinForms. This is a detailed description of C# WinForms environment tools for software project development and the technology of working with them. The considered technology for creating a database with information about customers of the exchange point. A software project was developed, including: forms with information about currency rates in banking institutions in Lutsk; forms with information on exchange rates in exchange offices in Lutsk; «Currency exchange» form for forecasting the results of currency exchange transactions; forms «Working with the database»; form «Currency exchange with database connection».

Keywords: C# programming language, C# winforms environment, currency rates, exchange point, software project

ЗМІСТ

ВСТУП.....	7
РОЗДІЛ 1 ОГЛЯД ПРОГРАМНИХ ЗАСОБІВ РОЗРОБКИ ІНФОРМАЦІЙНОЇ СИСТЕМИ ПУНКТУ ОБМІНУ ВАЛЮТ	9
1.1 Інформаційні системи пункту обміну валют та завдання щодо її розробки.....	9
1.2 Огляд мов програмування для розробки інформаційних систем	11
1.3 Вибір середовища програмування для розробки проекту.....	12
РОЗДІЛ 2 ІНСТРУМЕНТИ ДЛЯ РОЗРОБКИ ІНФОРМАЦІЙНОЇ СИСТЕМИ ПУНКТУ ОБМІНУ ВАЛЮТ ЗАСОБАМИ C# WINFORMS	15
2.1 Компоненти Панелі елементів середовища C#WinForms	15
2.2 Створення об'єктів проекту та керування їх властивостями та подіями.....	17
2.3 Технологія створення бази даних засобами C#WinForms.....	20
РОЗДІЛ 3 РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ ПУНКТУ ОБМІНУ ВАЛЮТ.....	24
3.1 Структура інформаційної системи пункту обміну валют	24
3.2 Створення форм проекту та забезпечення навігації по них	26
3.3 Форма Form1 «Банки»	30
3.4 Форма Form2 «Обмінники»	35
3.5 Форма Form3 «Обмін валют»	36
3.6 Форма Form4 «Робота з базою даних»	50
3.7 Форма Form5 «Обмін валют з підключенням бази даних»	62
ВИСНОВКИ	69
СПИСКИ ВИКОРИСТАНИХ ДЖЕРЕЛ	70
ДОДАТКИ	72

ВСТУП

Актуальність теми. У теперішній час інформаційні системи стрімко проникають у всі сфери діяльності людини. Сьогодні кращих результатів добиваються ті, хто вміє ефективно організовувати процеси збору, обробки, систематизації та передачі інформації. Інформаційні системи якраз і вирішують ці завдання. Не є винятком інформаційна система пункту обміну валют. Клієнти пункту обміну повинні оперативно отримувати інформацію про курси обміну валют у різних банківських установах та обмінниках, мати змогу прогнозувати результати виконання операцій з обміну валют з метою пошуку оптимальних варіантів. Вирішення цих питань шляхом розробки інформаційної системи пункту обміну валют у вигляді програмного проекту засобами C# WinForms є, на нашу думку, вкрай актуальною задачею.

Мета дослідження полягає у розробці інформаційної системи пункту обміну валют у вигляді програмного проекту засобами C# WinForms.

Об'єкт дослідження – інформаційні системи пункту обміну валют у вигляді програмних проектів засобами C# WinForms

Предмет дослідження – засоби та процес розробки програмних проектів на платформі C# WinForms. Розробка передбачає обґрунтування вибору мови і середовища програмування, інструментів для вирішення поставлених завдань та безпосереднє створення інформаційної системи пункту обміну валют у вигляді програмного проекту засобами C# WinForms.

Завдання, які необхідно виконати:

1. Виконати огляд програмних засобів для розробки інформаційної системи пункту обміну валют у вигляді програмного проекту засобами C# WinForms та обґрунтувати їх вибір для вирішення завдань проекту.

2. Розробити у складі програмного проекту форми з інформацією про курси валют банків та обмінників м. Луцька.

3. Розробити у складі програмного проекту форму для виконання операцій з обміну валют у разі ручного введення вхідних даних.

4. Розробити у складі програмного проекту форму для роботи з базою даних з інформацією про клієнтів.

5. Розробити у складі програмного проекту форму для виконання операцій з обміну валют з підключенням бази даних з інформацією про клієнтів.

Методи розробки. Розробка проекту базується на застосуванні сучасної технології об'єктно-орієнтованого програмування.

Наукова новизна одержаних результатів полягає у розробці інформаційної системи пункту обміну валют у вигляді програмного проекту сучасними засобами C# WinForms.

Практичне значення одержаних результатів. Розроблена інформаційна система пункту обміну валют у вигляді програмного проекту засобами C# WinForms може бути використана для оптимізації роботи реальних обмінних пунктів та у навчальному процесі.

Особистий внесок здобувача. Програмний комплекс, запропонований у кваліфікаційній роботі, розроблений здобувачем самостійно.

Апробація результатів.

Публікації. P. Pekh, N. Khrystinets, R. Gubish, V. Shulgach. Simulation of two-stage temperature regulation system. // Computer-integrated technologies: education, science, production: Scientific journal. Lutsk: LNTU, 2023. Issue No. 53. Article 50-57. <https://doi.org/10.36910/6775-2524-0560-2023-51-14>.

РОЗДІЛ 1

ОГЛЯД ПРОГРАМНИХ ЗАСОБІВ РОЗРОБКИ ІНФОРМАЦІЙНОЇ СИСТЕМИ ПУНКТУ ОБМІНУ ВАЛЮТ

1.1 Інформаційні системи пункту обміну валют та завдання щодо її розробки

Відомі різні визначення інформаційної системи, одне з яких найбільш повно характеризує її сутність, наводимо нижче.

«Інформаційна система (information system) – сукупність організаційних і технічних засобів для збереження та обробки інформації з метою забезпечення інформаційних потреб користувачів» [1].

Закон України «Про захист інформації в інформаційно-телекомунікаційних системах» інформаційна (автоматизована) система являє собою організовану структуру, що складається з технічних та програмних компонентів, призначених для обробки інформації.

Згідно зі статтею 1 цього Закону:

«Інформаційна (автоматизована) система - це організаційно-технічна система, в якій реалізується технологія обробки інформації з використанням технічних і програмних засобів» [2].

На впровадження інформаційних систем впливають цілий ряд факторів.

«Основними факторами, які впливають на впровадження інформаційних систем, є потреби організацій та користувачів, а також наявність відповідних засобів для їх формування. Найсуттєвіше на розвиток інформаційних систем вплинули досягнення в галузі комп'ютерної техніки та телекомунікаційних мереж» [1].

«Причини, що спонукають організації впроваджувати інформаційні системи, з одного боку обумовлюються прагненням збільшити продуктивність повсякденних робіт чи усунути їх повторне проведення, а з іншого боку бажанням підвищити ефективність управління діяльністю організації за рахунок прийняття оптимальних та раціональних управлінських рішень.

Перша причина доволі прозора і для її реалізації достатньо впроваджувати стандартизовані системи обробки інформації. Успішне функціонування організації значною мірою залежить від вдалого керівництва, яке базується на обґрунтуванні перспективних концепцій розвитку згідно зі своєчасною, достовірною та повною інформацією, яку може поставляти відповідна інформаційна система. Основне завдання інформаційної системи управління полягає у підпорядкуванні всіх внутрішніх процесів головним цілям організації. Для цього необхідно скоординувати процеси, пов'язані з діяльністю організації таким чином, щоб вони максимально забезпечували виконання поставлених задач в єдиному інформаційному полі. Тільки таким чином інформаційна озброєність організації починає безпосередньо впливати на ефективність її діяльності» [1].

«До основних напрямків автоматизації інформаційно-управлінської діяльності в організаційних структурах відносять:

– автоматизацію обробки документів шляхом впровадження систем для обробки тексту, автоматизацію обміну інформацією через різноманітні види комунікацій (які включають АТС підприємства, відеотермінальні системи, локальну комп'ютерну мережу, телекопіювальні апарати, відеоінформаційні системи);

– автоматизацію діяльності менеджерів на базі комп'ютерних систем комплексних інформаційних систем, які надають допомогу в прийнятті рішень, та електронних секретарів, що дозволяє підвищити рівень організації праці менеджерів на якісно вищій щабель.

Впровадження інформаційних систем дозволяє менеджеру отримувати оперативний доступ до довільної нагромадженої інформації з тим, щоб надалі ефективно її використовувати для вирішення поставлених задач (у сферах аналізу маркетингу, фінансів, тощо)» [1].

У кваліфікаційній роботі завдання на розробку інформаційної системи пункту обміну валют, яка базується на використанні комп'ютерних засобів та технологій, передбачає отримання, переробку та передачу інформації, потрібної

для оптимального управління фінансовими ресурсами клієнтів. Клієнти пункту обміну валют, інформація про ресурси яких зберігаються у локальній базі даних, повинен мати змогу оперативно отримати інформацію про курси валют, здійснити операції обміну однієї валюти на іншу, виконати за потреби різні платежі. Розробка такої системи і є завданням даної кваліфікаційної роботи.

1.2 Огляд мов програмування для розробки інформаційних систем

Виходячи зі завдань, які потрібно вирішити у кваліфікаційній роботі, необхідно обгрунтовано підійти до вибору мови програмування.

«Мова програмування – це система позначень для опису алгоритмів і структур даних, певна штучна формальна система, засобами якої можна виражати алгоритми. Мову програмування визначає набір лексичних, синтаксичних і семантичних правил, що задають зовнішній вигляд програми та дії, які виконує виконавець (комп'ютер) під її управлінням» [3].

Однією з найбільш популярних мов програмування високого рівня в останні роки стала мова Python. Ця мова стрімко розвивається і має велику бібліотеку функцій, за рахунок використання якої можна розв'язувати найскладніші завдання. Однак відмінною рисою мови Python є її простота. Саме це спричинило її шалену популярність. «Python - це мова програмування загального призначення, яка використовується в широкому спектрі застосувань, включаючи веб-розробку, наукові обчислення та машинне навчання» [4].

Гідними конкурентами мові Python були і залишаються такі мови високого рівня, як Java та C/C++. Ці дві мови мають різнопланове використання: це розробка сайтів, у тому числі їх фронтенду та бекенду, розробка баз даних, ігор, і багато іншого. Однак у порівнянні з мовою Python вони є складнішими для вивчення та застосування.

Мова JavaScript широко використовується для розробки сайтів, мобільних додатків, програмування серверів. Однак цей напрям розробки не входить у наші плани.

PHP – це мова програмування, якою розроблені багато відомих веб-сайтів. Варто також зазначити, що програма WordPress – система розробки контенту сайту (CMS) – розроблена саме на PHP.

Для роботи з базами даних використовується мова SQL/MySQL. У даній кваліфікаційній роботі ми також будемо її використовувати для створення бази даних з інформацією про клієнтів.

Для оцінки популярності мов програмування складають різні рейтинги. За ними можна зробити певні висновки та здійснити вибір мови програмування. Мовою, яка швидко і динамічно розвивається є мова високого рівня C#. Вона використовує багато різнопланових фреймворків (бібліотек), які дозволяють розробляти складні програмні проекти, сайти, бази даних, мобільні додатки. З огляду на зазначені переваги мови C#, ми вибрали її для розробки нашого проекту.

C# - універсальна мова програмування для Windows, мобільних, веб-, macOS та Linux-додатків. Visual Studio - популярне середовище розробки C#. Одним словом дуже популярне середовище розробки згідно з наступними книгами [5,6,7].

1.3 Вибір середовища програмування для розробки проекту

Оскільки нами вибрано мову C# для розробки інформаційної системи пункту обміну валют, то цілком логічно як середовище програмування вибрати .NET. Ці два поняття часто ототожнюють, хоча це не зовсім правильно. То в чому полягають переваги середовища .NET?

Microsoft презентувала .NET Framework 13 січня 2000 року як стратегію Next Generation Windows Services. Ця технологія, покликана вирішити проблеми попередніх середовищ, дебютувала у 2002 році. Станом на середину 2020 року останньою версією є .NET Framework 4.8 [8].

«Платформа .NET - це сучасна платформа для розробки програмного забезпечення, яка пропонує гнучкість, продуктивність та широкий спектр

можливостей для розробників». [9]. Також важливо зазначити, що «Основою платформи .NET є загальномовне середовище виконання Common Language Runtime (CLR), завдяки чому .NET підтримує кілька мов: поряд із C# це також VB.NET, C++, F#, і навіть різні діалекти інших мов, прив'язані до .NET, наприклад, Delphi.NET. При компіляції код на будь-якій із цих мов компілюється у збірку мовою CIL (Common Intermediate Language) – свого роду асемблер платформи .NET. Тому за певних умов ми можемо зробити окремі модулі однієї програми на окремих мовах.

.NET є платформою, що переноситься (з деякими обмеженнями). Наприклад, остання версія платформи на даний момент – .NET 7 підтримується на більшості сучасних ОС Windows, MacOS, Linux. Використовуючи різні технології на платформі .NET, можна розробляти програми мовою C# для різних платформ – Windows, MacOS, Linux, Android, iOS, Tizen.

.NET представляє єдину для всіх підтримуваних мов бібліотеку класів. І який би додаток ми не збиралися писати на C# – текстовий редактор, гру, чат або складний веб-сайт – так чи інакше ми використовуємо бібліотеку класів .NET.

Загальномовне середовище виконання CLR та базова бібліотека класів є основою цілого стеку технологій, які розробники можуть задіяти під час побудови тих чи інших додатків. Наприклад, для роботи з базами даних у цьому стеку технологій призначено технологію ADO.NET та Entity Framework Core. Для побудови графічних додатків із багатим насиченим інтерфейсом – технологія WPF та WinUI, для створення більш простих графічних додатків – Windows Forms. Для розробки кросплатформних мобільних та десктопних програм – Xamarin/MAUI. Для створення веб-сайтів та веб-додатків – ASP.NET і т.д. До цього варто додати активний Blazor-фреймворк, що розвивається і набирає популярності, який працює поверх .NET і який дозволяє створювати веб-додатки як на стороні сервера, так і на стороні клієнта. А в майбутньому підтримуватиме створення мобільних додатків та, можливо, десктоп-додатків.

Згідно з результатами тестування веб-програми на .NET у ряді випадків сильно випереджають веб-програми, побудовані за допомогою інших

технологій. Програми на .NET в принципі відрізняються високою продуктивністю.

Також слід відзначити таку особливість мови C# і фреймворку .NET, як автоматичне складання сміття. А це означає, що нам здебільшого не доведеться, на відміну від C++, дбати про звільнення пам'яті. Вищезгадане загальнономовне середовище CLR саме викличе збирач сміття та очистить пам'ять» [10].

Функціональність .NET 6 ґрунтується на великій кількості керованих типів, організованих у простори імен та компоненти. Деякі з них, критичні для роботи середовища, використовуються безпосередньо CLR і знаходяться в System.Private.CoreLib.dll. До них належать вбудовані типи C#, базові класи колекцій та типи для роботи з потоками, серіалізацією, рефлексією, багатопоточністю та native-кодом [11].

РОЗДІЛ 2

ІНСТРУМЕНТИ ДЛЯ РОЗРОБКИ ІНФОРМАЦІЙНОЇ СИСТЕМИ ПУНКТУ ОБМІНУ ВАЛЮТ ЗАСОБАМИ C# WINFORMS

2.1 Компоненти Панелі елементів середовища C#WinForms

Для створення C#WinForms–проектів у фреймворці .NET використовуються як готові компоненти, доступ до яких здійснюється за допомогою вікна Toolbox (Панель елементів), так і компоненти, розроблені самостійно програмістами. Всі компоненти вікна Toolbox поділені на категорії, що знаходяться на однойменних вкладках наступного призначення:

- common controls – компоненти загального призначення;
- menus & toolbars – забезпечення операцій зі створення меню;
- containers – комоненти для розміщення на них інших компонентів;
- data – забезпечення роботи з даними;
- components – компоненти різнопланового призначення;
- printing – забезпечення операцій з друку документів;
- dialogs – організація діалогових вікнах;
- wpf interoperability – забезпечення роботи з формами WPF.

Розглянемо більш детально три категорії елементів, використання яких передбачається використовувати у даній роботі.

Вкладка Common Controls – одна з найбільших і містить такі компоненти:

- button – кнопка (як правило, командна кнопка);
- checkbox – перемикач (прапорець) для організації вибору;
- checkedlistbox – відображає об'єкт ListBox, у якому прапорці знаходяться ліворуч кожного пункту списку;
- combobox – комбінований список;
- datatimepicker – компонент для відображення дати та часу;
- label – мітка для розміщення на ній надпису;
- linklabel – додавання Web посилань;

- `listbox` – відображає список значень;
- `listview` – відображає колекцію елементів, які можуть бути виведені на екран за умови використання одного з чотирьох варіантів;
- `maskedtextbox` – задає маску вводу;
- `monthcalendar` – дозволяє вибрати дату за допомогою візуального місячного календаря;
- `notifyicon` – створення значка (іконки), який відображається у заголовку вікна;
- `numericupdown` – відображає регулятор числових значень типу «вверх – вниз»;
- `picturebox` – вікно для відображення графічних зображень;
- `progressbar` – індикатор виконання програми;
- `radiobutton` – радіокнопка для вибору одного значення з кількох;
- `richtextbox` – управління полем відформатованого тексту;
- `toolstrip` – контейнер для розміщення об'єктів панелі елементів;
- `treeview` – відображає дерево елементів проекту;
- `webbrowser` – забезпечує переміщення по Web сторінках форми.

Вкладка `Containers` містить такі компоненти:

- `flowlayoutpanel` – панель, яка динамічно розміщує контент по горизонталі та вертикалі;
- `groupbox` – відображає рамку зі заголовком навкруги групи елементів;
- `panel` – панель для розміщення інших об'єктів панелі елементів;
- `splitcontainer` – відображає рухома лінію, яка ділить контейнер на дві частини;
- `tabcontrol` – компонент, який містить зв'язаний набір сторінок-вкладок;
- `tablelayoutpanel` – компонент, у якому дані відображаються на сітці.

Вкладка `Menus & Toolbars` містить такі компоненти:

- `contextmenustrip` – забезпечує створення контекстного меню;
- `menustrip` – забезпечує створення головного меню;

- `statusstrip` – відображає стрічку стану;
- `toolstrip` – контейнер для розміщення об'єктів панелі елементів;
- `toolstripcontainer` – відображає панелі по кожній із сторін форми, а також центральну панель, на яких можуть розміщуватися інші елементи управління.

2.2 Створення об'єктів проекту та керування їх властивостями та подіями

Проект, як правило, складається з багатьох форм, але одна (головна) форма у ньому є обов'язковою (рис. 2.1). Саме на форми ми перетягуємо компоненти, за допомогою яких конструємо відповідні об'єкти і з яких конструємо проект. Тому форми самі по собі є дуже важливими об'єктами проекту.



Рисунок 2.1 – Вигляд головної форми Form1 проекту

Розглянемо більш детально властивості форми Form1 та події, зв'язані з нею. Для цього нам знадобиться вікно Properties проекту (рис. 2.2).

Нижче заголовка вікна Properties бачимо назву того компонента, який виділений у даний момент, і назву компонента (класу), конструктор якого

створює цей об'єкт. Оскільки ми виділили форму проекту Form1, то у цій стрічці буде текст зі слів Form1 та System.Windows.Forms.Form.

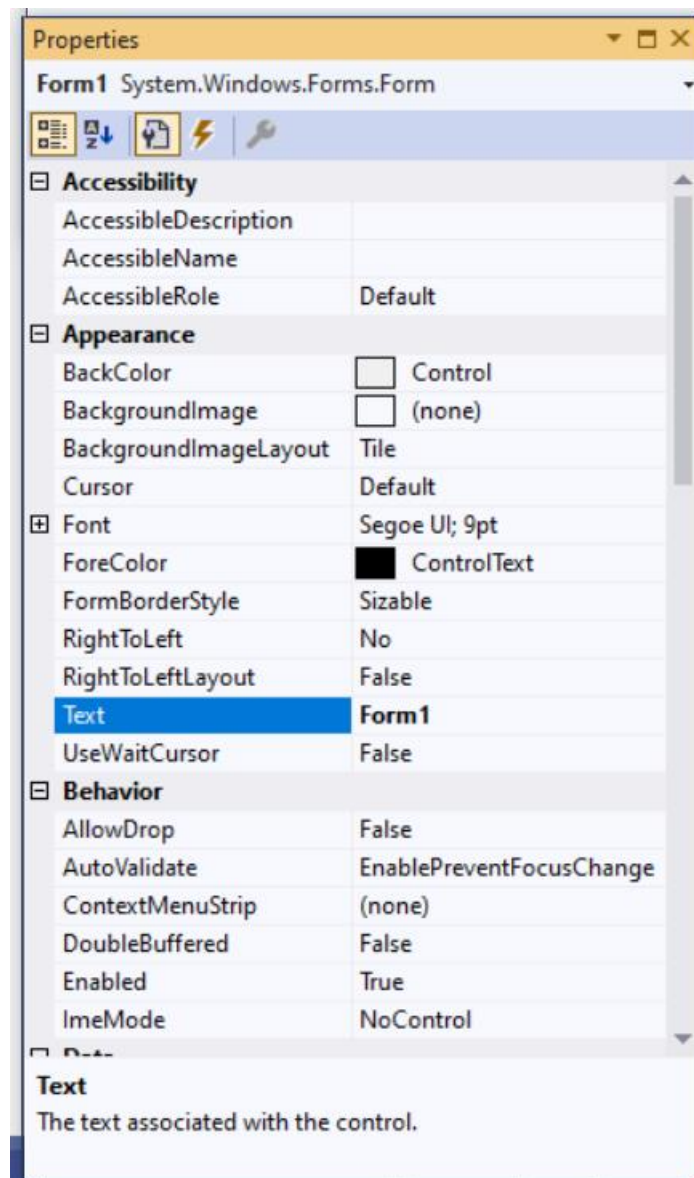


Рисунок 2.2 – Вікно Properties для задання властивостей об'єкта

У третій стрічці вікна Properties розміщено п'ять піктограм, які означають наступне:

- categorized означає, що всі елементи вікна впорядковані за категоріями;
- alphabetical означає, що всі елементи вікна впорядковані за алфавітом;
- properties означає, що у вікні розглядаються властивості об'єктів;
- events означає, що у вікні розглядаються події, пов'язані з об'єктами;

– property pages означає Сторінки властивостей.

Якщо, наприклад, зробити активними (виділити) піктограми Categorized та Properties, то це означає, що у вікні Properties всі властивості об'єкта Form1 будуть впорядкованими за категоріями. До них відносяться такі категорії:

- accesbility – доступність;
- apperance – вигляд;
- behavior – поведінка;
- data – дані;
- design – дизайн;
- focus – фокус;
- layout – за замовчуванням;
- window style – стиль вікон.

Якщо виділити піктограми Categorized та Events, то це означає, що у вікні Properties всі події, пов'язані з об'єктом Form1, будуть впорядкованими за категоріями. До них відносяться такі категорії:

- action – дії;
- apperance – вигляд;
- behavior – поведінка;
- dragdrop – перетягнути і кинути;
- design – дизайн;
- key – ключ;
- layout – за замовчуванням;
- mouse – миша;
- property changed – зміна властивостей.

Якщо клацнути по піктограмах Alphabetical та Events, то це означає, що у вікні Properties всі події, пов'язані з об'єктом Form1, будуть впорядкованими за алфавітом. Якщо ж виділити піктограми Alphabetical та Properties, то це означає, що у вікні Properties всі властивості об'єкта Form1 будуть впорядкованими за алфавітом.

Кожен об'єкт має багато властивостей, які під час його створення встановлює конструктор класу цього об'єкта. Одні властивості є однаковими для більшості об'єктів, інші – унікальні, тобто властиві тільки даному типу об'єктів. Наприклад, такими властивостями, як Name (назва об'єкта), Font (шрифт), BackColor (колір шрифту), Size (розміри) володіють більшість об'єктів, в тому числі форма Form1, іншими, наприклад, AutoSize (), володіють не всі об'єкти.

Властивості об'єктів програміст може змінювати за допомогою вікна Properties. Для цього у лівому стовпчику вікна слід виділити потрібну властивість, а у правому стовпчику вікна встановити значення цієї властивості. Є й інші способи регулювання властивостей. Наприклад, розміри вікна форми можна змінити вручну, навівши курсор на межу форми, і далі змінити розміри. На властивості об'єктів можна впливати і програмним чином. Тоді ці властивості будуть змінюватися під час виконання програми.

Щодо подій об'єкта, то для них програмісти обов'язково розробляють програми, які мають забезпечити такий функціонал, який має відбутися тоді, коли ця подія настає. Ці програми називаються обробниками подій. Наприклад, обробник події Click кнопки button1 (клацання по кнопці), наведений на рисунку 2.3, закриє форму Form1 і відкриє форму Form2.

```
private void button1_Click(object sender, EventArgs e)
{
    this.Hide();
    Form2 f2 = new Form2();
    f2.Show();}

```

Рисунок 2.3 – Обробник події Click кнопки button1

2.3 Технологія створення бази даних засобами C#WinForms

«Бази даних і, відповідно, системи управління базами даних (СУБД) займають центральне місце в сучасних інформаційних системах. Володіння відповідними знаннями та навичками є обов'язковим для спеціалістів ІТ-галузі. На сьогодні на світовому ринку розповсюджена велика кількість різномірних

СУБД, які працюють з реляційними базами даних (БД). Серед них є комерційні та вільно розповсюджені системи» [12].

Математичний формалізм алгоритмів тісно пов'язаний з мовами програмування. Мови програмування, такі як Python, Java або C++, надають спеціальні синтаксичні конструкції та елементи, що дозволяють чітко та лаконічно записувати алгоритми. Алгоритм - це чіткий план дій, який веде до вирішення задачі за певний час. Він описує послідовність кроків, які виконавець (людина, машина або комп'ютер) має виконати, щоб досягти мети. Алгоритми можна записати різними мовами, включаючи природну мову, математичні формули та мови програмування [13, 14].

Середовище C#WinForms має багатий інструментарій для створення БД. Нехай у проекті MyFirstDB потрібно створити БД Database1.mdf, яка складається лише з однієї таблиці Table з такими полями: Id – ідентифікатор запису, Name – назва товару, Price – ціна товару. Тобто, таблиця БД містить інформацію про назви товарів та їх ціни. На початку додаємо до проекту файл нової БД, що базується на службах. Клацаємо ПКМ по імені проекту MyFirstDB у вікні браузера проектів і виконуємо команду Add/Components... У вікні Add New Item, що з'являється, вибираємо вкладку Service-based Database, залишаємо стандартне ім'я файла бази даних Database1.mdf і клацаємо по кнопці Add. У браузері проектів з'явиться ім'я файла бази даних Database1.mdf.

У браузері проектів двічі клацаємо по імені файла бази даних Database1.mdf. У браузері серверів (рис. 2.4) з'явиться структура нашої БД, поки що порожньої. Клацаємо ПКМ по вкладці Table і вибираємо вкладку Add New Table. У скрипті нової таблиці можна задати нове ім'я таблиці. Ми ж залишимо його стандартним, тобто Table. Командою Save All завершуємо процес створення таблиці БД. Оновлюємо вміст БД і викликаємо таблицю Table для розробки її структури та заповнення її даними (рис. 2.5).

Поле Id таблиці Table встановлюємо ключовим, тип поля Name встановлюємо nvarchar(50), тип поля Price встановлюємо nchar(10).

Тепер наша БД розроблена. Її можна відкрити в режимі перегляду, для

того щоб ввести конкретні дані.

Створену базу необхідно зв'язати з нашим проектом. Для цього створюємо змінну-об'єкт `sqlConnection` (рис. 2.6) класу `SqlConnection`. Цій змінній присвоюємо значення стрічкової константи `connectionString`, яка визначає розташування БД на диску комп'ютера.

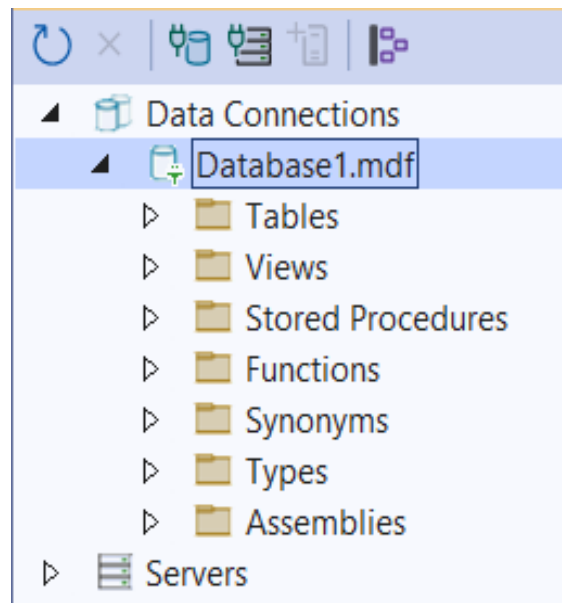


Рисунок 2.4 – Структура БД Database1.mdf

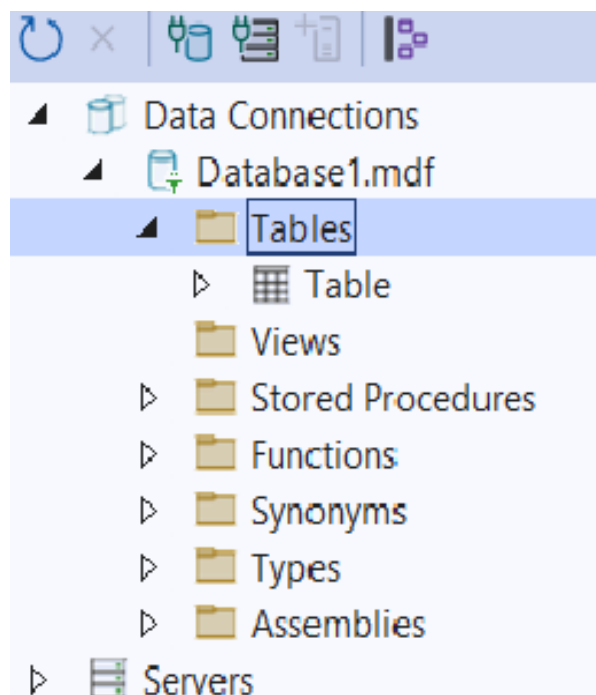


Рисунок 2.5 – Таблиця Table БД Database1

```

SqlConnection sqlConnection;

string connectionString = @"Data
Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename="C:\Users\Desktop\Kots Bohdan
2024\MyFirstDB\MyFirstDB\Database1.mdf";Integrated Security=True";

sqlConnection = new SqlConnection(connectionString);
|
SqlCommand command = new SqlCommand("SELECT * FROM [Table]", sqlConnection);

SqlCommand command = new SqlCommand("INSERT [Table] SET [Name] = @Name, [Price] =
@Price WHERE [Id] = @Id", sqlConnection);

SqlCommand command = new SqlCommand("UPDATE [Table] SET [Name] = @Name, [Price] =
@Price WHERE [Id] = @Id", sqlConnection);

SqlCommand command = new SqlCommand("DELETE FROM [Table] WHERE [Id] = @Id",
sqlConnection);

```

Рисунок 2.6 – Взаємодія БД Database1 з проектом MyFirstBD

Для роботи з БД необхідно також створити засобами C# стандартні команди-запити SELECT (вибір всіх даних для розміщення їх на формі), INSERT (вставка запису у кінець БД), UPDATE (редагування запису зі заданим Id), DELETE (вилучення запису зі заданим Id). Всі вони наведені на рисунку 2.6. У третьому розділі робота з БД розглянута детальніше.

РОЗДІЛ 3

РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ ПУНКТУ ОБМІНУ ВАЛЮТ

3.1 Структура інформаційної системи пункту обміну валют

Інформаційна система пункту обміну валют у кваліфікаційній роботі реалізована у вигляді C#WinForms-проекту, що має ім'я BachelorWork. Проект містить п'ять форм, кожна з яких є підсистемою загальної інформаційної системи і має своє функціональне призначення, наведене у таблиці 3.1.

Таблиця 3.1 – Призначення форм проекту

№ п/п	Назва форми	Призначення форми
1	Form1. Банки	Інформація про курси обміну валют у банках
2	Form2. Обмінники	Інформація про курси обміну валют в обмінниках
3	Form3. Обмін валют	Реалізація процесу обміну валют у разі ручного введення вхідних даних
4	Form4. Робота з базою даних	Реалізація операцій зі створення бази даних, яка містить інформацію про клієнтів, та роботи з нею
5	Form5. Обмін валют плюс БД	Реалізація процесу обміну валют у разі підключення ресурсів бази даних

Створення C#WinForms-проекту починається зі завантаження середовища програмування Visual Studio 2022 і появи його головного вікна (рис. 3.1). У ньому клацаємо по вкладці Create a new program, що приводить до появи наступного вікна (рис. 3.2), у якому ми вибираємо мову програмування C# та фреймворк .NET Framework для створення проекту. Клацнувши по кнопці Next, викликаємо третє вікно (рис. 1.3), у якому присвоюємо проекту ім'я BachelorWork та вибираємо папку C:\Users\Desktop\Kots Bogdan 2024 для його збереження. Остаточо клацаємо по кнопці Create і створюємо порожній проект.

Звичайно, всі згадані фреймворки мають бути встановлені під час інсталяції середовища програмування Visual Studio, або доданими до нього пізніше, якщо в цих ресурсах виникає потреба. У нашому випадку – це фреймворк .NET Framework.

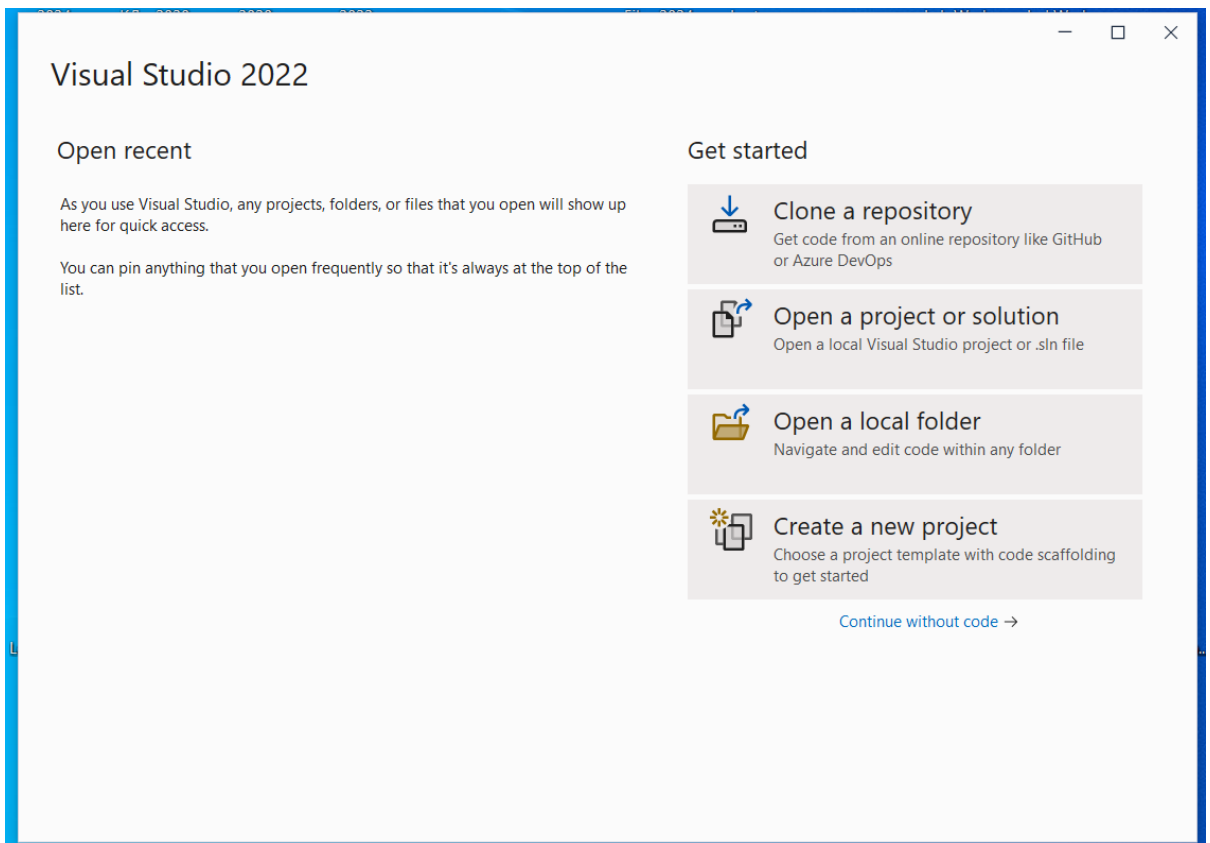


Рисунок 3.1 – Вигляд головного вікна середовища Visual Studio

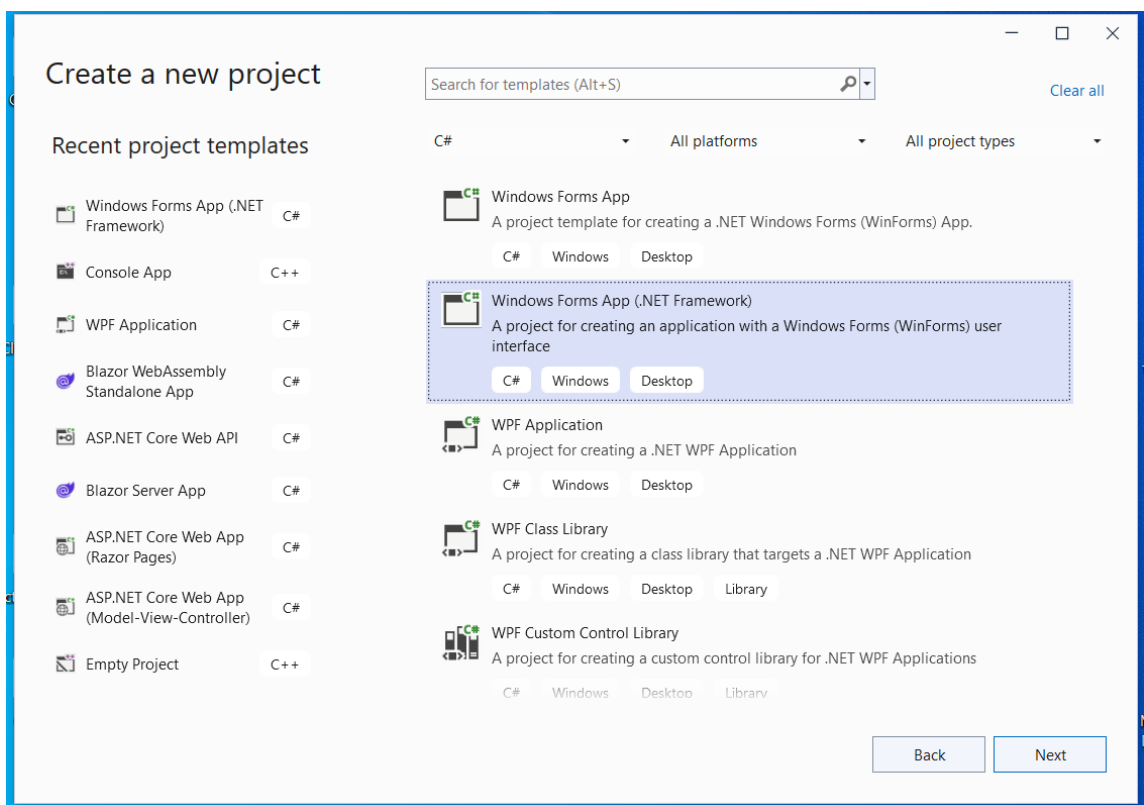


Рисунок 3.2 – Вибір мови C# та фреймворку .NET Framework для створення проекту

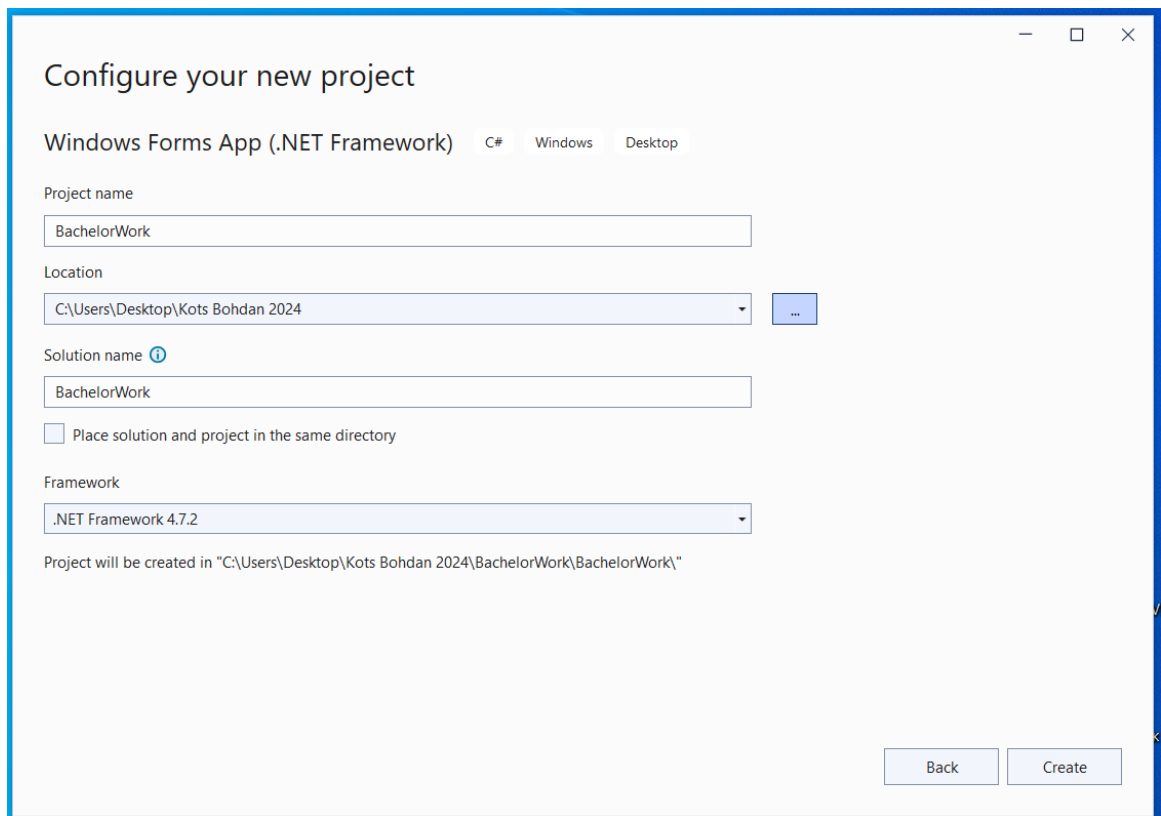


Рисунок 3.3 – Найменування проекту та вибір папки для його збереження

3.2 Створення форм проекту та забезпечення навігації по них

Вікно порожнього проекту зображене на рисунку 3.4. У ньому бачимо лише

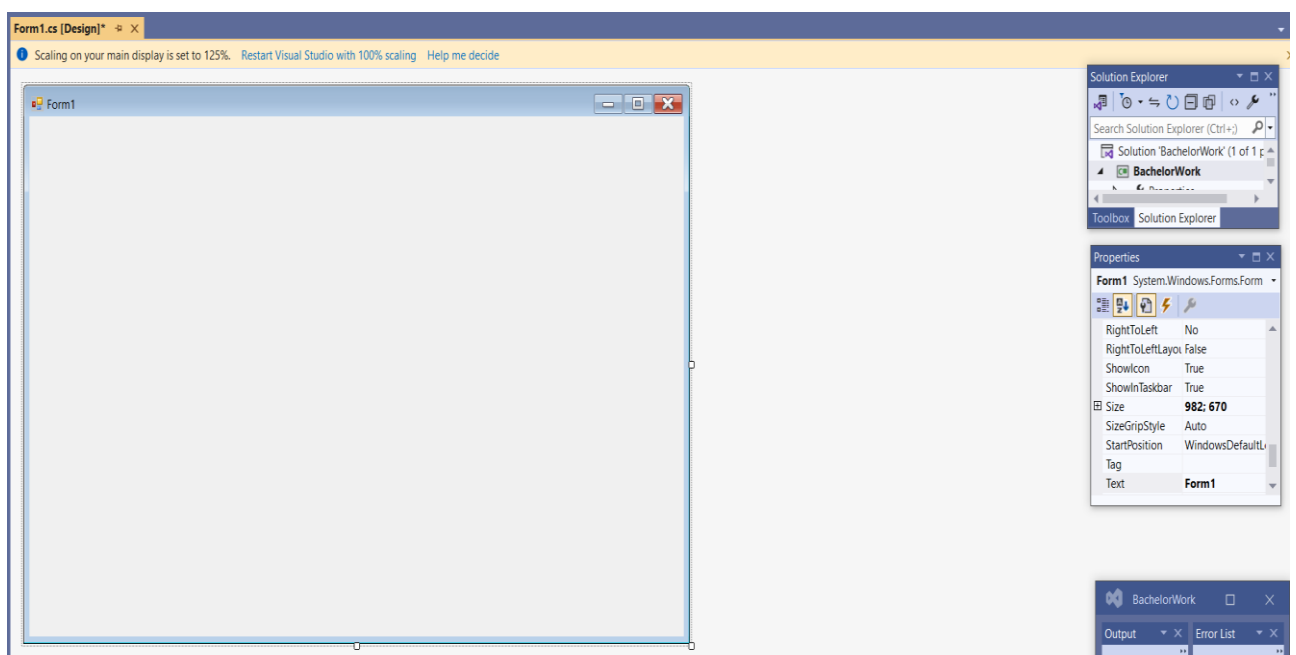


Рисунок 3.4 – Вікно порожнього проекту

форму Form1. Тому подальшу розробку продовжуємо з додавання до проекту ще чотирьох форм. Для цього клацаємо правою кнопкою миші по імені проекту BachelorWork (рис. 3.5), і у спадному меню виконуємо команду Add/Form (Windows Forms)...З'явиться вікно Add New Items (рис. 3.6), у якому вибираємо вкладку Form (Windows Forms), і клацаємо по кнопці Add. В результаті буде створена форма Form2, властивості якої за потреби ми можемо встановлювати на власний розсуд. Інші форми проекту створюються аналогічно.

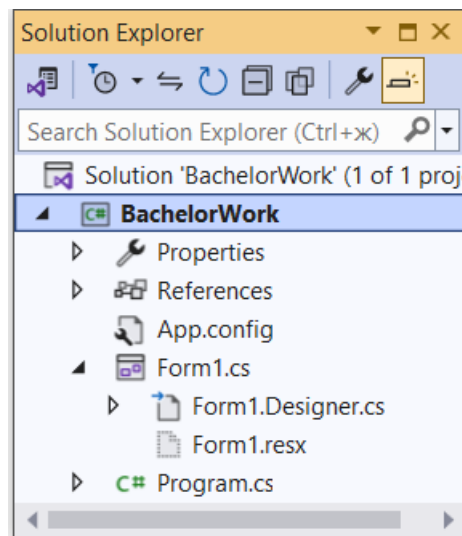


Рисунок 3.5 – Вікно браузера проектів

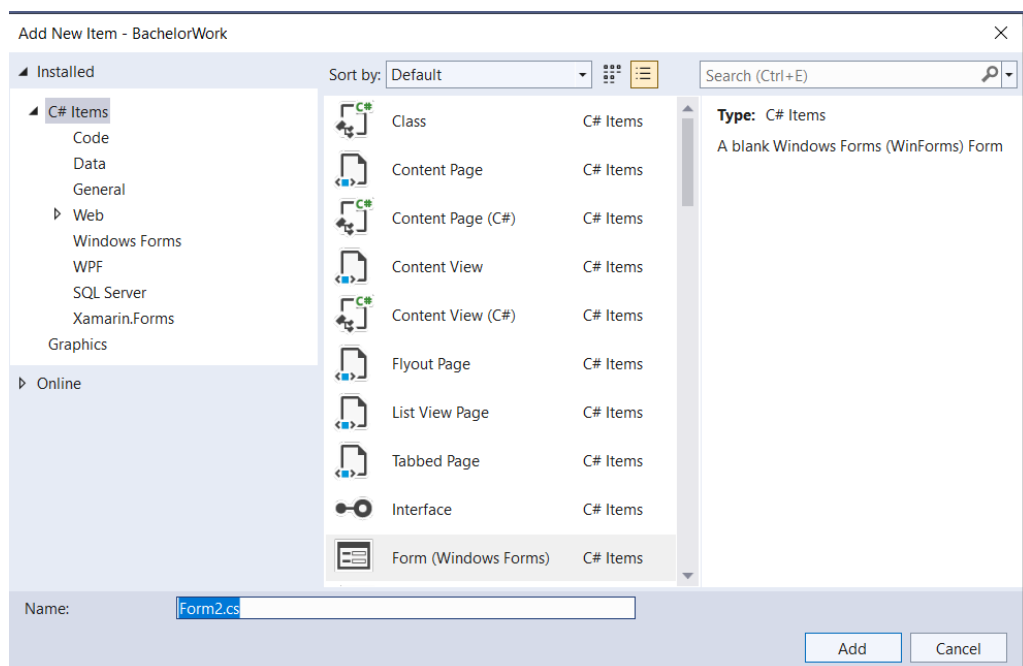


Рисунок 3.6 – Вікно додавання форм проекту

Для забезпечення навігації по формах проекту на кожній з форм передбачаємо розміщення п'яти кнопок на базі компонента Button. У вікні Toolbox (рис. 3.7) на вкладці Common Controls знаходимо компонент Button, клацаємо по ньому, і, утримуючи натисненою кнопку миші, переносимо його на форму Form1. У такий спосіб буде створено екземпляр класу Button – об'єкт button1 (рис. 3.8). Інші кнопки створюються аналогічно.

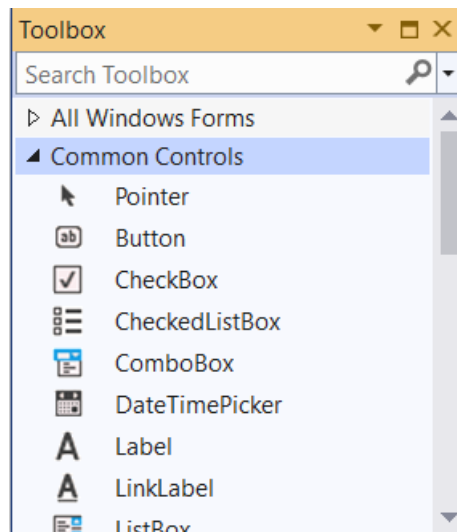


Рисунок 3.7 – Вікно панелі інструментів

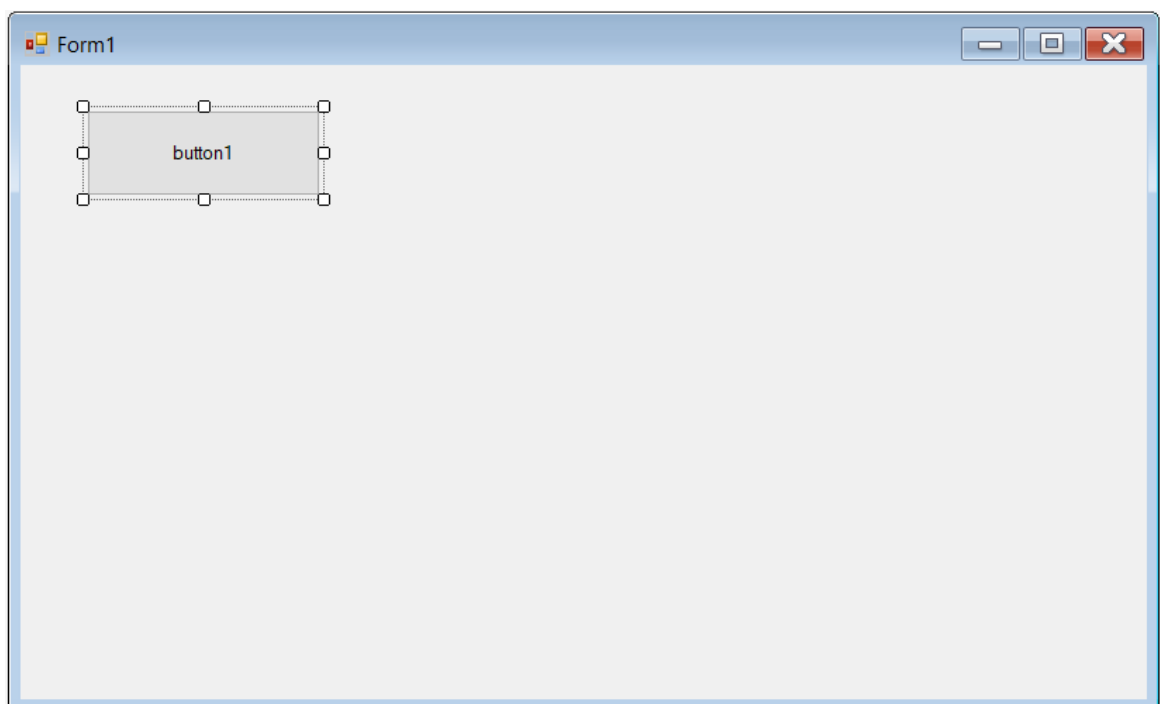


Рисунок 3.8 – Створення кнопки button1 на формі Form1

Кнопки нам потрібні для того, щоб, клацнувши по певній із них, ми могли перейти з однієї форми проекту на іншу. Для цього потрібно запрограмувати подію Click кожної кнопки. Відповідна функція називається обробником події Click цієї кнопки. У таблиці 3.2 наведені коди обробників подій Click всіх кнопок форми Form1, а у таблиці 3.3 – коди обробників подій Click всіх кнопок форми Form5. Наприклад, для переходу з форми Form1 на форму Form5 потрібно клацнути по кнопці button4 на формі Form1, а для переходу з форми Form5 на форму Form3 потрібно клацнути по кнопці button5 на формі Form5.

Таблиця 3.2 – Коди обробників події Click кнопок button форми Form1

Найменування кнопок	Властивість Text кнопки	Код обробників події Click кнопки
button1	Перейти на форму Обмінники	<code>this.Hide(); Form2 f2 = new Form2(); f2.Show();</code>
button2	Перейти на форму Обмін валют	<code>this.Hide(); Form3 f3 = new Form3(); F3.Show();</code>
button3	Перейти на форму Робота з БД	<code>this.Hide(); Form4 f4 = new Form4(); f2.Show();</code>
button4	Перейти на форму Обмін валют плюс БД	<code>this.Hide(); Form5 f5 = new f5; f5.Show();</code>
button11	Вихід	<code>Application.Exit();</code>

Таблиця 3.3 – Коди обробників події Click кнопок button форми Form5

Найменування кнопок	Властивість Text кнопки	Код обробників події Click кнопки
button1	Перейти на форму Банки	<code>this.Hide(); Form1 f1 = new Form1(); f1.Show();</code>
button2	Перейти на форму Обмінники	<code>this.Hide(); Form2 f2 = new Form2(); f2.Show();</code>
button5	Перейти на форму Обмін валют	<code>this.Hide(); Form3 f3 = new Form3(); F3.Show();</code>
button8	Перейти на форму Робота з БД	<code>this.Hide(); Form4 f4 = new f4; f4.Show();</code>
button7	Вихід	<code>Application.Exit();</code>

Клацнувши по певній кнопці button, ми закриваємо форму, на якій ця

кнопка розташована, і відкриваємо ту форму, екземпляр якої створює відповідний обробник події Click даної кнопки. За таким принципом організована навігація по всіх формах проекту. Тобто, ми можемо перейти за потреби з однієї форми проекту на будь-яку іншу.

3.3 Форма Form1 «Банки»

Форма Form1 «Банки» (рис. 3.9) є головною формою проекту у тому розумінні, що вона створена першою, а тому вона першою виконується під час запуску проекту на виконання. Однак за своїм функціоналом всі форми даного проекту рівноцінні. У таблиці 3.4 наведено склад об'єктів форми Form1 «Банки». Форма є важливим об'єктом проекту. На ній розміщуємо всі об'єкти, потрібні для вирішення тих завдань, які планується виконати за допомогою даної форми.

Завданням форми Form1 «Банки» є інформувати клієнтів про курси валют в банках м. Луцька станом на момент проведення операцій з обміном валюти.



Рисунок 3.9 – Вигляд форми Form1 «Банки»

Таблиця 3.4 – Об'єкти форми Form1 «Банки» та їх властивості

№ з/п	Ім'я об'єкта	Властивість	Значення властивості об'єкта
1	2	3	4
1	Form1	Text	Банки
2	label1	Text	КУРСИ ВАЛЮТ В БАНКІВСЬКИХ УСТАНОВАХ м. ЛУЦЬКА
3	label2	Text	USD
4	label3	Text	купівля
5	label4	Text	продаж
6	label6	Text	купівля
7	label5	Text	продаж
8	label7	Text	EUR
9	label9	Text	купівля
10	label8	Text	продаж
11	label0	Text	PLN
12	textBox1	Text	ДАТА
13	textBox2	Text	ЧАС
14	textBox3	Text	Відображення курсу купівлі USD в НБУ
15	textBox4	Text	Відображення курсу продажу USD в НБУ
16	textBox5	Text	Відображення курсу купівлі EUR в НБУ
17	textBox6	Text	Відображення курсу продажу EUR в НБУ
18	textBox7	Text	Відображення курсу купівлі PLN в НБУ
19	textBox8	Text	Відображення курсу продажу PLN в НБУ
20	textBox18	Text	Відображення курсу купівлі USD в ПриватБанку
21	textBox17	Text	Відображення курсу продажу USD в ПриватБанку
22	textBox16	Text	Відображення курсу купівлі EUR в ПриватБанку
23	textBox15	Text	Відображення курсу продажу EUR в ПриватБанку
24	textBox14	Text	Відображення курсу купівлі PLN в ПриватБанку
25	textBox13	Text	Відображення курсу продажу PLN в ПриватБанку

Продовження таблиці 3.4

1	2	3	4
26	textBox26	Text	Відображення курсу купівлі USD в Монобанку
27	textBox25	Text	Відображення курсу продажу USD в Монобанку
28	textBox24	Text	Відображення курсу купівлі EUR в Монобанку
29	textBox23	Text	Відображення курсу продажу EUR в Монобанку
30	textBox22	Text	Відображення курсу купівлі PLN в Монобанку
31	textBox21	Text	Відображення курсу продажу PLN в Монобанку
32	button1	Text	Перейти на форму Обмінники
33	button2	Text	Перейти на форму Обмін валют
34	button3	Text	Перейти на форму Робота з БД
35	button4	Text	Перейти на форму Обмін валют плюс БД
36	button11	Text	Вихід
37	button12	Text	Національний Банк України
38	button13	Text	ПриватБанк
39	button14	Text	Монобанк
40	pictureBox1	Image	Відображення прапора США
41	pictureBox2	Image	Відображення прапора Євросоюзу
42	pictureBox3	Image	Відображення прапора Польщі
43	pictureBox4	Image	Відображення прапора України

Форма Form1 «Банки» має бути запрограмована так, щоб у разі її запуску у відповідних полях об'єктів textBox відображалися поточні значення курсів валют, дати та часу. До того ж зазначимо, що ці величини мають бути доступними нам і під час запуску інших форм проекту, тобто, вони мають глобальний характер. У зв'язку з цим, виникла потреба у створенні окремого класу Rate, який опрацьовує поточні значення курсів валют (рис. 3.10).

Для додавання класу Rate клацаємо ПКМ по імені проекту BachelorWork у вікні браузера проектів і виконуємо команду Add/Class. У вікні Add New Items вибираємо вкладку Class, змінюємо назву класу Class.cs на Rate.cs і клацаємо по кнопці Add. У вікні браузера проектів помічаємо, що клас Rate.cs справді

доданий до проекту.

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace BachelorWork
{
    internal class Rate
    {
        public static double nb_rateGRNtoUSD_buy = 39.134;
        public static double nb_rateUSDtoGRN_sell = 39.134;
        public static double nb_rateGRNtoEUR_buy = 42.464;
        public static double nb_rateEURtoGRN_sell = 42.464;
        public static double nb_rateGRNtoPLN_buy = 9.826;
        public static double nb_ratePLNtoGRN_sell = 9.826;
        public static double pb_rateGRNtoUSD_buy = 38.800;
        public static double pb_rateUSDtoGRN_sell = 39.400;
        public static double pb_rateGRNtoEUR_buy = 42.000;
        public static double pb_rateEURtoGRN_sell = 43.00;
        public static double mb_rateGRNtoUSD_buy = 39.050;
        public static double mb_rateUSDtoGRN_sell = 39.350;
        public static double mb_rateGRNtoEUR_buy = 42.350;
        public static double mb_rateEURtoGRN_sell = 42.880;
        public static double va_rateGRNtoUSD_buy = 39.010;
        public static double va_rateUSDtoGRN_sell = 39.490;
        public static double va_rateGRNtoEUR_buy = 42.500;
        public static double va_rateEURtoGRN_sell = 42.990;
        public static double va_rateGRNtoPLN_buy = 9.750;

        public static double va_ratePLNtoGRN_sell = 10.000;
        public static double lu_rateGRNtoUSD_buy = 39.000;
        public static double lu_rateUSDtoGRN_sell = 39.350;
        public static double lu_rateGRNtoEUR_buy = 42.600;
        public static double lu_rateEURtoGRN_sell = 43.000;
        public static double lu_rateGRNtoPLN_buy = 9.800;
        public static double lu_ratePLNtoGRN_sell = 9.950;
        public static double gr_rateGRNtoUSD_buy = 39.000;
        public static double gr_rateUSDtoGRN_sell = 39.250;
        public static double gr_rateGRNtoEUR_buy = 42.600;
        public static double gr_rateEURtoGRN_sell = 43.000;
        public static double gr_rateGRNtoPLN_buy = 9.800;
        public static double gr_ratePLNtoGRN_sell = 9.950;
    }
}

```

Рисунок 3.10 – Код класу Rate

Тепер ми можемо використати поточні значення курсів валют у кожній формі проекту, у тому числі і в формі Form1, фрагмент коду якої наведено на рисунку 3.11. Запустивши форму Form1 «Банки» на виконання, ми пересвідчуємося, що на ній коректно відображаються поточні значення курсів валют, дати і часу (рис. 3.12). Ознайомившись з ними, клієнт може оцінити вигідні для себе варіанти обміну валют і приступити до їх реалізації.

```
public Form1()
{
    InitializeComponent();
    date1 = DateTime.Today;
    textBox1.Text = date1.ToString("dd:MM:yyyy");
    date2 = DateTime.Now;
    textBox2.Text = date2.ToString("hh:mm");
    textBox3.Text = Rate.nb_rateGRNtoUSD_buy.ToString("00.000");
    textBox4.Text = Rate.nb_rateUSDtoGRN_sell.ToString("00.000");
    textBox5.Text = Rate.nb_rateGRNtoEUR_buy.ToString("00.000");
    textBox6.Text = Rate.nb_rateEURtoGRN_sell.ToString("00.000");
    textBox7.Text = Rate.nb_rateGRNtoPLN_buy.ToString("0.000");
    textBox8.Text = Rate.nb_ratePLNtoGRN_sell.ToString("0.000");
    textBox18.Text = Rate.pb_rateGRNtoUSD_buy.ToString("00.000");
    textBox17.Text = Rate.pb_rateUSDtoGRN_sell.ToString("00.000");
    textBox16.Text = Rate.pb_rateGRNtoEUR_buy.ToString("00.000");
    textBox15.Text = Rate.pb_rateEURtoGRN_sell.ToString("00.000");
    textBox26.Text = Rate.mb_rateGRNtoUSD_buy.ToString("0.000");
    textBox25.Text = Rate.mb_rateUSDtoGRN_sell.ToString("0.000");
    textBox24.Text = Rate.mb_rateGRNtoEUR_buy.ToString("0.000");
    textBox23.Text = Rate.mb_rateEURtoGRN_sell.ToString("0.000");}
}
```

Рисунок 3.11 – Фрагмент коду форми Form1 «Банки»

Стосовно курсів валют, зауважимо наступне. У деяких банках чи обмінниках певні курси валют відсутні. Наприклад, ПриватБанк та Монобанк не пропонують цієї послуги. Це має бути враховано під час розробки проекту.

Значення курсів валют періодично міняються протягом дня. Тому доцільно зчитувати у клас Rate їх значення з відповідних сайтів, але хай це буде завданням на майбутнє.

Form1. Банки

КУРСИ ВАЛЮТ У БАНКІВСЬКИХ УСТАНОВАХ м. ЛУЦЬКА

	USD	EUR	PLN			
02:04:2024	Купівля	Продаж	Купівля	Продаж	Купівля	Продаж
10:56						
Національний Банк України	39,134	39,134	42,464	42,464	9,826	9,826
ПриватБанк	38,800	39,400	42,000	43,000		
Монобанк	39,050	39,350	42,350	42,880		

[Перейти на форму Обмінники](#)
[Перейти на форму Робота з БД](#)
[Перейти на форму Обмін валют](#)
[Вихід](#)

Рисунок 3.12 – Поточні значення курсів валют на формі Form1 «Банки»

3.4 Форма Form2 «Обмінники»

Форма Form2 «Обмінники» (рис. 3.13) розроблена аналогічно до попередньої. На рисунку 3.14 наведені результати її роботи.

На відміну від банків, всі обмінники м. Луцька пропонують обмін гривень на злоті та навпаки. Тому такий обмін можна зробити через обмінники.

Зауважимо також, що коли виникає потреба поміняти долари на євро, злоті на фунти або ж виконати подібні транзакції, то напряду цього зробити не можна. Потрібно спочатку поміняти долари на гривні, а тоді гривні на євро. Тобто всі транзакції здійснюються через гривню, як основну валюту. Звичайно, під час подвійного обміну клієнт двічі втрачає, але така політика всіх банків, і з цим нічого вдіяти не можна. Крім наведених у роботі валют, банки та обмінники можуть працювати і з іншими видами валют.

Рисунок 3.13 – Вигляд форми Form2 «Обмінники»

Банк	USD Купівля	USD Продаж	EUR Купівля	EUR Продаж	PLN Купівля	PLN Продаж
Національний Банк України	39,134	39,134	42,464	42,464	9,826	9,826
ПриватБанк	38,800	39,400	42,000	43,000		
Монобанк	39,050	39,350	42,350	42,880		

Рисунок 3.14 – Результати роботи форми Form2 «Обмінники»

3.5 Форма Form3 «Обмін валют»

Форма Form3 «Обмін валют» (рис. 3.15) призначена для виконання операцій з обміну валют. Вихідними даними у цій задачі є обсяги кожного виду

валюти (гривні, долари, євро, злоті) клієнта, які є у наявності і можуть бути конвертовані в інші види валют. Результатом роботи цієї форми є обсяги кожного виду валюти (гривні, долари, євро, злоті) клієнта, які отримані в процесі виконання дозволених транзакцій.

Рисунок 3.15 – Вигляд форми Form3 «Обмін валют»

Склад об'єктів форми Form3 «Обмін валют» та їх властивості наведено у таблиці 3.5. Розмістивши всі об'єкти на формі Form3 «Обмін валют», приступаємо до процесу її програмування. Перш за все оголошуємо та ініціалізуємо змінні для роботи з курсами валют (рис. 3.16 – 3.18). Тут необхідно врахувати всі курси валют, які виникають при конвертації гривні у долари, євро та злоті і навпаки. Звісно, курси валют, як правило, є різними у різних банках та обмінниках, тому змінних у кодї так багато. Оголошеним у формі Form3 змінним присвоюємо значення відповідних змінних раніше оголошеного класу Rate.

Звісно, банки та обмінники можуть пропонувати проведення операцій обміну валют і з іншими валютами, наприклад, фунтами, юанями і т.д. Та й банків та обмінників у м. Луцьку є значно більше. У кваліфікаційній роботі ми свідомо вибрали лише чотири типи валют, три банки та три обмінники.

Таблиця 3.5 – Об'єкти форми Form3 «Обмін валют» та їх властивості

№ з/п	Ім'я об'єкта	Властивість	Значення властивості об'єкта
1	2	3	4
1	Form3	Text	Обмін валют
2	label1	Text	ВИКОНАННЯ ТА АНАЛІЗ РЕЗУЛЬТАТІВ ОПЕРАЦІЙ З ОБМІНУ ВАЛЮТ
3	label2	Text	1. Введіть обсяги валют, яка у Вас є в наявності (через кому)
4	label3	Text	3. Перед виконанням операцій обміну валют у Вас є в наявності:
5	label16	Text	9. Після виконання операцій обміну валют у Вас є в наявності:
6	label7	Text	7. Введіть суму валюти, яка підлягає обміну:
7	label4	Text	GRN
8	label10	Text	GRN
9	label15	Text	GRN
10	label7	Text	USD
11	label8	Text	USD
12	label14	Text	USD
13	label	Text	EUR
14	label	Text	EUR
15	label	Text	EUR
16	label	Text	PLN
17	label	Text	PLN
18	label	Text	PLN
19	textBox1	Text	Відображення суми в GRN
20	textBox5	Text	Відображення суми в GRN
21	textBox9	Text	Відображення суми в GRN
22	textBox2	Text	Відображення суми в USD
23	textBox6	Text	Відображення суми в USD
24	textBox10	Text	Відображення суми в USD
25	textBox3	Text	Відображення суми в EUR
26	textBox7	Text	Відображення суми в EUR
27	textBox11	Text	Відображення суми в EUR
28	textBox4	Text	Відображення суми в PLN
29	textBox8	Text	Відображення суми в PLN
30	textBox12	Text	Відображення суми в PLN
31	textBox13	Text	Відображення суми валюти, яка підлягає обміну
32	button1	Text	Перейти на форму Банки

Продовження таблиці 3.5

1	2	3	4
33	button2	Text	Перейти на форму Обмінники
34	button5	Text	Перейти на форму Робота з БД
35	button8	Text	Перейти на форму Обмін валют плюс БД
36	button11	Text	Вихід
37	button4	Text	Перейти на форму Обмін валют плюс БД
38	button7	Text	Вихід
39	button3	Text	8. Натисніть тут, щоб виконати операцію обміну валюти!
40	button4	Text	н
41	button6	Text	11. Натисніть тут, щоб продовжити чи завершити!
42	groupBox1	Text	6. Виберіть банк чи обмінний пункт:
43	groupBox2	Text	4. Яку валюту хочете поміняти:
44	groupBox3	Text	5. На яку валюту хочете поміняти:
45	groupBox4	Text	10. Хочете продовжити?
46	radioButton1	Text	Національний Банк України
47	radioButton2	Text	ПриватБанк
48	radioButton3	Text	Монобанк
49	radioButton4	Text	Варшавка
50	radioButton5	Text	Luchesk
51	radioButton6	Text	Grand Volyn
52	radioButton7	Text	GRN
53	radioButton8	Text	USD
54	radioButton9	Text	EUR
55	radioButton10	Text	PLN
56	radioButton11	Text	GRN
57	radioButton12	Text	USD
58	radioButton13	Text	EUR
59	radioButton14	Text	PLN

Після встановлення перерахованих у таблиці 3.5 об'єктів, значення їх властивостей мають бути вибрані за допомогою вікна Properties, щоб забезпечити відповідний вигляд, та запрограмовані, щоб забезпечити відповідний відповідний функціонал. Тут мова йде про обробники різних подій, які власне і повинні забезпечувати цей функціонал. У подальшому ми будемо звертати увагу

саме на розробку цього функціоналу.

```
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using static System.Windows.Forms.VisualStyles.VisualStyleElement;
using static System.Windows.Forms.VisualStyles.VisualStyleElement.Button;
namespace BachelorWork
{
    public partial class Form3 : Form
    {
        //-----
        double nb_rateUSDtoGRN = Rate.nb_rateUSDtoGRN_sell; // Нац Банк. Продати 1 долар на гривні
        double nb_rateGRNtoUSD = Rate.nb_rateGRNtoUSD_buy; // Нац Банк. Купити 1 долар за гривні
        double pb_rateUSDtoGRN = Rate.pb_rateUSDtoGRN_sell; // Приват Банк. Продати 1 долар на гривні
        double pb_rateGRNtoUSD = Rate.pb_rateGRNtoUSD_buy; // Приват Банк. Купити 1 долар за гривні
        double mb_rateUSDtoGRN = Rate.mb_rateUSDtoGRN_sell; // Моно Банк. Продати 1 долар на гривні
        double mb_rateGRNtoUSD = Rate.mb_rateGRNtoUSD_buy; // Моно Банк. Купити 1 долар за гривні
        double va_rateUSDtoGRN = Rate.va_rateUSDtoGRN_sell; // Варшавка. Продати 1 долар на гривні
        double va_rateGRNtoUSD = Rate.va_rateGRNtoUSD_buy; // Варшавка. Купити 1 долар за гривні
        double lu_rateUSDtoGRN = Rate.lu_rateUSDtoGRN_sell; // Лучеськ. Продати 1 долар на гривні
        double lu_rateGRNtoUSD = Rate.lu_rateGRNtoUSD_buy; // Лучеськ. Купити 1 долар за гривні
        double gr_rateUSDtoGRN = Rate.gr_rateUSDtoGRN_sell; // Гр. Волинь. Продати 1 долар на гривні
        double gr_rateGRNtoUSD = Rate.gr_rateGRNtoUSD_buy; // Гр. Волинь. Купити 1 долар за гривні
```

Рисунок 3.16 – Оголошення та ініціалізація змінних класу Form3

```
//-----
double nb_rateEURtoGRN = Rate.nb_rateEURtoGRN_sell; // Нац Банк. Продати 1 євро на гривні
double nb_rateGRNtoEUR = Rate.nb_rateGRNtoEUR_buy; // Нац Банк. Купити 1 євро за гривні
double pb_rateEURtoGRN = Rate.pb_rateEURtoGRN_sell; // Приват Банк. Продати 1 євро на гривні
double pb_rateGRNtoEUR = Rate.pb_rateGRNtoEUR_buy; // Приват Банк. Купити 1 євро за гривні
double mb_rateEURtoGRN = Rate.mb_rateEURtoGRN_sell; // Моно Банк. Продати 1 євро на гривні
double mb_rateGRNtoEUR = Rate.mb_rateGRNtoEUR_buy; // Моно Банк. Купити 1 євро за гривні
double va_rateEURtoGRN = Rate.va_rateEURtoGRN_sell; // Варшавка. Продати 1 євро на гривні
double va_rateGRNtoEUR = Rate.va_rateGRNtoEUR_buy; // Варшавка. Купити 1 євро за гривні
double lu_rateEURtoGRN = Rate.lu_rateEURtoGRN_sell; // Лучеськ. Продати 1 євро на гривні
double lu_rateGRNtoEUR = Rate.lu_rateGRNtoEUR_buy; // Лучеськ. Купити 1 долар за гривні
double gr_rateEURtoGRN = Rate.gr_rateEURtoGRN_sell; // Гранд Волинь. Продати 1 євро на гривні
double gr_rateGRNtoEUR = Rate.gr_rateGRNtoEUR_buy; // Гранд Волинь. Купити 1 євро за гривні
```

Рисунок 3.17 – Оголошення та ініціалізація змінних класу Form3

```
//-----
double nb_ratePLNtoGRN = Rate.nb_ratePLNtoGRN_sell; // Нац Банк. Продати 1 злот на гривні
double nb_rateGRNtoPLN = Rate.nb_rateGRNtoPLN_buy; // Нац Банк. Купити 1 злот за гривні
double pb_ratePLNtoGRN = Rate.pb_ratePLNtoGRN_sell; // Приват Банк. Продати 1 злот на гривні
double pb_rateGRNtoPLN = Rate.pb_rateGRNtoPLN_buy; // Приват Банк. Купити 1 злот за гривні
double mb_ratePLNtoGRN = Rate.mb_ratePLNtoGRN_sell; // Моно Банк. Продати 1 злот на гривні
double mb_rateGRNtoPLN = Rate.mb_rateGRNtoPLN_buy; // Моно Банк. Купити 1 злот за гривні
double va_ratePLNtoGRN = Rate.va_ratePLNtoGRN_sell; // Варшавка. Продати 1 злот на гривні
double va_rateGRNtoPLN = Rate.va_rateGRNtoPLN_buy; // Варшавка. Купити 1 злот за гривні
double lu_ratePLNtoGRN = Rate.lu_ratePLNtoGRN_sell; // Лучеськ. Продати 1 злот на гривні
double lu_rateGRNtoPLN = Rate.lu_rateGRNtoPLN_buy; // Лучеськ. Купити 1 злот за гривні
double gr_ratePLNtoGRN = Rate.gr_ratePLNtoGRN_sell; // Гранд Волинь. Продати 1 злот на гривні
double gr_rateGRNtoPLN = Rate.gr_rateGRNtoPLN_buy; // Гранд Волинь. Купити 1 злот за гривні
```

Рисунок 3.18 – Оголошення та ініціалізація змінних класу Form3

Крім змінних для роботи з курсами валют, у класі Form3 оголошуємо також змінні для роботи з датою та часом, а також робочі змінні, які ми будемо використовувати для програмування процесу обміну валют (рис. 3.19).

```
//-----
DateTime date1 = new DateTime();
DateTime date2 = new DateTime();
double grnN;
double usdN;
double eurN;
double plnN;
double rate, rez;
double ExchangeSum;
double grnLim = 40; // Мінімальна сума обміну у гривнях
double usdLim = 10; // Мінімальна сума обміну у доларах
double eurLim = 10; // Мінімальна сума обміну у євро
double plnLim = 20; // Мінімальна сума обміну у злотих
bool v = true; // Перша операція обміну валют дозволена
public Form3()
{ InitializeComponent();
  textBox13.Text = "0"; }
```

Рисунок 3.19 – Оголошення та ініціалізація змінних класу Form3

Запустивши проект на виконання, ми викликаємо на екран форму Form1, з якої викликаємо форму Form3. Мітка label2 виводить текст «1. Введіть обсяги валют, яка у Вас є в наявності (через кому)». Вводимо значення обсягів відповідних валют у вікна об'єктів textBox1, textBox2, textBox3, textBox4 і натискаємо кнопку button4, на якій відображається надпис «2. Натисніть цю кнопку, щоб зберегти обсяги валюти, які є у Вас в наявності!», що ми й виконуємо. Мітка label3 виводить текст «3. Перед виконанням операцій обміну валют у Вас є в наявності:». У вікнах об'єктів textBox5, textBox6, textBox7, textBox8 будуть відображені обсяги валют, які є в наявності і які можна використовувати для проведення операцій обміну валют. Код обробника події Click кнопки button4, який забезпечує цей функціонал, наведено на рисунку 3.20. З цього моменту ми вже готові виконувати операції обміну валют, які на формі Form3 позначені пунктами від 4 до 11.

```
// Введення та збереження обсягів валюти, яка є в наявності
private void button4_Click(object sender, EventArgs e)
{
    grnN = Double.Parse(textBox1.Text);
    usdN = Double.Parse(textBox2.Text);
    eurN = Double.Parse(textBox3.Text);
    plnN = Double.Parse(textBox4.Text);
    textBox5.Text = grnN.ToString("0.00");
    textBox6.Text = usdN.ToString("0.00");
    textBox7.Text = eurN.ToString("0.00");
    textBox8.Text = plnN.ToString("0.00");
    radioButton18.Checked = true;
}
```

Рисунок 3.20 – Введення та збереження значень обсягів валют, що є в клієнта

Виконання операції обміну валют починаємо з виконання пунктів 4 – 6 форми. Тобто, ми вибираємо:

- вид валюти, яку міняємо;
- вид валюти, на яку міняємо;
- банк чи обмінник, у якому здійснюємо обмін.

Для цього потрібно активізувати ту чи іншу кнопку radioButton у кожній з трьох груп кнопок. Виконання транзакції забезпечує обробник події Click кнопки button3, початок коду якого наведено на рисунку 3.21. Отже, перш, ніж виконати операцію обміну, ми перевіряємо, чи дозволена взагалі ця операція. Зауважимо, що можна поміняти лише гривні на долари, євро та злоті або навпаки. Всі інші варіанти обміну заборонені. Наприклад, не можна поміняти долари на євро. Позбавлена смислу операція обміну гривень на гривні. Крім того, заборонені і такі транзакції, які не передбачені в даному банку чи обміннику.

```
// Виконання операцій обміну валют
private void button3_Click(object sender, EventArgs e)
{ ExchangeSum = Double.Parse(textBox13.Text);
  // Перевірка, чи не вибрано заборонені транзакції
  if ((radioButton7.Checked && radioButton11.Checked) ||
      (radioButton8.Checked && radioButton12.Checked) || (radioButton9.Checked && radioButton13.Checked)
      || (radioButton10.Checked && radioButton14.Checked)
      || (radioButton8.Checked && radioButton12.Checked) || (radioButton8.Checked &&
radioButton13.Checked) || (radioButton8.Checked && radioButton14.Checked)
      || (radioButton9.Checked && radioButton12.Checked) || (radioButton9.Checked &&
radioButton13.Checked) || (radioButton9.Checked && radioButton14.Checked)
      || (radioButton10.Checked && radioButton12.Checked) || (radioButton10.Checked &&
radioButton13.Checked) || (radioButton10.Checked && radioButton14.Checked)
      || (radioButton7.Checked && radioButton14.Checked && radioButton2.Checked) || (radioButton7.Checked
&& radioButton14.Checked && radioButton3.Checked)
      || (radioButton10.Checked && radioButton11.Checked && radioButton2.Checked) ||
(radioButton10.Checked && radioButton11.Checked && radioButton3.Checked))
  { MessageBox.Show("Вибрана Вами транзакція не передбачена!", "Помилка вводу даних або вибору
опцій", MessageBoxButtons.OK, MessageBoxIcon.Information);
    MessageBox.Show("Увага! Правильно виберіть види валют та банк чи обмінник!", "Помилка вводу
даних або вибору опцій", MessageBoxButtons.OK, MessageBoxIcon.Information);
    MessageBox.Show("Виконайте пункт 10, а потім пункт 11 даного додатку!", "Помилка вводу даних або
вибору опцій", MessageBoxButtons.OK, MessageBoxIcon.Information);
    v = false; // Операція обміну валют не може бути виконана }
}
```

Рисунок 3.21 – Перевірка, чи не є вибрана транзакція забороненою

Наприклад, ПриваБанк та Монобанк не працюють зі злотими, тому відповідні варіанти обміну варто розглядати, як заборонені. Всі ці нюанси враховані у наведеному на рисунку 3.21 коді. Однак, на цьому перевірки не закінчилися, як і код обробника.

Далі ми виконуємо ще ряд перевірок, які полягають у тому, чи сума валюти, яка заявлена для обміну, не менша за мінімально дозволена величину і чи ця сума не перевищує наявний обсяг відповідної валюти. Ці перевірки здійснюються відносно кожного виду валюти:

- гривні (рис. 3.22);
- долара (рис. 3.23);
- євро (рис. 3.24);
- злотого (рис. 3.25).

Варто зазначити, що передбачити всі можливі ситуації, за яких операція обміну валют втрачає смисл, важко. Вони можуть бути виявлені лише в процесі довготривалого тестування програми і внесення відповідних поправок. Тут корисною може бути співпраця користувачів та розробників програми.

У разі успішного проходження усіх перевірок, приступаємо до власне виконання операцій обміну валют.

```
// Перевірка, чи сума гривень, яка підлягає обміну, не менша за мінімальну дозволена і не більша за
наявну суму
else if (radioButton7.Checked && ((ExchangeSum <= grnLim) || (ExchangeSum > grnN)))
{
    {MessageBox.Show("Сума гривень, яка підлягає обміну, менша за мінімальну дозволена або перевищує
наявну!", "Помилка вводу даних", MessageBoxButtons.OK, MessageBoxIcon.Information);
        MessageBox.Show("Увага! Правильно введіть суму гривень, яка підлягає обміну!", "Помилка вводу
даних", MessageBoxButtons.OK, MessageBoxIcon.Information);
        MessageBox.Show("Виконайте пункт 10, а потім пункт 11 даного додатку!", "Помилка вводу даних",
MessageBoxButtons.OK, MessageBoxIcon.Information); }
    v = false; // Операція обміну валют не може бути виконана } }
```

Рисунок 3.22 – Перевірка, чи є сума гривень, яка підлягає обміну, допустимою

```

// Перевірка, чи сума доларів, яка підлягає обміну, не менша за мінімальну дозволена і не більша за
наявну суму
else if (radioButton8.Checked && ((ExchangeSum <= usdLim) || (ExchangeSum > usdN)))
{ { MessageBox.Show("Сума доларів, яка підлягає обміну, менша за мінімальну дозволена або перевищує
наявну!", "Помилка вводу даних", MessageBoxButtons.OK, MessageBoxIcon.Information);
    MessageBox.Show("Увага! Правильно введіть суму доларів, яка підлягає обміну!", "Помилка вводу
даних", MessageBoxButtons.OK, MessageBoxIcon.Information);
    MessageBox.Show("Виконайте пункт 10, а потім пункт 11 даного додатку!", "Помилка вводу даних",
MessageBoxButtons.OK, MessageBoxIcon.Information); }
    v = false; // Операція обміну валют не може бути виконана }

```

Рисунок 3.23 – Перевірка, чи є сума доларів, яка підлягає обміну, допустимою

```

// Перевірка, чи сума євро, яка підлягає обміну, не менша за мінімальну дозволена і не більша за наявну
суму
else if (radioButton9.Checked && ((ExchangeSum <= eurLim) || (ExchangeSum > eurN)))
{ { MessageBox.Show("Сума євро, яка підлягає обміну, менша за мінімальну дозволена або перевищує
наявну!", "Помилка вводу даних", MessageBoxButtons.OK, MessageBoxIcon.Information);
    MessageBox.Show("Увага! Правильно введіть суму євро, яка підлягає обміну!", "Помилка вводу
даних", MessageBoxButtons.OK, MessageBoxIcon.Information);
    MessageBox.Show("Виконайте пункт 10, а потім пункт 11 даного додатку!", "Помилка вводу даних",
MessageBoxButtons.OK, MessageBoxIcon.Information); }
    v = false; // Операція обміну валют не може бути виконана }

```

Рисунок 3.24 – Перевірка, чи є сума євро, яка підлягає обміну, допустимою

```

// Перевірка, чи сума злотих, яка підлягає обміну, не менша за мінімальну дозволена і не більша за наявну суму
else if (radioButton10.Checked && ((ExchangeSum <= plnLim) || (ExchangeSum > plnN)))
{ { MessageBox.Show("Сума злотих, яка підлягає обміну, менша за мінімальну дозволена або перевищує
наявну!", "Помилка вводу даних", MessageBoxButtons.OK, MessageBoxIcon.Information);
    MessageBox.Show("Увага! Правильно введіть суму злотих, яка підлягає обміну!", "Помилка вводу
даних", MessageBoxButtons.OK, MessageBoxIcon.Information);
    MessageBox.Show("Виконайте пункт 10, а потім пункт 11 даного додатку!", "Помилка вводу даних",
MessageBoxButtons.OK, MessageBoxIcon.Information); }
    v = false; // Операція обміну валют не може бути виконана }

```

Рисунок 3.25 – Перевірка, чи є сума злотих, яка підлягає обміну, допустимою

На рисунку 3.26 наведено фрагмент коду, який забезпечує виконання операції обміну гривень на долари через ПриватБанк. Спочатку перевіряється

умова, чи ми дійсно вибрали Приватбанк у якості місця обміну (`radioButton2.Checked`), чи ми дійсно хочемо поміняти гривні (`radioButton7.Checked`) на долари (`radioButton12.Checked`) і чи взагалі операція обміну валют є доступною у даний момент (`v == true`). Якщо так, то виконується операція обміну валют, у якій сума гривень `ExchangeSum`, заявлена для обміну, перераховується за відповідним курсом обміну `rate = pb_rateGRNtoUSD`, а отриманий результат додається до значення `usdN` і віднімається від `grnN`.

```
// Обмін гривні на долари через ПриватБанк
else if (radioButton2.Checked && radioButton7.Checked && radioButton12.Checked && v)
{ rate = pb_rateGRNtoUSD;
  rez = ExchangeSum / rate;
  grnN -= ExchangeSum;
  usdN += rez;
  textBox9.Text = grnN.ToString("0.00");
  textBox10.Text = usdN.ToString("0.00");
  textBox11.Text = textBox7.Text;
  textBox12.Text = textBox8.Text; }
```

Рисунок 3.26 – Обмін гривні на долари через ПриватБанк

Поточні значення євро та злотих при цьому не змінюються. Відповідні значення заносяться у відповідні поля об'єктів `textBox9`, `textBox10`, `textBox11`, `textBox12`.

Операції обміну гривень на євро та злоті виконуються аналогічно.

На рисунку 3.27 наведено фрагмент коду, який забезпечує виконання операції обміну доларів на гривні через ПриватБанк. Легко бачити, що він подібний до попереднього, різниця лише у формулах для перерахунку. У даному випадку обсяг доларів зменшиться, обсяг гривень збільшиться, а поточні значення євро та злотих залишаться попередніми.

Операції обміну євро та злотих на гривні виконуються аналогічно. Інші операції обміну валют вважаються недозволеними, що встановлюється на етапі виконання перевірок.

```
// Обмін доларів на гривні через ПриватБанк
else if (radioButton2.Checked && radioButton8.Checked && radioButton11.Checked && v)
{ rate = pb_rateUSDtoGRN;
  rez = ExchangeSum * rate;
  grnN += rez;
  usdN -= ExchangeSum;
  textBox9.Text = grnN.ToString("0.00");
  textBox10.Text = usdN.ToString("0.00");
  textBox11.Text = textBox7.Text;
  textBox12.Text = textBox8.Text; }
```

Рисунок 3.27 – Обмін доларів на гривні через ПриватБанк

У разі успішного проведення поточної операції обміну, яке відбувається під час клацання по кнопці button3, на екран буде виведено вікно повідомлення «Інформація про операцію обміну валют», у якому буде відображений такий текст: «Операція обміну валют виконана! Проаналізуйте отримані результати!». Закривши це вікно, побачимо ще одне вікно повідомлення «Порада щодо подальших дій», у якому буде відображений текст: «Виконайте пункт 10, а потім пункт 11 даного додатку!».

Пункт 11 на формі – це клацання по кнопці button6. Обробник події Click цієї кнопки наведено на рисунку 3.29. Якщо попередньо було вибрано опцію «Так», то вносяться відповідні зміни у текстові вікна textBox1, textBox2, textBox3, textBox4, textBox9, textBox10, textBox11, textBox12, textBox13, змінній v присвоюємо значення true, і переходимо до виконання наступної транзакції. Якщо попередньо було вибрано опцію «Так», то закінчуємо виконання проекту.

```
v = false; // Операція обміну валют вже виконана
radioButton18.Checked = true;
MessageBox.Show("Операція обміну валют виконана! Проаналізуйте отримані результати!", "Інформація про операцію обміну валют", MessageBoxButtons.OK, MessageBoxIcon.Information);
MessageBox.Show("Виконайте пункт 10, а потім пункт 11 даного додатку!", "Порада щодо подальших дій", MessageBoxButtons.OK, MessageBoxIcon.Information);}
```

Рисунок 3.28 – Завершення поточної операції обміну валют

```

// Продовження виконання операцій обміну валют
private void button6_Click(object sender, EventArgs e)
{ if (radioButton18.Checked) // Закінчення виконання операцій обміну валют
  { this.Hide();
    Form1 f1 = new Form1();
    f1.Show(); } // Перехід на голону форму
v = true; // Операція обміну валют ще має бути виконана
// Відображення обсягів валют, що є в наявності
textBox1.Text = textBox9.Text;
textBox2.Text = textBox10.Text;
textBox3.Text = textBox11.Text;
textBox4.Text = textBox12.Text;
// Відображення обсягів валют,
// що є в наявності перед виконанням операції обміну
textBox5.Text = textBox9.Text;
textBox6.Text = textBox10.Text;
textBox7.Text = textBox11.Text;
textBox8.Text = textBox12.Text;
textBox13.Text = "0"; }

```

Рисунок 3.29 – Виконання наступної операції обміну валют

На рисунках 3.30 – 3.32 відображено результат тестування роботи проекту під час виконання операції введення наявного обсягу валют клієнта та обміну 1000 гривень на долари через ПриватБанк за поточним курсом обміну валют.

Спочатку вводяться у відповідні текстові поля наявні у клієнта обсяги валют (рис. 3.30), виконується команди «1. Введіть обсяги валют, які є у вас в наявності (через кому):» та «2. Натисніть цю кнопку, щоб зберегти обсяги валют, які є у Вас у наявності!».

Далі вибираємо типи валюти, яку міняємо, на яку міняємо, та банк чи обмінник для здійснення операції обміну, вводимо у текстове поле суму валюти, яка підлягає обміну та виконуємо команду «8. Натисніть тут, щоб виконати операцію обміну!» (рис. 3.31). Щоб отримати можливість продовжувати транзакції, слід вибрати значення «Так» в пункті 10.

Повний лістинг коду форми Form3 «Обмін валют» наведено у додатку А.

Form3. Обмін валют

ВИКОНАННЯ ТА АНАЛІЗ РЕУЛЬТАТІВ ОПЕРАЦІЙ З ОБМІНУ ВАЛЮТ

1. Введіть обсяги валют, яка є у Вас в наявності (через кому):

GRN	20000	USD	2000
EUR	1000	PLN	5000

2. Натисніть цю кнопку, щоб зберегти обсяги валют, яка є у Вас в наявності!

3. Перед виконанням операції обміну валют у Вас є в наявності:

GR	20000,00	USD	2000,00
EUR	1000,00	PLN	5000,00

4. Яку валюту хочете обміняти:

- GRN
- USD
- EUR
- PLN

5. На яку валюту хочете помінати:

- GRN
- USD
- EUR
- PLN

6. Виберіть банк чи обмінний пункт:

- НБУ
- ПриватБанк
- Монобанк
- Варшавка
- Luchesk
- Grand Volyn

7. Введіть суму валюту, яка підлягає обміну:

0

8. Натисніть тут, щоб виконати операцію обміну валюти!

9. Після виконання операції обміну валют у вас є в наявності:

GRN		USD	
EUR		PLN	

10. Хотите продовжити?

- ТАК
- НІ

11. Натисніть тут, щоб продовжити

Перейти на форму Банки

Перейти на форму Обмінники

Перейти на форму Робота з БД

Перейти на форму Обмін валют плюс

Вихід

Рисунок 3.30 – Введення обсягів валют, яка є в наявності у клієнта

Form3. Обмін валют

ВИКОНАННЯ ТА АНАЛІЗ РЕУЛЬТАТІВ ОПЕРАЦІЙ З ОБМІНУ ВАЛЮТ

1. Введіть обсяги валют, яка є у Вас в наявності (через кому):

GRN	20000	USD	2000
EUR	1000	PLN	5000

2. Натисніть цю кнопку, щоб зберегти обсяги валют, яка є у Вас в наявності!

3. Перед виконанням операції обміну валют у Вас є в наявності:

GR	20000,00	USD	2000,00
EUR	1000,00	PLN	5000,00

4. Яку валюту хочете обміняти:

- GRN
- USD
- EUR
- PLN

5. На яку валюту хочете помінати:

- GRN
- USD
- EUR
- PLN

6. Виберіть банк чи обмінний пункт:

- НБУ
- ПриватБанк
- Монобанк
- Варшавка
- Luchesk
- Grand Volyn

7. Введіть суму валюту, яка підлягає обміну:

1000

8. Натисніть тут, щоб виконати операцію обміну валюти!

9. Після виконання операції обміну валют у вас є в наявності:

GRN	19000,00	USD	2025,77
EUR	1000,00	PLN	5000,00

10. Хотите продовжити?

- ТАК
- НІ

11. Натисніть тут, щоб продовжити

Перейти на форму Банки

Перейти на форму Обмінники

Перейти на форму Робота з БД

Перейти на форму Обмін валют плюс

Вихід

Рисунок 3.31 – Початок виконання операції обміну 1000 гривень на долари через ПриватБанк

Form3. Обмін валют

ВИКОНАННЯ ТА АНАЛІЗ РЕЗУЛЬТАТІВ ОПЕРАЦІЙ З ОБМІНУ ВАЛЮТ

1. Введіть обсяги валют, яка є у Вас в наявності (через кому):

GRN	19000,00	USD	2025,77
EUR	1000,00	PLN	5000,00

4. Яку валюту хочете обміняти:

GRN

USD

EUR

PLN

5. На яку валюту хочете поміняти:

GRN

USD

EUR

PLN

6. Виберіть банк чи обмінний пункт:

НБУ

ПриватБанк

Монобанк

Варшавка

Luchesk

Grand Volyn

Перейти на форму Банки

Перейти на форму Обмінники

Перейти на форму Робота з БД

Перейти на форму Обмін валют плюс

2. Натисніть цю кнопку, щоб зберегти обсяги валют, яка є у Вас в наявності!

7. Введіть суму валюту, яка підлягає обміну:

0

8. Натисніть тут, щоб виконати операцію обміну валюту!

Перейти на форму Обмін валют плюс

3. Перед виконанням операції обміну валют у Вас є в наявності:

GR	19000,00	USD	2025,77
EUR	1000,00	PLN	5000,00

9. Після виконання операції обміну валют у вас є в наявності:

GRN	19000,00	USD	2025,77
EUR	1000,00	PLN	5000,00

10. Хочете продовжити?

ТАК НІ

Вихід

11. Натисніть тут, щоб продовжити

Рисунок 3.32 – Завершення обміну 1000 гривень на долари через ПриватБанк

3.6 Форма Form4 «Робота з базою даних»

Форма Form4 «Робота з базою даних» (рис. 3.33) призначена для створення БД з інформацією про клієнтів пункту обміну валют та виконання операцій з цією БД. Потреба у такій БД виникає тоді, коли пункт обміну валют має своїх постійних клієнтів, у чому він, звичайно, зацікавлений.

Приступаємо до створення самої форми. На ній розміщуємо п'ять кнопок button для організації навігації по формах проекту, про що говорилося раніше. Далі виконуємо команду Toolbox/Mtnus & Toolbars/MenuStrip і перетягуємо компонент MenuStrip на форму, щоб створити об'єкт menuStrip1, тобто меню команд. У цьому меню передбачаємо команди Файл, Інструменти, Довідка. У команді Файл передбачаємо підлеглу команду Вихід, а в команді Інструменти передбачаємо підлеглу команду Обновити.

Виконуємо команду Toolbox/Mtnus & Toolbars/TabControl і перетягуємо компонент TabControl на форму, щоб створити об'єкт tabControl1, тобто ще одне меню команд для роботи вже безпосередньо з БД. Властивість Dock цього об'єкта встановлюємо Left, у результаті він буде притиснутий до лівого боку

вікна форми. Далі знаходимо властивість TabPages об'єкта tabControl1 і за допомогою вікна Collection... створюємо такі об'єкти:

- tabPage1, властивість Text якого встановлюємо SELECT;
- tabPage2, властивість Text якого встановлюємо INSERT;
- tabPage3, властивість Text якого встановлюємо UPDATE;
- tabPage4, властивість Text якого встановлюємо DELETE.

Це й будуть команди меню для роботи з БД.

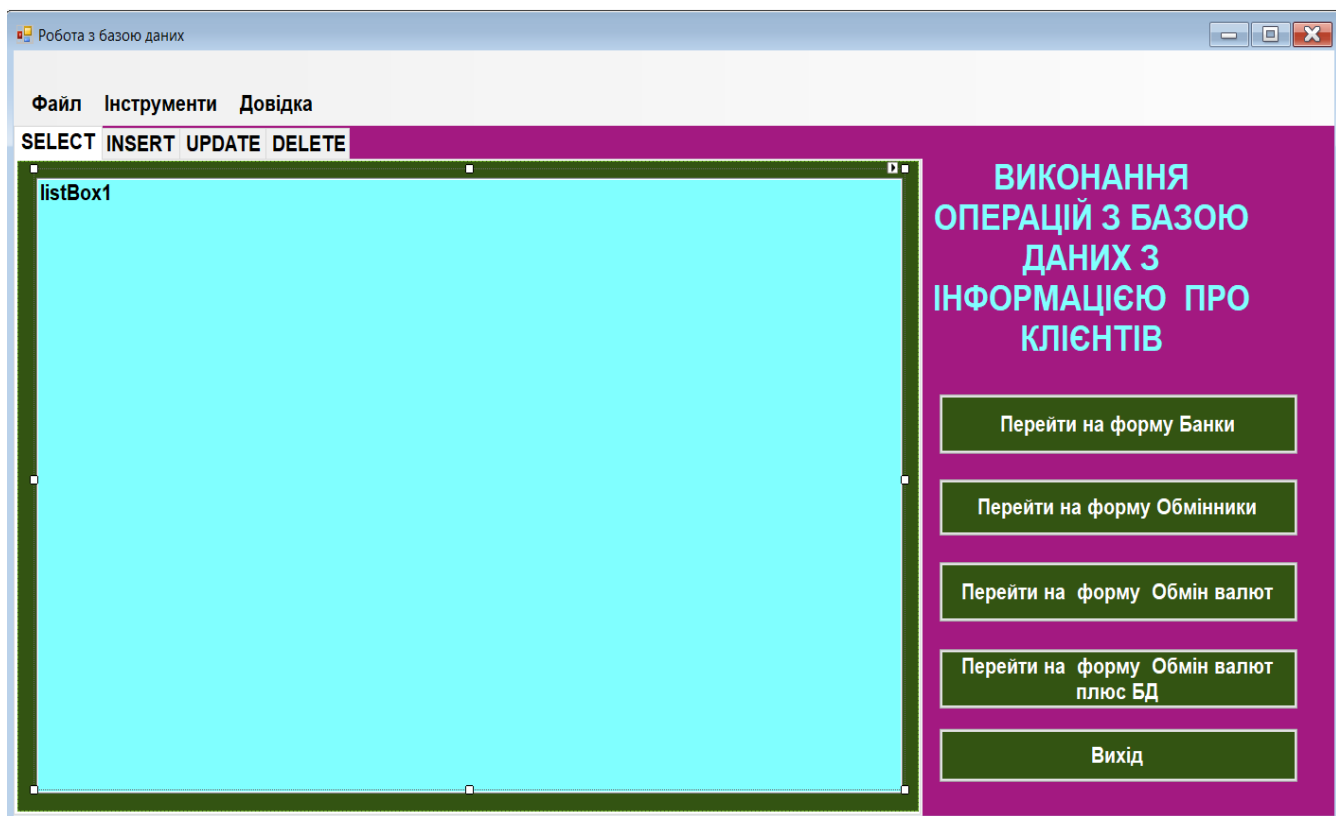


Рисунок 3.33 – Вигляд форми Form4 «Робота з базою даних» при завантаженні

Тепер на часі створити саму БД. Перш за все додаємо до проекту файл нової БД, що базується на службах. Клацаємо ПКМ по імені проекту BachelorWork у вікні браузера проектів і виконуємо команду Add/Components... У вікні Add New Item, що з'являється, вибираємо вкладку Service-based Database, залишаємо стандартне ім'я файла бази даних Database.mdf і клацаємо по кнопці Add. У браузері проектів з'явиться ім'я файла бази даних Database.mdf.

У браузері проектів двічі клацаємо по імені файла бази даних Database.mdf.

У браузері серверів (рис. 3.34) з'явиться структура нашої БД, поки що порожньої. Клацаємо ПКМ по вкладці Table і вибираємо вкладку Add New Table. У скрипті нової таблиці можна задати нове ім'я таблиці. Ми ж залишимо його стандартним, тобто Table. Командою Save All завершуємо процес створення таблиці БД. Оновлюємо вміст БД і викликаємо таблицю Table для розробки її структури та заповнення її даними (рис. 3.35).

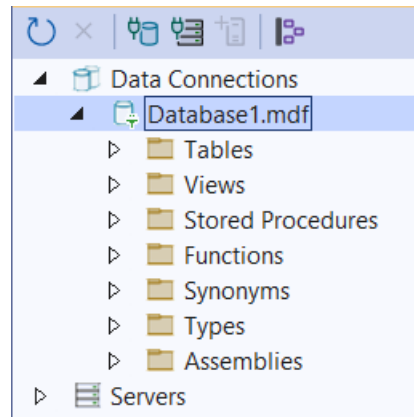


Рисунок 3.34 – Структура БД Database1

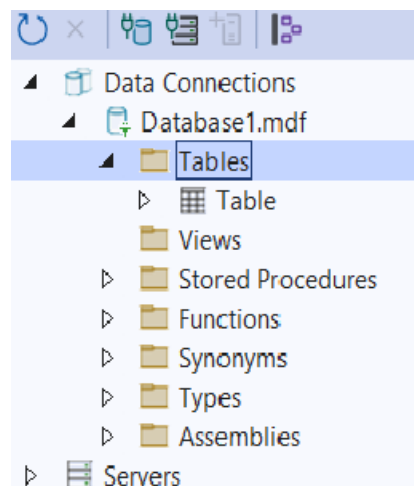


Рисунок 3.35 – Вибір таблиці Table БД Database1

Наша БД повинна мати такі поля: Id, Name, GRN, USD, EUR, PLN, причому поле Id буде ключовим, а решта полів матиме тип nvarchar(50). На рисунку 3.36 показана БД Database1, заповнена даними з інформацією про клієнтів.

	Id	Name	GRN	USD	EUR	PLN
▶	1	Коць Богдан С...	20000	2000	1000	5000
	2	Кузьмюк Олекс...	22000	2200	1200	5200
	3	Серський Дмит...	23000	2300	1300	5300
	4	Климець Влад...	24000	2400	1400	5400
	5	Шевчук Юрій ...	25000	2500	1500	5500
	6	Христинець На...	25000	5000	10000	20000
	11	Лавренчук Світ...	20000	2000	2000	20000
*	NULL	NULL	NULL	NULL	NULL	NULL

Рисунок 3.36 – Контент БД Database1

Наступним кроком є розроблення коду команди SELECT, яка виконується під час завантаження форми Form4. Тому обробник події Load форми Form4 (рис. 3.37) є одночасно і командою SELECT.

```
//-----
//Реалізація пункту меню SELECT
private async void Form4_Load(object sender, EventArgs e)
{
    string connectionString = @"Data
Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=""C:\Users\Desktop\Kots Bohdan
2024\BachelorWork\BachelorWork\Database1.mdf"";Integrated Security=True";
    sqlConnection = new SqlConnection(connectionString);
    await sqlConnection.OpenAsync();
    SqlDataReader sqlReader = null;
    SqlCommand command = new SqlCommand("SELECT * FROM [Table]",
sqlConnection);
    try
    {
        sqlReader = await command.ExecuteReaderAsync();
        while (await sqlReader.ReadAsync())
        {
            listBox1.Items.Add(Convert.ToString(sqlReader["Id"]) + " " +
Convert.ToString(sqlReader["Name"]) + " " + Convert.ToString(sqlReader["GRN"]) + " "
+ Convert.ToString(sqlReader["USD"]) + " " + Convert.ToString(sqlReader["EUR"]) + " "
+ Convert.ToString(sqlReader["PLN"]));
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message.ToString(), ex.Source.ToString(),
MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
    finally
    {
        if (sqlReader != null)
            sqlReader.Close();
    }
}
```

Рисунок 3.37 – Код обробника події Load форми Form4 (команда SELECT)

Зазначимо, що на самій формі записи БД будуть відображатися в об'єкті listBox1, який має бути попередньо розміщений на вкладці tabPage1. Для отримання значення змінної connectionString, яка визначає з'єднання фізичного файлу БД з проектом, необхідно клацнути ПКМ по імені файла Database1.mdf у браузері серверів, викликати вікно Властивостей (рис. 3.38) і вибрати властивість Connection String, значення якої і визначає шукане з'єднання.

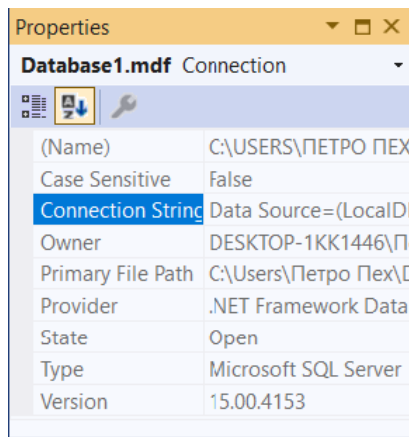


Рисунок 3.38 – Властивість Connection String з даними про підключення БД

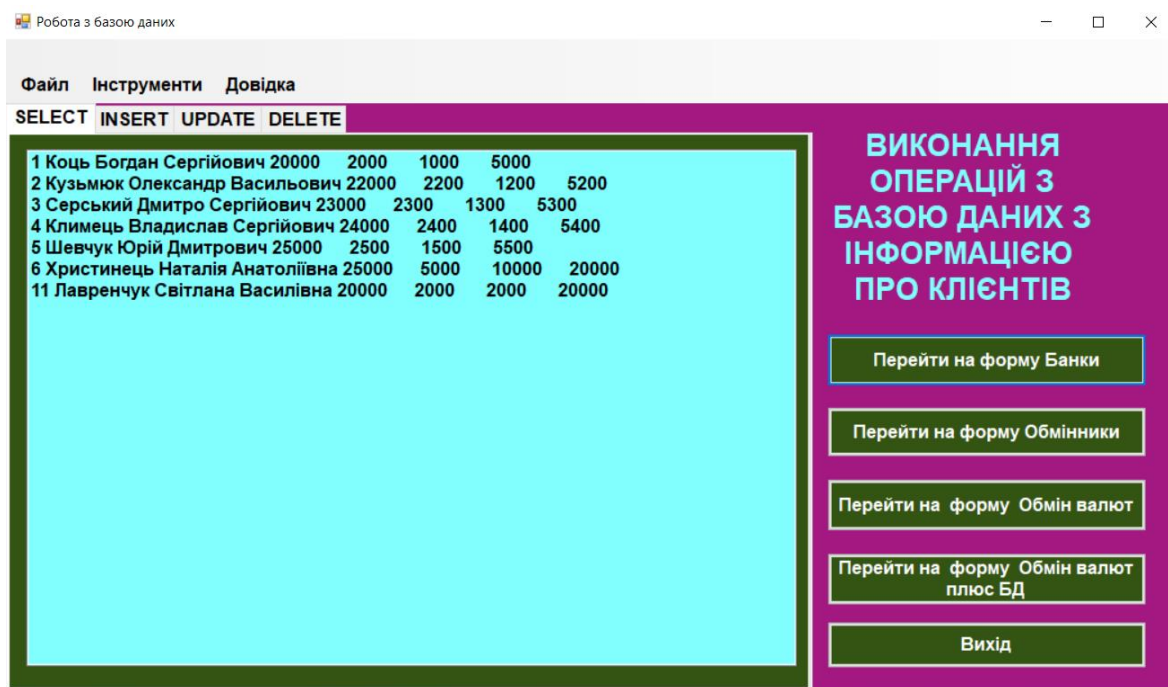


Рисунок 3.39 – Результат виконання команди SELECT

Звичайно, сама змінна connectionString попередньо має бути оголошена у

класі Form4 як об’єкт класу Connection String. Виконавши команду SELECT, на екрані побачимо вміст БД Database1.mdf (рис. 3.39).

Команда INSERT призначена для додавання нового запису у кінець таблиці БД. На вкладці tabPages2 (рис. 3.40), яка реалізує команду INSERT, розміщено мітки label2 з надписом «Прізвище, ім’я та по батькові клієнта:», label3 з надписом «GRN», label4 з надписом «USD», label5 з надписом «EUR», label6 з надписом «PLN», label7 з надписом «Повідомлення про помилку», текстові поля textBox1 – textBox5 для введення в них значень полів записів БД та кнопка button3 з текстом «Додати запис до БД!».

Рисунок 3.40 – Підготовка до виконання команди INSERT

Код обробника події Click кнопки button3 (рис. 41), яка власне реалізує додавання нового запису до БД, базується на використанні об’єкта command класу SqlCommand, оголошеного у спосіб: `SqlCommand command = new SqlCommand("INSERT INTO [Table], (Name, GRN, USD, EUR, PLN)VALUES(@Name, @GRN, @USD, @EUR, @PLN)", sqlConnection)`. Після кожної модифікації БД її потрібно оновити.

```
//Реалізація пункту меню INSERT
private async void button3_Click(object sender, EventArgs e)
{
    if (label7.Visible) label7.Visible = false;
    if (!string.IsNullOrEmpty(textBox1.Text) &&
!string.IsNullOrEmpty(textBox2.Text) &&
!string.IsNullOrEmpty(textBox3.Text) &&
!string.IsNullOrEmpty(textBox4.Text) &&
!string.IsNullOrEmpty(textBox5.Text) &&
!string.IsNullOrWhiteSpace(textBox1.Text) &&
!string.IsNullOrWhiteSpace(textBox2.Text) &&
!string.IsNullOrWhiteSpace(textBox3.Text) &&
!string.IsNullOrWhiteSpace(textBox4.Text) &&
!string.IsNullOrWhiteSpace(textBox5.Text))
    {
        SqlCommand command = new SqlCommand("INSERT INTO [Table] (Name, GRN,
USD, EUR, PLN)VALUES(@Name, @GRN, @USD, @EUR, @PLN)", sqlConnection);
        command.Parameters.AddWithValue("Name", textBox1.Text);
        command.Parameters.AddWithValue("GRN", textBox2.Text);
        command.Parameters.AddWithValue("USD", textBox3.Text);
        command.Parameters.AddWithValue("EUR", textBox4.Text);
        command.Parameters.AddWithValue("PLN", textBox5.Text);
        await command.ExecuteNonQueryAsync();
        MessageBox.Show("Команда INSERT виконана!", "Підтвердження виконання
команди INSERT.", MessageBoxButtons.OK, MessageBoxIcon.Information);
    }
    else
    {
        label7.Visible = true;
        label7.Text = "Поля 'Name', 'GRN', 'USD', 'EUR', 'PLN' повинні бути
заповнені!";
    }
}
```

Рисунок 3.41 – Код обробника події Click кнопки button3

Код обробника події Click елемента ToolStripMenuItem наведено на рисунку 3.42. Результат виконання команди INSERT, наведейго на рисунку 3.43, свідчить про її коректну роботу.

Для чого потрібно оновлювати БД? Просте правило – якщо ми вносимо хоча би якусь зміну у структуру або контент БД, її кожного разу потрібно оновлювати. У протилежному випадку ми завжди ризикуємо втратити цілісність БД або її отримати ситуацію, кол БД працює некоректно. Крім того, якщо ми в певний момент відкрили БД, після завершення роботи з нею ми обов’язково повинні її закрити.

```
//Реалізація пункту меню Обновити
private async void обновитиToolStripMenuItem_Click(object sender, EventArgs e)
{
    listBox1.Items.Clear();
    SqlDataReader sqlReader = null;
    SqlCommand command = new SqlCommand("SELECT * FROM [Table]",
sqlConnection);
    try
    {
        sqlReader = await command.ExecuteReaderAsync();
        while (await sqlReader.ReadAsync())
        {
            listBox1.Items.Add(Convert.ToString(sqlReader["Id"]) + " " +
Convert.ToString(sqlReader["Name"]) + " " + Convert.ToString(sqlReader["GRN"]) + " "
+ Convert.ToString(sqlReader["USD"]) + " " + Convert.ToString(sqlReader["EUR"]) + " "
+ Convert.ToString(sqlReader["PLN"])); } }
        catch (Exception ex)
        {
            MessageBox.Show(ex.Message.ToString(), ex.Source.ToString(),
MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
    }
    finally
    {
        if (sqlReader != null)
            sqlReader.Close();
    }
}
}
```

Рисунок 3.42 – Код обробника події Click елемента ToolStripMenuItem



Рисунок 3.43 – Результат виконання команди INSERT

Команда UPDATE призначена для редагування запису БД зі заданим Id. На вкладці tabPages3 (рис. 3.44), яка реалізує команду UPDATE, розміщено мітки label13 з надписом «Прізвище, ім'я та по батькові клієнта:», label12 з надписом «GRN», label11 з надписом «USD», label10 з надписом «EUR», label9 з надписом «PLN», label8 з надписом «Повідомлення про помилку», текстові поля textBox6 – textBox10, textBox17 для введення в них значень полів записів БД та кнопка button3 з текстом «Виправити запис БД!».

Після кожної модифікації БД, тобто виконання команди UPDATE, БД потрібно оновити. Результат виконання команди UPDATE, наведений на рисунку 3.45, свідчить про коректну роботу команди.

Код обробника події Click кнопки button4 (рис. 3.46), яка реалізує операцію модифікації запису БД зі заданим Id, базується на використанні об'єкта command класу SqlCommand, оголошеного у спосіб: `SqlCommand command = new SqlCommand("UPDATE [Table] SET [Name] = @Name, [GRN] = @GRN, [USD] = @USD, [EUR] = @EUR, [PLN] = @PLN WHERE [Id] = @Id", sqlConnection).`

Рисунок 3.44 – Підготовка до виконання команди UPDATE

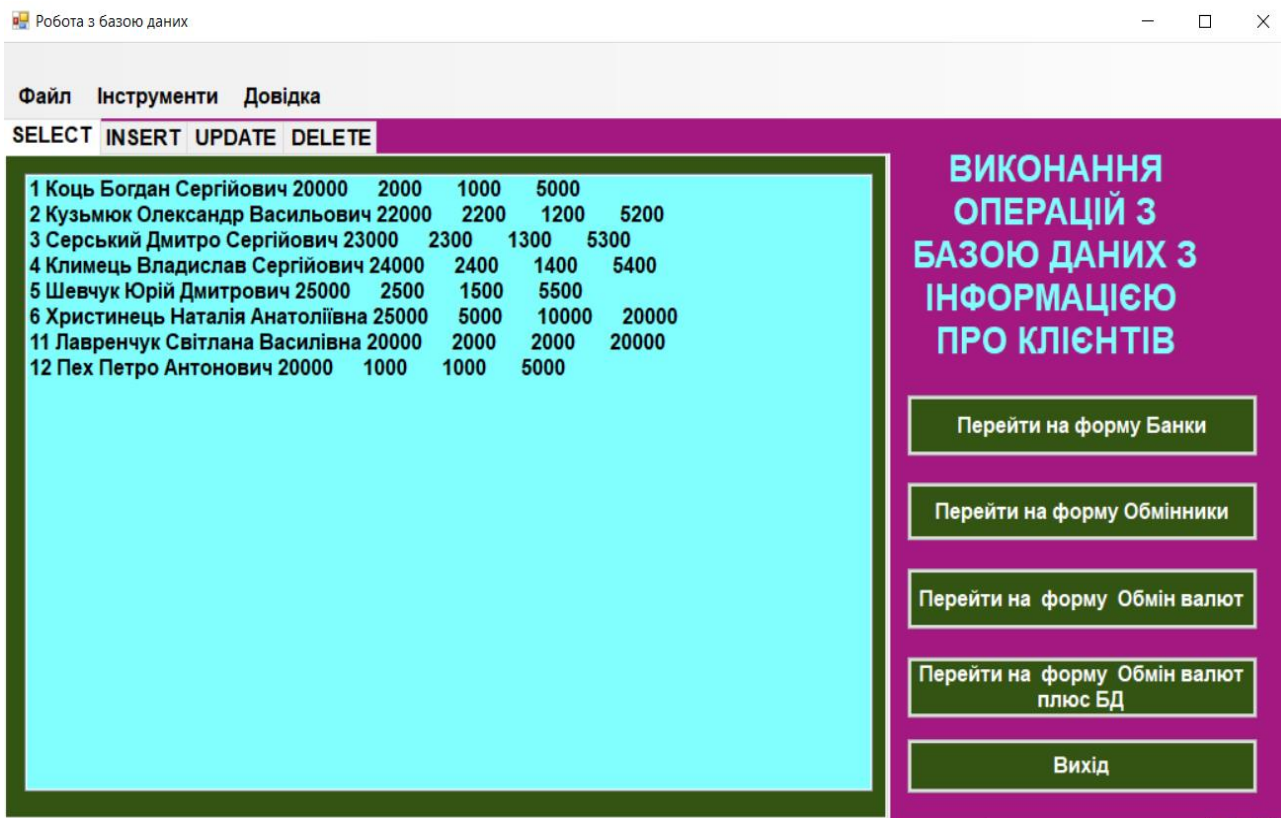


Рисунок 3.45 – Результат виконання команди UPDATE

Команда DELETE призначена для вилучення запису БД зі заданим Id. На вкладці tabPages4 (рис. 3.47), яка реалізує команду DELETE, розміщено мітку label21 з надписом «Id», мітку label20 з надписом «Повідомлення про помилку», текстове поле textBox16 для введення в нього значення Id та кнопка button8 з текстом «Вилучити запис з БД!».

Після виконання команди DELETE БД потрібно оновити. Результат виконання команди DELETE, наведений на рисунку 3.48, свідчить про коректну роботу команди.

Код обробника події Click кнопки button8 (рис. 3.49), яка реалізує операцію вилучення запису БД зі заданим Id, базується на використанні об'єкта command класу SqlCommand, оголошеного у такий спосіб: `SqlCommand command = new SqlCommand("DELETE FROM [Table] DELETE WHERE [Id] = @Id", sqlConnection)`. У разі, коли Id клієнта не введено, передбачено виведення відповідного повідомлення за допомогою методу `MessageBox.Show()`, один з аргументів якого містить це повідомлення.

Команда DELETE призначена для вилучення запису БД зі заданим Id. На вкладці tabPages4 (рис. 3.47), яка реалізує команду DELETE, розміщено мітку label21 з надписом «Id», мітку label20 з надписом «Повідомлення про помилку», текстове поле textBox16 для введення в нього значення Id та кнопка button8 з текстом «Вилучити запис з БД!».

Після виконання команди DELETE БД потрібно оновити. Результат виконання команди DELETE, наведений на рисунку 3.48, свідчить про коректну роботу команди.

Код обробника події Click кнопки button8 (рис. 3.49), яка реалізує операцію вилучення запису БД зі заданим Id, базується на використанні об'єкта command класу SqlCommand, оголошеного у такий спосіб: `SqlCommand command = new SqlCommand("DELETE FROM [Table] DELETE WHERE [Id] = @Id", sqlConnection)`. У разі, коли Id клієнта не введено, передбачено виведення відповідного повідомлення за допомогою методу `MessageBox.Show()`, один з аргументів якого містить це повідомлення.

```
//Реалізація пункту меню UPDATE
private async void button4_Click(object sender, EventArgs e)
{
    if (label8.Visible)
        label8.Visible = false;
    if (!string.IsNullOrEmpty(textBox17.Text) &&
        !string.IsNullOrEmpty(textBox10.Text) &&
        !string.IsNullOrEmpty(textBox9.Text) &&
        !string.IsNullOrEmpty(textBox8.Text) &&
        !string.IsNullOrEmpty(textBox7.Text) &&
        !string.IsNullOrEmpty(textBox6.Text) )
        {
            SqlCommand command = new SqlCommand("UPDATE [Table] SET [Name] =
@Name, [GRN] = @GRN, [USD] = @USD, [EUR] = @EUR, [PLN] = @PLN WHERE [Id] = @Id",
sqlConnection);
```

Рисунок 3.46 – Код обробника події Click кнопки button4

У разі виконання команди Вихід меню форми спрацьовує код обробника події Click елемента ToolStripMenuItem (рис. 3.50), який забезпечує збереження та закриття БД. Те ж саме відбувається під час закриття форми Form4.

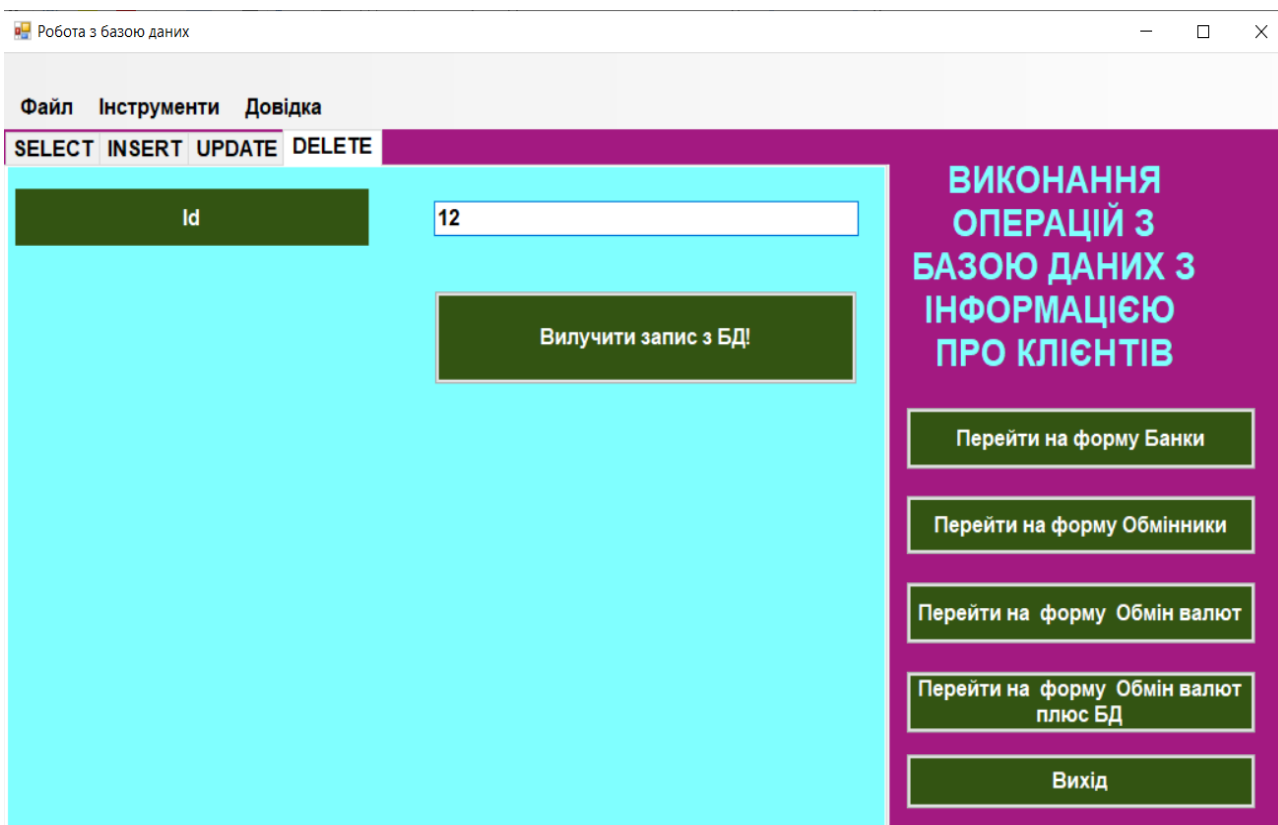


Рисунок 3.47 – Підготовка до виконання команди DELETE

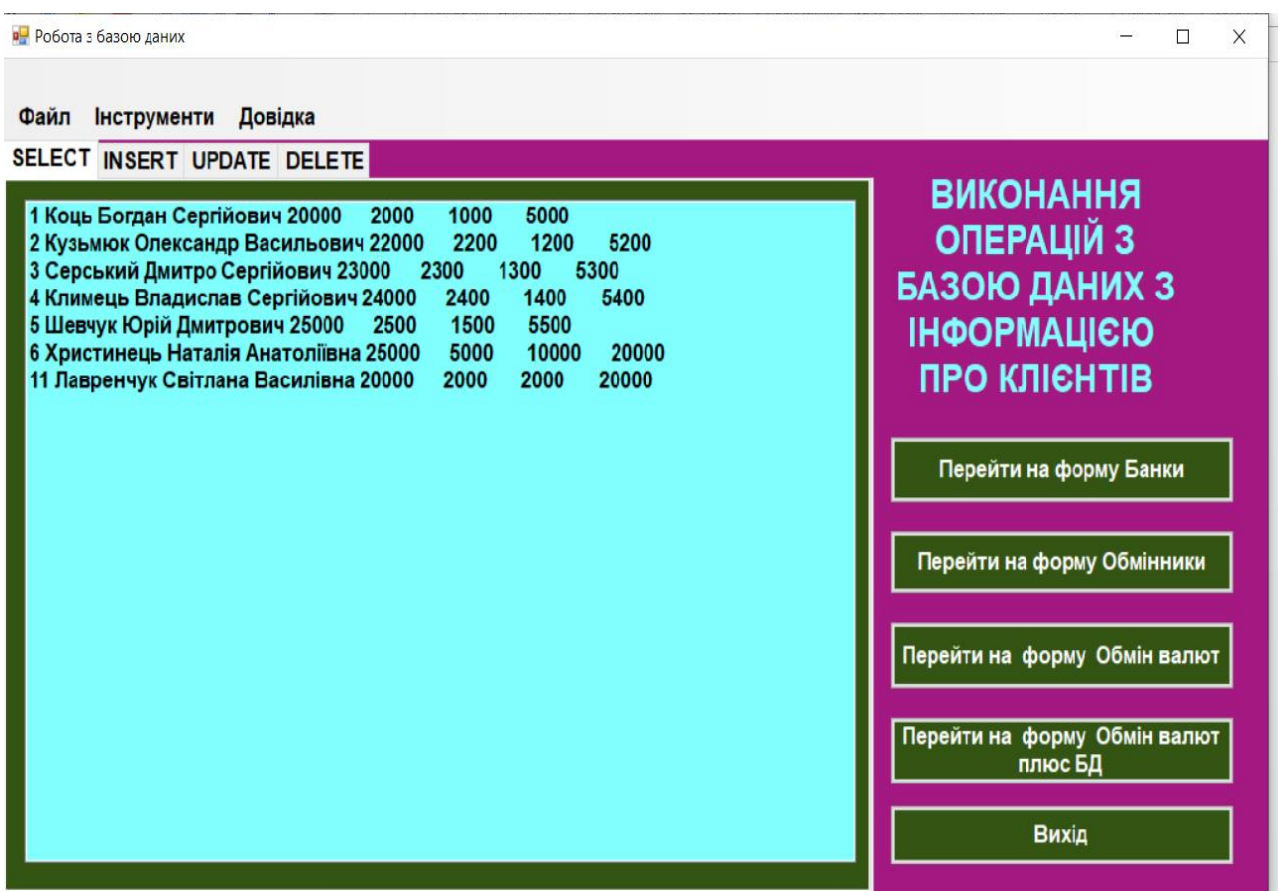


Рисунок 3.48 – Результат виконання команди DELETE

```
//Реалізація пункту меню DELETE
private async void button8_Click(object sender, EventArgs e)
{
    if (label20.Visible)
        label20.Visible = false;
    if (!string.IsNullOrEmpty(textBox16.Text) &&
!string.IsNullOrWhiteSpace(textBox16.Text))
        {
            SqlCommand command = new SqlCommand("DELETE FROM [Table] WHERE [Id] =
@Id", sqlConnection);
            command.Parameters.AddWithValue("Id", textBox16.Text);
            await command.ExecuteNonQueryAsync();
            MessageBox.Show("Команда DELETE виконана!", "Підтвердження виконання
команди DELETE.", MessageBoxButtons.OK, MessageBoxIcon.Information); }
        else
        {
            label20.Visible = true;
            label20.Text = "Поле Id має бути заповнене!"; } }
} }
```

Рисунок 3.49 – Код обробника події Click кнопки button8

```
//Реалізація пункту меню Вихід
private void вихідToolStripMenuItem_Click(object sender, EventArgs e)
{
    if (sqlConnection != null && sqlConnection.State !=
ConnectionState.Closed)
        sqlConnection.Close(); }

private void Form4_FormClosing(object sender, FormClosingEventArgs e)
{
    if (sqlConnection != null && sqlConnection.State !=
ConnectionState.Closed)
        sqlConnection.Close(); }
```

Рисунок 3.50 – Код обробника події Click елемента ToolStripMenuItem
(команда Вихід)

Повний лістинг коду форми Form4 «Робта з БД» наведено у додатку Б.

3.7 Форма Form5 «Обмін валют з підключенням бази даних»

Форма Form5 «Обмін валют з підключенням БД» (рис. 3.51) призначена для використання раніше створеної БД з інформацією про клієнтів пункту обміну валют для виконання операцій обміну валют. Ця форма поєднує у собі

функціональні можливості двох попередніх форм і додає деякі нові.

Згідно поставленого завдання, на формі «Обмін валют з підключенням БД» ми повинні задати Id клієнта пункту обміну валют. Використовуючи Id клієнта, форма має вирішити задачі: зчитування відповідного запису з БД, виконання операцій обміну валют стосовно зчитаного запису, збереження запису зі заданим значенням Id та зі зміненими іншими полями в БД.

З метою вирішення усіх цих завдань на форму Form5 в об'єкт listBox1 виводяться всі записи БД, що дає можливість переглянути всі записи БД і вибрати потрібний. Далі значення Id вводиться у перше текстове поле (рис. 3.52) і виконується команда «1. Введіть у вікно, що зверху, Id клієнта і натисніть цю кнопку, щоб отримати його дані». Дані про клієнта відобразяться на формі у відповідних текстових полях (рис. 3.53). Для того, щоб їх можна було змінювати в процесі подальшого опрацювання, виконуємо команду «2. Натисніть цю кнопку, щоб зберегти обсяги валют, що у Вас є в наявності!». Величини обсягів валют, що є в наявності у клієнта перед виконанням операцій з обміну валют, відобразяться на формі у відповідних текстових полях (рис. 3.54). Наступні пункти 4 – 9 виконуються так, як описано в розділі 3.6 (рис. 3.55 – 3.56).

У результаті виконання операцій з обміну валют дані щодо обсягів валют клієнта будуть змінені. Ці змінені дані потрібно зберегти у нашій БД (рис. 3.57) та оновити БД, оскільки вона була змінена (рис. 3.58). Власне, на цьому реалізація однієї транзакції завершується. Далі маємо визначитися, чи хочемо ми продовжувати виконання операцій з обміну валют (рис. 3.59), і, якщо так, то отримаємо для роботи вікно, зображене на рис. 3.60; якщо ні, то відбувається вихід з проекту.

Повний лістинг коду форми Form5 наведено у додатку В.

Form5. Обмін валют з підключенням БД

ВИКОНАННЯ ОПЕРАЦІЙ З ОБМІНУ ВАЛЮТ З ПІДКЛЮЧЕННЯМ БАЗИ ДАНИХ

1. Введіть у вікно, що зверху, Іd клієнта і натисніть цю кнопку, щоб отримати його дані!

listBox1

4. Яку валюту хочете обміняти:

GRN

USD

EUR

PLN

5. На яку валюту хочете поміняти:

GRN

USD

EUR

PLN

6. Виберіть банк чи обмінний пункт:

НБУ

ПриватБанк

Монобанк

Варшавка

Luchesk

Grand Volyn

7. Введіть суму валюту, яка підлягає обміну:

8. Натисніть тут, щоб виконати операцію обміну валюти!

9. Після виконання операції обміну валют у Вас є в наявності:

10. Натисніть тут, щоб зберегти запис у БД

11. Натисніть тут, щоб оновити БД

12. Хотите продовжити?

ТАК НІ

Перейти на форму Банки

Перейти на форму Обмінники

Перейти на форму Обмін валют

Перейти на форму Робота З БД

Вихід

14. Натисніть тут, щоб продовжити чи вийти

Рисунок 3.51 – Вигляд форми Form5. Обмін валют з підключенням бази даних

Form5. Обмін валют з підключенням БД

ВИКОНАННЯ ОПЕРАЦІЙ З ОБМІНУ ВАЛЮТ З ПІДКЛЮЧЕННЯМ БАЗИ ДАНИХ

2

1. Введіть у вікно, що зверху, Іd клієнта і натисніть цю кнопку, щоб отримати його дані!

1 Коць Богдан Сергійович	20000	2000	1000	5000
2 Кузьмюк Олександр Васильович	18000.00	2302.92	1247.06	5200.1
3 Серський Дмитро Сергійович	23000	2300	1300	5300
4 Климець Владислав Сергійович	24000	2400	1400	5400

4. Яку валюту хочете обміняти:

GRN

USD

EUR

PLN

5. На яку валюту хочете поміняти:

GRN

USD

EUR

PLN

6. Виберіть банк чи обмінний пункт:

НБУ

ПриватБанк

Монобанк

Варшавка

Luchesk

Grand Volyn

7. Введіть суму валюту, яка підлягає обміну:

0

8. Натисніть тут, щоб виконати операцію обміну валюти!

9. Після виконання операції обміну валют у Вас є в

10. Натисніть тут, щоб зберегти запис у БД

11. Натисніть тут, щоб оновити БД

12. Хотите продовжити?

ТАК НІ

Перейти на форму Банки

Перейти на форму Обмінники

Перейти на форму Обмін валют

Перейти на форму Робота З БД

Вихід

14. Натисніть тут, щоб продовжити чи вийти

Рисунок 3.52 – Введення ідентифікатора запису клієнта

ВИКОНАННЯ ОПЕРАЦІЙ З ОБМІНУ ВАЛЮТ З ПІДКЛЮЧЕННЯМ БАЗИ ДАНИХ

2

1. Введіть у вікно, що зверху, Іd клієнта і натисніть цю кнопку, щоб отримати його дані!

1 Коць Богдан Сергійович	20000	2000	1000	5000
2 Кузьмюк Олександр Васильович	18000,00	2302,92	1247,06	5200,00
3 Серський Дмитро Сергійович	23000	2300	1300	5300
4 Климець Владислав Сергійович	24000	2400	1400	5400

Кузьмюк Олександр Васильович

GRN	18000,00	USD	2302,92
EUR	1247,06	PLN	5200,00

2. Натисніть цю кнопку, щоб зберегти обсяги валют, яка є у Вас в наявності!

3. Перед виконанням операції обміну валют у Вас є в наявності:

GR		USD	
EUR		PLN	

4. Яку валюту хочете обміняти:

GRN

5. На яку валюту хочете поміняти:

GRN

6. Виберіть банк чи обмінний пункт:

НБУ

ПриватБанк

Монобанк

Варшавка

Luchesk

Grand Volyn

Перейти на форму Банки

Перейти на форму Обмінники

Перейти на форму Обмін валют

Перейти на форму Робота з БД

Вихід

7. Введіть суму валюту, яка підлягає обміну:

0

8. Натисніть тут, щоб виконати операцію обміну валюту!

9. Після виконання операції обміну валют у Вас є в

10. Натисніть тут, щоб зберегти запис у БД

11. Натисніть тут, щоб оновити БД

12. Хотите продовжити?

ТАК НІ

14. Натисніть тут, щоб продовжити чи вийти

Підтвердження виконання команди вибірки запису з БД

Команда вибірки запису з БД виконана! Далі виконайте пункт 2!

OK

Рисунок 3.53 – Відображення запису клієнта на формі

ВИКОНАННЯ ОПЕРАЦІЙ З ОБМІНУ ВАЛЮТ З ПІДКЛЮЧЕННЯМ БАЗИ ДАНИХ

2

1. Введіть у вікно, що зверху, Іd клієнта і натисніть цю кнопку, щоб отримати його дані!

1 Коць Богдан Сергійович	20000	2000	1000	5000
2 Кузьмюк Олександр Васильович	18000,00	2302,92	1247,06	5200,00
3 Серський Дмитро Сергійович	23000	2300	1300	5300
4 Климець Владислав Сергійович	24000	2400	1400	5400

Кузьмюк Олександр Васильович

GRN	18000,00	USD	2302,92
EUR	1247,06	PLN	5200,00

2. Натисніть цю кнопку, щоб зберегти обсяги валют, яка є у Вас в наявності!

3. Перед виконанням операції обміну валют у Вас є в наявності:

GR	18000,00	USD	2302,92
EUR	1247,06	PLN	5200,00

4. Яку валюту хочете обміняти:

GRN

USD

EUR

PLN

5. На яку валюту хочете поміняти:

GRN

USD

EUR

PLN

6. Виберіть банк чи обмінний пункт:

НБУ

ПриватБанк

Монобанк

Варшавка

Luchesk

Grand Volyn

Перейти на форму Банки

Перейти на форму Обмінники

Перейти на форму Обмін валют

Перейти на форму Робота з БД

Вихід

7. Введіть суму валюту, яка підлягає обміну:

0

8. Натисніть тут, щоб виконати операцію обміну валюту!

9. Після виконання операції обміну валют у Вас є в

10. Натисніть тут, щоб зберегти запис у БД

11. Натисніть тут, щоб оновити БД

12. Хотите продовжити?

ТАК НІ

14. Натисніть тут, щоб продовжити чи вийти

Рисунок 3.54 – Відображення обсягів валют клієнта перед операцією обміну

ВИКОНАННЯ ОПЕРАЦІЙ З ОБМІНУ ВАЛЮТ З ПІДКЛЮЧЕННЯМ БАЗИ ДАНИХ

2

1. Введіть у вікно, що зверху, Іd клієнта і натисніть цю кнопку, щоб отримати його дані!

1 Коць Богдан Сергійович	20000	2000	1000	5000
2 Кузьмюк Олександр Васильович	18000,00	2302,92	1247,06	5200,00
3 Серський Дмитро Сергійович	23000	2300	1300	5300
4 Климець Владислав Сергійович	24000	2400	1400	5400

Кузьмюк Олександр Васильович

GRN	18000,00	USD	2302,92
EUR	1247,06	PLN	5200,00

2. Натисніть цю кнопку, щоб зберегти обсяги валют, яка є у Вас в наявності!

3. Перед виконанням операції обміну валют у Вас є в наявності:

GR	18000,00	USD	2302,92
EUR	1247,06	PLN	5200,00

4. Яку валюту хочете обміняти:

GRN

USD

EUR

PLN

5. На яку валюту хочете поміняти:

GRN

USD

EUR

PLN

6. Виберіть банк чи обмінний пункт:

НБУ

ПриватБанк

Монобанк

Варшавка

Luchesk

Grand Volyn

7. Введіть суму валюту, яка підлягає обміну:

1000

8. Натисніть тут, щоб виконати операцію обміну валюту!

9. Після виконання операції обміну валют у Вас є в

GRN		USD	
EUR		PLN	

10. Натисніть тут, щоб зберегти запис у БД

11. Натисніть тут, щоб оновити БД

12. Хочете продовжити?

ТАК НІ

14. Натисніть тут, щоб продовжити чи вийти

Перейти на форму Банки

Перейти на форму Обмінники

Перейти на форму Обмін валют

Перейти на форму Робота з БД

Вихід

Рисунок 3.55 – Вибір типу валюти, яку міняємо, на яку міняємо і банку чи обмінника

ВИКОНАННЯ ОПЕРАЦІЙ З ОБМІНУ ВАЛЮТ З ПІДКЛЮЧЕННЯМ БАЗИ ДАНИХ

2

1. Введіть у вікно, що зверху, Іd клієнта і натисніть цю кнопку, щоб отримати його дані!

1 Коць Богдан Сергійович	20000	2000	1000	5000
2 Кузьмюк Олександр Васильович	18000,00	2302,92	1247,06	5200,00
3 Серський Дмитро Сергійович	23000	2300	1300	5300
4 Климець Владислав Сергійович	24000	2400	1400	5400

Кузьмюк Олександр Васильович

GRN	18000,00	USD	2302,92
EUR	1247,06	PLN	5200,00

2. Натисніть цю кнопку, щоб зберегти обсяги валют, яка є у Вас в наявності!

3. Перед виконанням операції обміну валют у Вас є в наявності:

GR	18000,00	USD	2302,92
EUR	1247,06	PLN	5200,00

4. Яку валюту хочете обміняти:

GRN

5. На яку валюту хочете поміняти:

GRN

6. Виберіть банк чи обмінний пункт:

НБУ

ПриватБанк

Монобанк

Варшавка

Luchesk

Grand Volyn

7. Введіть суму валюту, яка підлягає обміну:

1000

8. Натисніть тут, щоб виконати операцію обміну валюту!

9. Після виконання операції обміну валют у Вас є в

GRN	17000,00	USD	2328,69
EUR	1247,06	PLN	5200,00

10. Натисніть тут, щоб зберегти запис у БД

11. Натисніть тут, щоб оновити БД

12. Хочете продовжити?

ТАК НІ

14. Натисніть тут, щоб продовжити чи вийти

Перейти на форму Банки

Перейти на форму Обмінники

Перейти на форму Обмін валют

Перейти на форму Робота з БД

Вихід

Рисунок 3.56 – Введення суми валюти, яка підлягає обміну, і виконання операції обміну

ВИКОНАННЯ ОПЕРАЦІЙ З ОБМІНУ ВАЛЮТ З ПІДКЛЮЧЕННЯМ БАЗИ ДАНИХ

2

1. Введіть у вікно, що зверху, Іd клієнта і натисніть цю кнопку, щоб отримати його дані!

1 Коць Богдан Сергійович	20000	2000	1000	5000
2 Кузьмюк Олександр Васильович	18000,00	2302,92	1247,06	5200,00
3 Серський Дмитро Сергійович	23000	2300	1300	5300
4 Климець Владислав Сергійович	24000	2400	1400	5400

Кузьмюк Олександр Васильович

GRN	18000,00	USD	2302,92
EUR	1247,06	PLN	5200,00

2. Натисніть цю кнопку, щоб зберегти обсяги валют, яка є у Вас в наявності!

3. Перед виконанням операції обміну валют у Вас є в наявності:

GR	18000,00	USD	2302,92
EUR	1247,06	PLN	5200,00

4. Яку валюту хочете обміняти:

GRN

5. На яку валюту хочете поміняти:

GRN

7. Введіть суму валюту, яка підлягає обміну:

1000

9. Після виконання операції обміну валют у Вас є в

GRN	17000,00	USD	2328,69
EUR	1247,06	PLN	5200,00

6. Виберіть банк чи обмінний пункт:

НБУ

ПриватБанк

Монобанк

Варшавка

Luchesk

Grand Volyn

8. Натисніть тут, щоб виконати операцію обміну валюти!

10. Натисніть тут, щоб зберегти запис у БД

11. Натисніть тут, щоб оновити БД

12. Хотите продовжити?

ТАК НІ

Перейти на форму Банки

Перейти на форму Обмінники

Перейти на форму Обмін валют

Перейти на форму Робота з БД

Вихід

14. Натисніть тут, щоб продовжити чи вийти

Підтвердження виконання команди додавання запису.

Команда додавання запису виконана! Виконайте пункт 11!

OK

Рисунок 3.57 – Збереження запису про клієнта у записі БД

ВИКОНАННЯ ОПЕРАЦІЙ З ОБМІНУ ВАЛЮТ З ПІДКЛЮЧЕННЯМ БАЗИ ДАНИХ

2

1. Введіть у вікно, що зверху, Іd клієнта і натисніть цю кнопку, щоб отримати його дані!

1 Коць Богдан Сергійович	20000	2000	1000	5000
2 Кузьмюк Олександр Васильович	17000,00	2328,69	1247,06	5200,00
3 Серський Дмитро Сергійович	23000	2300	1300	5300
4 Климець Владислав Сергійович	24000	2400	1400	5400

Кузьмюк Олександр Васильович

GRN	18000,00	USD	2302,92
EUR	1247,06	PLN	5200,00

2. Натисніть цю кнопку, щоб зберегти обсяги валют, яка є у Вас в наявності!

3. Перед виконанням операції обміну валют у Вас є в наявності:

GR	18000,00	USD	2302,92
EUR	1247,06	PLN	5200,00

4. Яку валюту хочете обміняти:

GRN

5. На яку валюту хочете поміняти:

GRN

7. Введіть суму валюту, яка підлягає обміну:

1000

9. Після виконання операції обміну валют у Вас є в

GRN	17000,00	USD	2328,69
EUR	1247,06	PLN	5200,00

6. Виберіть банк чи обмінний пункт:

НБУ

ПриватБанк

Монобанк

Варшавка

Luchesk

Grand Volyn

8. Натисніть тут, щоб виконати операцію обміну валюти!

10. Натисніть тут, щоб зберегти запис у БД

11. Натисніть тут, щоб оновити БД

12. Хотите продовжити?

ТАК НІ

Перейти на форму Банки

Перейти на форму Обмінники

Перейти на форму Обмін валют

Перейти на форму Робота з БД

Вихід

14. Натисніть тут, щоб продовжити чи вийти

Підтвердження виконання команди оновлення ...

Команда оновлення БД виконана!

OK

Рисунок 3.58 – Оновлення БД

ВИКОНАННЯ ОПЕРАЦІЙ З ОБМІНУ ВАЛЮТ З ПІДКЛЮЧЕННЯМ БАЗИ ДАНИХ

2

1. Введіть у вікно, що зверху, Іd клієнта і натисніть цю кнопку, щоб отримати його дані!

1 Коць Богдан Сергійович	20000	2000	1000	5000
2 Кузьмюк Олександр Васильович	17000,00	2328,69	1247,06	5200,00
3 Серський Дмитро Сергійович	23000	2300	1300	5300
4 Климець Владислав Сергійович	24000	2400	1400	5400

Кузьмюк Олександр Васильович

GRN	18000,00	USD	2302,92
EUR	1247,06	PLN	5200,00

2. Натисніть цю кнопку, щоб зберегти обсяги валют, яка є у Вас в наявності!

3. Перед виконанням операції обміну валют у Вас є в наявності:

GR	18000,00	USD	2302,92
EUR	1247,06	PLN	5200,00

4. Яку валюту хочете обміняти:

GRN
 USD
 EUR
 PLN

5. На яку валюту хочете поміняти:

GRN
 USD
 EUR
 PLN

6. Виберіть банк чи обмінний пункт:

НБУ
 ПриватБанк
 Монобанк
 Варшавка
 Luchesk
 Grand Volyn

Перейти на форму Банки
Перейти на форму Обмінники
Перейти на форму Обмін валют

7. Введіть суму валюти, яка підлягає обміну:

1000

8. Натисніть тут, щоб виконати операцію обміну валюти!

Перейти на форму Робота з БД

9. Після виконання операції обміну валют у Вас є в

10. Натисніть тут, щоб зберегти запис у БД

11. Натисніть тут, щоб оновити БД

Вихід

GRN	17000,00	USD	2328,69
EUR	1247,06	PLN	5200,00

12. Хочете продовжити?

ТАК НІ

14. Натисніть тут, щоб продовжити чи вийти

Рисунок 3.59 – Вибір щодо продовження виконання операцій обміну чи виходу з програми

ВИКОНАННЯ ОПЕРАЦІЙ З ОБМІНУ ВАЛЮТ З ПІДКЛЮЧЕННЯМ БАЗИ ДАНИХ

2

1. Введіть у вікно, що зверху, Іd клієнта і натисніть цю кнопку, щоб отримати його дані!

1 Коць Богдан Сергійович	20000	2000	1000	5000
2 Кузьмюк Олександр Васильович	17000,00	2328,69	1247,06	5200,00
3 Серський Дмитро Сергійович	23000	2300	1300	5300
4 Климець Владислав Сергійович	24000	2400	1400	5400

Кузьмюк Олександр Васильович

GRN	17000,00	USD	2328,69
EUR	1247,06	PLN	5200,00

2. Натисніть цю кнопку, щоб зберегти обсяги валют, яка є у Вас в наявності!

3. Перед виконанням операції обміну валют у Вас є в наявності:

GR	17000,00	USD	2328,69
EUR	1247,06	PLN	5200,00

4. Яку валюту хочете обміняти:

GRN
 USD
 EUR
 PLN

5. На яку валюту хочете поміняти:

GRN
 USD
 EUR
 PLN

6. Виберіть банк чи обмінний пункт:

НБУ
 ПриватБанк
 Монобанк
 Варшавка
 Luchesk
 Grand Volyn

Перейти на форму Банки
Перейти на форму Обмінники
Перейти на форму Обмін валют

7. Введіть суму валюти, яка підлягає обміну:

0

8. Натисніть тут, щоб виконати операцію обміну валюти!

Перейти на форму Робота з БД

9. Після виконання операції обміну валют у Вас є в

10. Натисніть тут, щоб зберегти запис у БД

11. Натисніть тут, щоб оновити БД

Вихід

GRN	17000,00	USD	2328,69
EUR	1247,06	PLN	5200,00

12. Хочете продовжити?

ТАК НІ

14. Натисніть тут, щоб продовжити чи вийти

Рисунок 3.60 – Вигляд форми Form5 у разі продовження виконання операцій обміну

ВИСНОВКИ

Всі завдання, вирішення яких передбачалися у кваліфікаційній роботі бакалавра, виконані у повному обсязі.

Зокрема, у кваліфікаційній роботі бакалавра:

1. Виконано огляд програмних засобів для розробки інформаційної системи пункту обміну валют у вигляді програмного проекту засобами C# WinForms та обгрунтовано їх вибір для вирішення завдань проекту.
2. Розроблено засобами C# WinForms проект у складі п'яти форм.
3. Розроблено форму «Банки» з інформацією про курси валют у банківських установах м. Луцька. В результаті клієнти пункту обміну валют дістали можливість оперативно отримувати інформацію про курси валют.
4. Розроблено форму «Обмінники» з інформацією про курси валют в обмінниках м. Луцька.
5. Розроблено форму «Обмін валют» для виконання операцій з обміну валют у разі ручного введення вхідних даних.
6. Розроблено форму «Робота з базою даних» для роботи з базою даних з інформацією про клієнтів.
7. Розроблено форму «Обмін валют з підключенням бази даних» для виконання операцій з обміну валют з підключенням бази даних з інформацією про клієнтів.

Програмний проект відлагоджений та протестований, в результаті чого було підтверджено факт його коректної роботи.

Розроблена у кваліфікаційній роботі інформаційна система пункту обміну валют у вигляді програмного проекту засобами C# WinForms може бути використана для оптимізації роботи реального обмінного пункту та у навчальному процесі.

СПИСКИ ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Учасники проєктів Вікімедіа. Інформаційна система – Вікіпедія. Вікіпедія. URL: https://uk.wikipedia.org/wiki/Інформаційна_система (дата звернення: 10.04.2024).
2. Про захист інформації в інформаційно-комунікаційних системах. Офіційний вебпортал парламенту України. URL: <https://zakon.rada.gov.ua/laws/show/80/94-вр> (дата звернення: 10.04.2024).
3. Учасники проєктів Вікімедіа. Мова програмування – Вікіпедія. Вікіпедія. URL: https://uk.wikipedia.org/wiki/Мова_програмування (дата звернення: 10.04.2024).
4. The Python Tutorial. Python documentation. URL: <https://docs.python.org/3/tutorial/> (дата звернення: 11.04.2024).
5. Янковський А. C# book/ : навч. посібн. Київ, 2021. 64 с.
6. Dirk Strauss C# Features Succinctly. Syncfusion , 2023. 216 с.
7. Jason Alls Clean Code in C# - Birmingham, 2020. 487 с.
8. Коноваленко І.В., Марущак П.О. Платформа .NET та мова програмування C# 8.0: навч. посібн. Тернопіль: ФОП Паляниця В.А., 2020. 320 с.
9. .NET | Build. Test. Deploy. Microsoft. URL: <https://dotnet.microsoft.com/en-us/> (дата звернення: 15.05.2024).
10. 1.1. Що таке .NET? – Про Програмування. Про Програмування – ABout IT And Programming. URL: <https://abitap.com/1-1/> (дата звернення: 16.04.2024).
11. Joseph A. C# 10 IN A NUTSHELL THE DEFINITIVE REFERENCE. O'Reilly Media. 2022. 1024 с.
12. Павловський В.І., Петрашенко А.В., Победа Д.В. Бази даних та засоби управління. Практикум: навч. посіб. для студ. спеціальності 123 – Комп'ютерна інженерія. КПІ ім. Ігоря Сікорського, Київ, 2021. 112 с.

13. Учасники проектів Вікімедіа. Алгоритм – Вікіпедія. Вікіпедія. URL: <https://uk.wikipedia.org/wiki/Алгоритм> (дата звернення: 16.04.2024).
14. Крєневич А.П. Алгоритми і структури даних. Підручник. К.: ВПЦ «Київський Університет», 2021. 200 с.
15. Пех П.А. Програмування: Конспект лекцій для здобувачів першого (бакалаврського) рівня освітньо-професійної програми «Комп'ютерна інженерія» галузі знань 12 «Інформаційні технології» спеціальності 123 «Комп'ютерна інженерія» денної та заочної форм навчання. Луцьк: ЛНТУ, 2020. 146 с.
16. Database Design Succinctly / J. Bieringer et al. Syncfusion, Inc., 2022. 87 с.
17. Aditya B. Grokking Algorithms. An illustrated guide for programmers and other curious people / ed. by О. Плаксий ; trans. from англ by О. Медвед. ArtHuss, 2023. 256 с.
18. Martin R. Clean Coder: A Code of Conduct for Professional Programmers. Pearson Education, Limited, 2023. 256 с.
19. Winters T., Manshreck T., Wright H. Software Engineering at Google: Lessons Learned from Programming Over Time. O'Reilly Media, 2020. 602 с.
20. Thompson C. Coders: Who They Are, What They Think and How They Are Changing Our World. Picador, 2020. 448 с.
21. Zimmerman C. Rules of Programming: How to Write Better Code. O'Reilly Media, Incorporated, 2023. 343 с.
22. Nichols C., Klabnik S. Rust Programming Language, 2nd Edition. No Starch Press, Incorporated, 2023. 528 с.
23. Dk Publishing. Beginner's Step-By-Step Coding Course. Kindersley Ltd., Dorling, 2020. 360 с.
24. Фрімен Е., Робсон Е. Head First. Патерни проектування. 2020. 672 с.
25. Brian “Beej Jorgensen” Hall. Beej’s Guide to C Programming. 2023. 309 с.

ДОДАТКИ

Додаток А

Програмний код форми Form3 «Обмін валют»

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using static System.Windows.Forms.VisualStyleElement;
using static
System.Windows.Forms.VisualStyleElement.Button;

namespace BachelorWork
{

public partial class Form3 : Form
{

double nb_rateUSDtoGRN = Rate.nb_rateUSDtoGRN_sell; // Нац Банк.
Продати 1 долар на гривні
double nb_rateGRNtoUSD = Rate.nb_rateGRNtoUSD_buy; // Нац Банк.
Купити 1 долар за гривні

double pb_rateUSDtoGRN = Rate.pb_rateUSDtoGRN_sell; // Приват Банк.
Продати 1 долар на гривні

```

double pb_rateGRNtoUSD = Rate.pb_rateGRNtoUSD_buy; // Приват Банк.

Купити 1 долар за гривні

double mb_rateUSDtoGRN = Rate.mb_rateUSDtoGRN_sell; // Моно Банк.

Продати 1 долар на гривні

double mb_rateGRNtoUSD = Rate.mb_rateGRNtoUSD_buy; // Моно Банк.

Купити 1 долар за гривні

double va_rateUSDtoGRN = Rate.va_rateUSDtoGRN_sell; // Варшавка.

Продати 1 долар на гривні

double va_rateGRNtoUSD = Rate.va_rateGRNtoUSD_buy; // Варшавка.

Купити 1 долар за гривні

double lu_rateUSDtoGRN = Rate.lu_rateUSDtoGRN_sell; // Лучеськ.

Продати 1 долар на гривні

double lu_rateGRNtoUSD = Rate.lu_rateGRNtoUSD_buy; // Лучеськ.

Купити 1 долар за гривні

double gr_rateUSDtoGRN = Rate.gr_rateUSDtoGRN_sell; // Гранд Волинь.

Продати 1 долар на гривні

double gr_rateGRNtoUSD = Rate.gr_rateGRNtoUSD_buy; // Гранд

Волинь. Купити 1 долар за гривні

//-----

double nb_rateEURtoGRN = Rate.nb_rateEURtoGRN_sell; // Нац Банк.

Продати 1 євро на гривні

double nb_rateGRNtoEUR = Rate.nb_rateGRNtoEUR_buy; // Нац Банк.

Купити 1 євро за гривні

double pb_rateEURtoGRN = Rate.pb_rateEURtoGRN_sell; // Приват Банк.

Продати 1 євро на гривні

double pb_rateGRNtoEUR = Rate.pb_rateGRNtoEUR_buy; // Приват Банк.

Купити 1 євро за гривні

double mb_rateEURtoGRN = Rate.mb_rateEURtoGRN_sell; // Моно Банк.

Продати 1 євро на гривні

double mb_rateGRNtoEUR = Rate.mb_rateGRNtoEUR_buy; // Моно Банк.

Купити 1 євро за гривні

double va_rateEURtoGRN = Rate.va_rateEURtoGRN_sell; // Варшавка.

Продати 1 євро на гривні

double va_rateGRNtoEUR = Rate.va_rateGRNtoEUR_buy; // Варшавка.

Купити 1 євро за гривні

double lu_rateEURtoGRN = Rate.lu_rateEURtoGRN_sell; // Лучеськ.

Продати 1 євро на гривні

double lu_rateGRNtoEUR = Rate.lu_rateGRNtoEUR_buy; // Лучеськ.

Купити 1 долар за гривні

double gr_rateEURtoGRN = Rate.gr_rateEURtoGRN_sell; // Гранд Волинь.

Продати 1 євро на гривні

double gr_rateGRNtoEUR = Rate.gr_rateGRNtoEUR_buy; // Гранд

Волинь. Купити 1 євро за гривні

//-----

double nb_ratePLNtoGRN = Rate.nb_ratePLNtoGRN_sell; // Нац Банк.

Продати 1 злот на гривні

double nb_rateGRNtoPLN = Rate.nb_rateGRNtoPLN_buy; // Нац Банк.

Купити 1 злот за гривні

double pb_ratePLNtoGRN = Rate.pb_ratePLNtoGRN_sell; // Приват Банк.

Продати 1 злот на гривні

double pb_rateGRNtoPLN = Rate.pb_rateGRNtoPLN_buy; // Приват Банк.

Купити 1 злот за гривні

double mb_ratePLNtoGRN = Rate.mb_ratePLNtoGRN_sell; // Моно Банк.

Продати 1 злот на гривні

double mb_rateGRNtoPLN = Rate.mb_rateGRNtoPLN_buy; // Моно Банк.

Купити 1 злот за гривні

double va_ratePLNtoGRN = Rate.va_ratePLNtoGRN_sell; // Варшавка.

Продати 1 злот на гривні

double va_rateGRNtoPLN = Rate.va_rateGRNtoPLN_buy; // Варшавка.

Купити 1 злот за гривні

double lu_ratePLNtoGRN = Rate.lu_ratePLNtoGRN_sell; // Лучеськ.

Продати 1 злот на гривні

double lu_rateGRNtoPLN = Rate.lu_rateGRNtoPLN_buy; // Лучеськ.

Купити 1 злот за гривні

double gr_ratePLNtoGRN = Rate.gr_ratePLNtoGRN_sell; // Гранд Волинь.

Продати 1 злот на гривні

double gr_rateGRNtoPLN = Rate.gr_rateGRNtoPLN_buy; // Гранд Волинь.

Купити 1 злот за гривні

//-----

```
DateTime date1 = new DateTime();
DateTime date2 = new DateTime();

double grnN;
double usdN;
double eurN;
double plnN;
double rate, rez;
double ExchangeSum;

double grnLim = 40; // Мінімальна сума обміну у гривнях
double usdLim = 10; // Мінімальна сума обміну у доларах
double eurLim = 10; // Мінімальна сума обміну у євро
double plnLim = 20; // Мінімальна сума обміну у злотих

bool v = true; // Перша операція обміну валют дозволена

public Form3()
{
    InitializeComponent();

    textBox13.Text = "0";
}

// Введення та збереження обсягів валюти, яка є в наявності
private void button4_Click(object sender, EventArgs e)
{
    grnN = Double.Parse(textBox1.Text);
    usdN = Double.Parse(textBox2.Text);
```

```

eurN = Double.Parse(textBox3.Text);
plnN = Double.Parse(textBox4.Text);

```

```

textBox5.Text = grnN.ToString("0.00");
textBox6.Text = usdN.ToString("0.00");
textBox7.Text = eurN.ToString("0.00");
textBox8.Text = plnN.ToString("0.00");
radioButton18.Checked = true;
}

```

```

private void button1_Click(object sender, EventArgs e)
{
this.Hide();
Form1 f1 = new Form1();
f1.Show();
}

```

// Продовження виконання операцій обміну валют

```

private void button6_Click(object sender, EventArgs e)
{

```

```

if (radioButton18.Checked) // Закінчення виконання операцій обміну

```

ВАЛЮТ

```

{
this.Hide();
Form1 f1 = new Form1();
f1.Show(); // Перехід на голону форму
}

```

```

v = true; // Операція обміну валют ще має бути виконана

```

```
// Відображення обсягів валют, що є в наявності
textBox1.Text = textBox9.Text;
textBox2.Text = textBox10.Text;
textBox3.Text = textBox11.Text;
textBox4.Text = textBox12.Text;

// Відображення обсягів валют,
// що є в наявності перед виконанням операції обміну
textBox5.Text = textBox9.Text;
textBox6.Text = textBox10.Text;
textBox7.Text = textBox11.Text;
textBox8.Text = textBox12.Text;
textBox13.Text = "0";
}

private void button2_Click(object sender, EventArgs e)
{
    this.Hide();
    Form2 f2 = new Form2();
    f2.Show();
}

private void button5_Click(object sender, EventArgs e)
{
    this.Hide();
    Form4 f4 = new Form4();
    f4.Show();
}

private void button7_Click(object sender, EventArgs e)
```

```
{
Application.Exit();
}

private void button8_Click(object sender, EventArgs e)
{
this.Hide();
Form5 f5 = new Form5();
f5.Show();
}

// Виконання операцій обміну валют
private void button3_Click(object sender, EventArgs e)
{
ExchangeSum = Double.Parse(textBox13.Text);

// Перевірка, чи не вибрано заборонені транзакції
if ((radioButton7.Checked && radioButton11.Checked) ||
(radioButton8.Checked && radioButton12.Checked) ||
(radioButton9.Checked && radioButton13.Checked) || (radioButton10.Checked &&
radioButton14.Checked)
|| (radioButton8.Checked && radioButton12.Checked) ||
(radioButton8.Checked && radioButton13.Checked) || (radioButton8.Checked &&
radioButton14.Checked)
|| (radioButton9.Checked && radioButton12.Checked) ||
(radioButton9.Checked && radioButton13.Checked) || (radioButton9.Checked &&
radioButton14.Checked))
```

```

    || (radioButton10.Checked && radioButton12.Checked) ||
(radioButton10.Checked && radioButton13.Checked) || (radioButton10.Checked &&
radioButton14.Checked)

    || (radioButton7.Checked && radioButton14.Checked &&
radioButton2.Checked) || (radioButton7.Checked && radioButton14.Checked &&
radioButton3.Checked)

    || (radioButton10.Checked && radioButton11.Checked &&
radioButton2.Checked) || (radioButton10.Checked && radioButton11.Checked &&
radioButton3.Checked))
    {

        MessageBox.Show("Вибрана Вами транзакція не передбачена!",
"Помилка вводу даних або вибору опцій", MessageBoxButtons.OK,
MessageBoxIcon.Information);

        MessageBox.Show("Увага! Правильно виберіть види валют та банк чи
обмінник!", "Помилка вводу даних або вибору опцій", MessageBoxButtons.OK,
MessageBoxIcon.Information);

        MessageBox.Show("Виконайте пункт 10, а потім пункт 11 даного
додатку!!", "Помилка вводу даних або вибору опцій", MessageBoxButtons.OK,
MessageBoxIcon.Information);

        v = false; // Операція обміну валют не може бути виконана
    }

    // Перевірка, чи сума гривень, яка підлягає обміну, не менша за
мінімальну дозволена і не більша за наявну суму
    else if (radioButton7.Checked && ((ExchangeSum <= grnLim) ||
(ExchangeSum > grnN)))
    {
    {

```

```

    MessageBox.Show("Сума гривень, яка підлягає обміну, менша за мінімальну дозволена або перевищує наявну!", "Помилка вводу даних", MessageBoxButtons.OK, MessageBoxIcon.Information);

```

```

    MessageBox.Show("Увага! Правильно введіть суму гривень, яка підлягає обміну!", "Помилка вводу даних", MessageBoxButtons.OK, MessageBoxIcon.Information);

```

```

    MessageBox.Show("Виконайте пункт 10, а потім пункт 11 даного додатку!", "Помилка вводу даних", MessageBoxButtons.OK, MessageBoxIcon.Information);

```

```

    }

```

```

    v = false; // Операція обміну валют не може бути виконана

```

```

    }

```

```

    // Перевірка, чи сума доларів, яка підлягає обміну, не менша за мінімальну дозволена і не більша за наявну суму

```

```

    else if (radioButton8.Checked && ((ExchangeSum <= usdLim) || (ExchangeSum > usdN)))

```

```

    {

```

```

    {

```

```

        MessageBox.Show("Сума доларів, яка підлягає обміну, менша за мінімальну дозволена або перевищує наявну!", "Помилка вводу даних", MessageBoxButtons.OK, MessageBoxIcon.Information);

```

```

        MessageBox.Show("Увага! Правильно введіть суму доларів, яка підлягає обміну!", "Помилка вводу даних", MessageBoxButtons.OK, MessageBoxIcon.Information);

```

```

        MessageBox.Show("Виконайте пункт 10, а потім пункт 11 даного додатку!", "Помилка вводу даних", MessageBoxButtons.OK, MessageBoxIcon.Information);

```

```

    }

```

```

    v = false; // Операція обміну валют не може бути виконана

```

```

    }

    // Перевірка, чи сума євро, яка підлягає обміну, не менша за мінімальну
    // дозволена і не більша за наявну суму
    else if (radioButton9.Checked && ((ExchangeSum <= eurLim) ||
    (ExchangeSum > eurN)))
    {
    {
    MessageBox.Show("Сума євро, яка підлягає обміну, менша за
    мінімальну дозволена або перевищує наявну!", "Помилка вводу даних",
    MessageBoxButtons.OK, MessageBoxIcon.Information);

    MessageBox.Show("Увага! Правильно введіть суму євро, яка підлягає
    обміну!", "Помилка вводу даних", MessageBoxButtons.OK,
    MessageBoxIcon.Information);

    MessageBox.Show("Виконайте пункт 10, а потім пункт 11 даного
    додатку!", "Помилка вводу даних", MessageBoxButtons.OK,
    MessageBoxIcon.Information);
    }
    v = false; // Операція обміну валют не може бути виконана
    }

    // Перевірка, чи сума злотих, яка підлягає обміну, не менша за
    // мінімальну дозволена і не більша за наявну суму
    else if (radioButton10.Checked && ((ExchangeSum <= plnLim) ||
    (ExchangeSum > plnN)))
    {
    {
    MessageBox.Show("Сума злотих, яка підлягає обміну, менша за
    мінімальну дозволена або перевищує наявну!", "Помилка вводу даних",
    MessageBoxButtons.OK, MessageBoxIcon.Information);

```

```

    MessageBox.Show("Увага! Правильно введіть суму злотих, яка підлягає
обміну!", "Помилка вводу даних", MessageBoxButtons.OK,
MessageBoxIcon.Information);

```

```

    MessageBox.Show("Виконайте пункт 10, а потім пункт 11 даного
додатку!", "Помилка вводу даних", MessageBoxButtons.OK,
MessageBoxIcon.Information);

```

```

    }

```

```

    v = false; // Операція обміну валют не може бути виконана

```

```

    }

```

```

// Обмін гривні на долари через Національний Банк

```

```

else if (radioButton1.Checked && radioButton7.Checked &&
radioButton12.Checked && v)

```

```

{

```

```

    rate = nb_rateGRNtoUSD;

```

```

    rez = ExchangeSum / rate;

```

```

    grnN -= ExchangeSum;

```

```

    usdN += rez;

```

```

    textBox9.Text = grnN.ToString("0.00");

```

```

    textBox10.Text = usdN.ToString("0.00");

```

```

    textBox11.Text = textBox7.Text;

```

```

    textBox12.Text = textBox8.Text;

```

```

}

```

```

// Обмін гривні на долари через ПриватБанк

```

```

else if (radioButton2.Checked && radioButton7.Checked &&
radioButton12.Checked && v)

```

```

{

```

```

    rate = pb_rateGRNtoUSD;

```

```

    rez = ExchangeSum / rate;

```

```
grnN -= ExchangeSum;
usdN += rez;
textBox9.Text = grnN.ToString("0.00");
textBox10.Text = usdN.ToString("0.00");
textBox11.Text = textBox7.Text;
textBox12.Text = textBox8.Text;
}
```

```
// Обмін гривні на долари через МоноБанк
```

```
else if (radioButton3.Checked && radioButton7.Checked &&
radioButton12.Checked && v)
```

```
{
rate = mb_rateGRNtoUSD;
rez = ExchangeSum / rate;
grnN -= ExchangeSum;
usdN += rez;
textBox9.Text = grnN.ToString("0.00");
textBox10.Text = usdN.ToString("0.00");
textBox11.Text = textBox7.Text;
textBox12.Text = textBox8.Text;
}
```

```
// Обмін гривні на долари через Варшавку
```

```
else if (radioButton4.Checked && radioButton7.Checked &&
radioButton12.Checked && v)
```

```
{
rate = va_rateGRNtoUSD;
rez = ExchangeSum / rate;
```

```
grnN -= ExchangeSum;
usdN += rez;
textBox9.Text = grnN.ToString("0.00");
textBox10.Text = usdN.ToString("0.00");
textBox11.Text = textBox7.Text;
textBox12.Text = textBox8.Text;
}
```

```
// Обмін гривні на долари через Лучеськ
```

```
else if (radioButton5.Checked && radioButton7.Checked &&
radioButton12.Checked && v)
```

```
{
rate = lu_rateGRNtoUSD;
rez = ExchangeSum / rate;
grnN -= ExchangeSum;
usdN += rez;
textBox9.Text = grnN.ToString("0.00");
textBox10.Text = usdN.ToString("0.00");
textBox11.Text = textBox7.Text;
textBox12.Text = textBox8.Text;
}
```

```
// Обмін гривні на долари через Гранд Волинь
```

```
else if (radioButton6.Checked && radioButton7.Checked &&
radioButton12.Checked && v)
```

```
{
rate = gr_rateGRNtoUSD;
rez = ExchangeSum / rate;
grnN -= ExchangeSum;
usdN += rez;
```

```

textBox9.Text = grnN.ToString("0.00");
textBox10.Text = usdN.ToString("0.00");
textBox11.Text = textBox7.Text;
textBox12.Text = textBox8.Text;
}

//-----
// Обмін гривні на євро через Національний Банк
if (radioButton1.Checked && radioButton7.Checked &&
radioButton13.Checked && v)
{
rate = nb_rateGRNtoEUR;
rez = ExchangeSum / rate;
grnN -= ExchangeSum;
eurN += rez;
textBox9.Text = grnN.ToString("0.00");
textBox11.Text = eurN.ToString("0.00");
textBox10.Text = textBox6.Text;
textBox12.Text = textBox8.Text;
}

// Обмін гривні на євро через ПриватБанк
else if (radioButton2.Checked && radioButton7.Checked &&
radioButton13.Checked && v)
{
rate = pb_rateGRNtoEUR;
rez = ExchangeSum / rate;
grnN -= ExchangeSum;
eurN += rez;
textBox9.Text = grnN.ToString("0.00");

```

```
textBox11.Text = eurN.ToString("0.00");  
textBox10.Text = textBox6.Text;  
textBox12.Text = textBox8.Text;  
}
```

```
// Обмін гривні на євро через МоноБанк
```

```
else if (radioButton3.Checked && radioButton7.Checked &&  
radioButton13.Checked && v)
```

```
{  
rate = mb_rateGRNtoEUR;  
rez = ExchangeSum / rate;  
grnN -= ExchangeSum;  
eurN += rez;  
textBox9.Text = grnN.ToString("0.00");  
textBox11.Text = eurN.ToString("0.00");  
textBox10.Text = textBox6.Text;  
textBox12.Text = textBox8.Text;  
}
```

```
// Обмін гривні на євро через Варшавку
```

```
else if (radioButton4.Checked && radioButton7.Checked &&  
radioButton13.Checked && v)
```

```
{  
rate = va_rateGRNtoEUR;  
rez = ExchangeSum / rate;  
grnN -= ExchangeSum;  
eurN += rez;  
textBox9.Text = grnN.ToString("0.00");  
textBox11.Text = eurN.ToString("0.00");  
textBox10.Text = textBox6.Text;
```

```
textBox12.Text = textBox8.Text;
}

// Обмін гривні на євро через Лучеськ
else if (radioButton5.Checked && radioButton7.Checked &&
radioButton13.Checked && v)
{
    rate = lu_rateGRNtoEUR;
    rez = ExchangeSum / rate;
    grnN -= ExchangeSum;
    eurN += rez;
    textBox9.Text = grnN.ToString("0.00");
    textBox11.Text = eurN.ToString("0.00");
    textBox10.Text = textBox6.Text;
    textBox12.Text = textBox8.Text;
}

// Обмін гривні на євро через Гранд Волинь
else if (radioButton6.Checked && radioButton7.Checked &&
radioButton13.Checked && v)
{
    rate = gr_rateGRNtoEUR;
    rez = ExchangeSum / rate;
    grnN -= ExchangeSum;
    eurN += rez;
    textBox9.Text = grnN.ToString("0.00");
    textBox11.Text = eurN.ToString("0.00");
    textBox10.Text = textBox6.Text;
    textBox12.Text = textBox8.Text;
}
```

```
// Обмін гривні на злоті через Національний Банк
if (radioButton1.Checked && radioButton7.Checked &&
radioButton14.Checked && v)
{
rate = nb_rateGRNtoPLN;
rez = ExchangeSum / rate;
grnN -= ExchangeSum;
plnN += rez;
textBox9.Text = grnN.ToString("0.00");
textBox12.Text = plnN.ToString("0.00");
textBox10.Text = textBox6.Text;
textBox11.Text = textBox7.Text;
}

// Обмін гривні на злоті через ПриватБанк
else if (radioButton2.Checked && radioButton7.Checked &&
radioButton14.Checked && v)
{
rate = pb_rateGRNtoPLN;
rez = ExchangeSum / rate;
grnN -= ExchangeSum;
plnN += rez;
textBox9.Text = grnN.ToString("0.00");
textBox12.Text = plnN.ToString("0.00");
textBox10.Text = textBox6.Text;
textBox11.Text = textBox7.Text;
}

// Обмін гривні на злоті через МоноБанк
```

```
else if (radioButton3.Checked && radioButton7.Checked &&  
radioButton14.Checked && v)
```

```
{  
rate = mb_rateGRNtoPLN;  
rez = ExchangeSum / rate;  
grnN -= ExchangeSum;  
plnN += rez;  
textBox9.Text = grnN.ToString("0.00");  
textBox12.Text = plnN.ToString("0.00");  
textBox10.Text = textBox6.Text;  
textBox11.Text = textBox7.Text;  
}
```

```
// Обмін гривні на злоті через Варшавку
```

```
else if (radioButton4.Checked && radioButton7.Checked &&  
radioButton14.Checked && v)
```

```
{  
rate = va_rateGRNtoPLN;  
rez = ExchangeSum / rate;  
grnN -= ExchangeSum;  
plnN += rez;  
textBox9.Text = grnN.ToString("0.00");  
textBox12.Text = plnN.ToString("0.00");  
textBox10.Text = textBox6.Text;  
textBox11.Text = textBox7.Text;  
}
```

```
// Обмін гривні на злоті через Лучеськ
```

```
else if (radioButton5.Checked && radioButton7.Checked &&  
radioButton14.Checked && v)
```

```

{
rate = lu_rateGRNtoPLN;
rez = ExchangeSum / rate;
grnN -= ExchangeSum;
plnN += rez;
textBox9.Text = grnN.ToString("0.00");
textBox12.Text = plnN.ToString("0.00");
textBox10.Text = textBox6.Text;
textBox11.Text = textBox7.Text;
}

```

// Обмін гривні на злоті через Гранд Волинь

```

else if (radioButton6.Checked && radioButton7.Checked &&
radioButton14.Checked && v)

```

```

{
rate = gr_rateGRNtoPLN;
rez = ExchangeSum / rate;
grnN -= ExchangeSum;
plnN += rez;
textBox9.Text = grnN.ToString("0.00");
textBox12.Text = plnN.ToString("0.00");
textBox10.Text = textBox6.Text;
textBox11.Text = textBox7.Text;
}

```

// Обмін доларів на гривні через Національний Банк

```

if (radioButton1.Checked && radioButton8.Checked &&
radioButton11.Checked && v)

```

```

{
rate = nb_rateUSDtoGRN;

```

```
rez = ExchangeSum * rate;
grnN += rez;
usdN -= ExchangeSum;
textBox9.Text = grnN.ToString("0.00");
textBox10.Text = usdN.ToString("0.00");
textBox11.Text = textBox7.Text;
textBox12.Text = textBox8.Text;
}

// Обмін доларів на гривні через ПриватБанк
else if (radioButton2.Checked && radioButton8.Checked &&
radioButton11.Checked && v)
{
rate = pb_rateUSDtoGRN;
rez = ExchangeSum * rate;
grnN += rez;
usdN -= ExchangeSum;
textBox9.Text = grnN.ToString("0.00");
textBox10.Text = usdN.ToString("0.00");
textBox11.Text = textBox7.Text;
textBox12.Text = textBox8.Text;
}

// Обмін доларів на гривні через МоноБанк
else if (radioButton3.Checked && radioButton8.Checked &&
radioButton11.Checked && v)
{
rate = mb_rateUSDtoGRN;
rez = ExchangeSum * rate;
```

```
grnN += rez;  
usdN -= ExchangeSum;  
textBox9.Text = grnN.ToString("0.00");  
textBox10.Text = usdN.ToString("0.00");  
textBox11.Text = textBox7.Text;  
textBox12.Text = textBox8.Text;  
}
```

```
// Обмін доларів на гривні через Варшавку
```

```
else if (radioButton4.Checked && radioButton8.Checked &&  
radioButton1.Checked && v)
```

```
{  
rate = va_rateUSDtoGRN;  
rez = ExchangeSum * rate;  
grnN += rez;  
usdN -= ExchangeSum;  
textBox9.Text = grnN.ToString("0.00");  
textBox10.Text = usdN.ToString("0.00");  
textBox11.Text = textBox7.Text;  
textBox12.Text = textBox8.Text;  
}
```

```
// Обмін доларів на гривні через Лучеськ
```

```
else if (radioButton5.Checked && radioButton8.Checked &&  
radioButton11.Checked && v)
```

```
{  
rate = lu_rateUSDtoGRN;  
rez = ExchangeSum * rate;  
grnN += rez;  
usdN -= ExchangeSum;
```

```
textBox9.Text = grnN.ToString("0.00");
textBox10.Text = usdN.ToString("0.00");
textBox11.Text = textBox7.Text;
textBox12.Text = textBox8.Text;
}
```

```
// Обмін доларів на гривні через Гранд Волинь
```

```
else if (radioButton6.Checked && radioButton8.Checked &&
radioButton11.Checked && v)
```

```
{
rate = gr_rateUSDtoGRN;
rez = ExchangeSum * rate;
grnN += rez;
usdN -= ExchangeSum;
textBox9.Text = grnN.ToString("0.00");
textBox10.Text = usdN.ToString("0.00");
textBox11.Text = textBox7.Text;
textBox12.Text = textBox8.Text;
}
```

```
// Обмін євро на гривні через Національний Банк
```

```
if (radioButton1.Checked && radioButton9.Checked &&
radioButton11.Checked && v)
```

```
{
rate = nb_rateEURtoGRN;
rez = ExchangeSum * rate;
grnN += rez;
eurN -= ExchangeSum;
textBox9.Text = grnN.ToString("0.00");
textBox11.Text = eurN.ToString("0.00");
```

```
textBox10.Text = textBox6.Text;
textBox12.Text = textBox8.Text;
}

// Обмін євро на гривні через ПриватБанк
else if (radioButton2.Checked && radioButton9.Checked &&
radioButton11.Checked && v)
{
    rate = pb_rateEURtoGRN;
    rez = ExchangeSum * rate;
    grnN += rez;
    eurN -= ExchangeSum;
    textBox9.Text = grnN.ToString("0.00");
    textBox11.Text = eurN.ToString("0.00");
    textBox10.Text = textBox6.Text;
    textBox12.Text = textBox8.Text;
}

// Обмін євро на гривні через МоноБанк
else if (radioButton3.Checked && radioButton9.Checked &&
radioButton11.Checked && v)
{
    rate = mb_rateEURtoGRN;
    rez = ExchangeSum * rate;
    grnN += rez;
    eurN -= ExchangeSum;
    textBox9.Text = grnN.ToString("0.00");
    textBox11.Text = eurN.ToString("0.00");
    textBox10.Text = textBox6.Text;
    textBox12.Text = textBox8.Text;
```

```
}

// Обмін євро на гривні через Варшавку
else if (radioButton4.Checked && radioButton9.Checked &&
radioButton11.Checked && v)
{
    rate = va_rateEURtoGRN;
    rez = ExchangeSum * rate;
    grnN += rez;
    eurN -= ExchangeSum;
    textBox9.Text = grnN.ToString("0.00");
    textBox11.Text = eurN.ToString("0.00");
    textBox10.Text = textBox6.Text;
    textBox12.Text = textBox8.Text;
}

// Обмін євро на гривні через Лучеськ
else if (radioButton5.Checked && radioButton9.Checked &&
radioButton11.Checked && v)
{
    rate = lu_rateEURtoGRN;
    rez = ExchangeSum * rate;
    grnN += rez;
    eurN -= ExchangeSum;
    textBox9.Text = grnN.ToString("0.00");
    textBox11.Text = eurN.ToString("0.00");
    textBox10.Text = textBox6.Text;
    textBox12.Text = textBox8.Text;
}
```

```

// Обмін євро на гривні через Гранд Волинь
else if (radioButton6.Checked && radioButton9.Checked &&
radioButton11.Checked && v)
{
rate = gr_rateEURtoGRN;
rez = ExchangeSum * rate;
grnN += rez;
eurN -= ExchangeSum;
textBox9.Text = grnN.ToString("0.00");
textBox11.Text = eurN.ToString("0.00");
textBox10.Text = textBox6.Text;
textBox12.Text = textBox8.Text;
}

// Обмін злотих на гривні через Національний Банк
if (radioButton1.Checked && radioButton10.Checked &&
radioButton11.Checked && v)
{
rate = nb_ratePLNtoGRN;
rez = ExchangeSum * rate;
grnN += rez;
plnN -= ExchangeSum;
textBox9.Text = grnN.ToString("0.00");
textBox12.Text = plnN.ToString("0.00");
textBox10.Text = textBox6.Text;
textBox11.Text = textBox7.Text;
}

// Обмін злотих на гривні через ПриватБанк

```

```
else if (radioButton2.Checked && radioButton10.Checked &&  
radioButton11.Checked && v)
```

```
{  
rate = pb_ratePLNtoGRN;  
rez = ExchangeSum * rate;  
grnN += rez;  
plnN -= ExchangeSum;  
textBox9.Text = grnN.ToString("0.00");  
textBox12.Text = plnN.ToString("0.00");  
textBox10.Text = textBox6.Text;  
textBox11.Text = textBox7.Text;  
}
```

```
// Обмін злотих на гривні через МоноБанк
```

```
else if (radioButton3.Checked && radioButton10.Checked &&  
radioButton11.Checked && v)
```

```
{  
rate = mb_ratePLNtoGRN;  
rez = ExchangeSum * rate;  
grnN += rez;  
plnN -= ExchangeSum;  
textBox9.Text = grnN.ToString("0.00");  
textBox12.Text = plnN.ToString("0.00");  
textBox10.Text = textBox6.Text;  
textBox11.Text = textBox7.Text;  
}
```

```
// Обмін злотих на гривні через Варшавку
```

```
else if (radioButton4.Checked && radioButton10.Checked &&  
radioButton11.Checked && v)
```

```

{
rate = va_ratePLNtoGRN;
rez = ExchangeSum * rate;
grnN += rez;
plnN -= ExchangeSum;
textBox9.Text = grnN.ToString("0.00");
textBox12.Text = plnN.ToString("0.00");
textBox10.Text = textBox6.Text;
textBox11.Text = textBox7.Text;
}

```

// Обмін злотих на гривні через Лучеськ

```

else if (radioButton5.Checked && radioButton10.Checked &&
radioButton11.Checked && v)

```

```

{
rate = lu_ratePLNtoGRN;
rez = ExchangeSum * rate;
grnN += rez;
plnN -= ExchangeSum;
textBox9.Text = grnN.ToString("0.00");
textBox12.Text = plnN.ToString("0.00");
textBox10.Text = textBox6.Text;
textBox11.Text = textBox7.Text;
}

```

// Обмін злотих на гривні через Гранд Волинь

```

else if (radioButton6.Checked && radioButton10.Checked &&
radioButton11.Checked && v)

```

```

{
rate = gr_ratePLNtoGRN;

```

```
rez = ExchangeSum * rate;
grnN += rez;
plnN -= ExchangeSum;
textBox9.Text = grnN.ToString("0.00");
textBox12.Text = plnN.ToString("0.00");
textBox10.Text = textBox6.Text;
textBox11.Text = textBox7.Text;
}

v = false; // Операція обміну валют вже виконана
```

```
radioButton18.Checked = true;
```

```
MessageBox.Show("Операція обміну валют виконана! Проаналізуйте  
отримані результати!", "Інформація про операцію обміну валют",  
MessageBoxButtons.OK, MessageBoxIcon.Information);
```

```
MessageBox.Show("Виконайте пункт 10, а потім пункт 11 даного  
додатку!", "Порада щодо подальших дій", MessageBoxButtons.OK,  
MessageBoxIcon.Information);
```

```
}  
}  
}
```

Додаток Б
Програмний код форми Form4 «Робота з БД»

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

namespace BachelorWork
{
    public partial class Form4 : Form
    {
        SqlConnection sqlConnection;

        public Form4()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            this.Hide();
            Form1 f1 = new Form1();
            f1.Show();
        }
    }
}
```

```
}

```

```
private void button2_Click(object sender, EventArgs e)
{
    this.Hide();
    Form2 f2 = new Form2();
    f2.Show();
}

```

```
private void button5_Click(object sender, EventArgs e)
{
    this.Hide();
    Form3 f3 = new Form3();
    f3.Show();
}

```

```
private void button7_Click(object sender, EventArgs e)
{
    Application.Exit();
}

```

```
private async void Form4_Load(object sender, EventArgs e)
{
    string connectionString = @"Data
Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=""C:\Users\Петро
Пех\Desktop\Kots Bohdan
2024\BachelorWork\BachelorWork\Database1.mdf"";Integrated Security=True";
    sqlConnection = new SqlConnection(connectionString);
    await sqlConnection.OpenAsync();
    SqlDataReader sqlReader = null;

```

```

SqlCommand command = new SqlCommand("SELECT * FROM [Table]",
sqlConnection);

try
{
sqlReader = await command.ExecuteReaderAsync();
while (await sqlReader.ReadAsync())
{
listBox1.Items.Add(Convert.ToString(sqlReader["Id"]) + " " +
Convert.ToString(sqlReader["Name"]) + " " + Convert.ToString(sqlReader["GRN"])
+ " " + Convert.ToString(sqlReader["USD"]) + " " +
Convert.ToString(sqlReader["EUR"]) + " " + Convert.ToString(sqlReader["PLN"]));
}
}
catch (Exception ex)
{
MessageBox.Show(ex.Message.ToString(), ex.Source.ToString(),
MessageBoxButtons.OK, MessageBoxIcon.Error);
}
finally
{
if (sqlReader != null)
sqlReader.Close();
}
}

private void вихідToolStripMenuItem_Click(object sender, EventArgs e)
{
if (sqlConnection != null && sqlConnection.State !=
ConnectionState.Closed)
sqlConnection.Close();
}

```

```

    }

    private void Form4_FormClosing(object sender, FormClosingEventArgs e)
    {
        if (sqlConnection != null && sqlConnection.State !=
ConnectionState.Closed)
            sqlConnection.Close();
    }

    private async void button3_Click(object sender, EventArgs e)
    {
        if (label7.Visible)
            label7.Visible = false;
        if (!string.IsNullOrEmpty(textBox1.Text) &&
!string.IsNullOrEmpty(textBox2.Text) &&
!string.IsNullOrEmpty(textBox3.Text) &&
!string.IsNullOrEmpty(textBox4.Text) &&
!string.IsNullOrEmpty(textBox5.Text) &&
!string.IsNullOrEmpty(textBox1.Text) &&
!string.IsNullOrEmpty(textBox2.Text) &&
!string.IsNullOrEmpty(textBox3.Text) &&
!string.IsNullOrEmpty(textBox4.Text) &&
!string.IsNullOrEmpty(textBox5.Text))
        {
            SqlCommand command = new SqlCommand("INSERT INTO [Table]
(Name, GRN, USD, EUR, PLN)VALUES(@Name, @GRN, @USD, @EUR,
@PLN)", sqlConnection);

            command.Parameters.AddWithValue("Name", textBox1.Text);

```

```

command.Parameters.AddWithValue("GRN", textBox2.Text);
command.Parameters.AddWithValue("USD", textBox3.Text);
command.Parameters.AddWithValue("EUR", textBox4.Text);
command.Parameters.AddWithValue("PLN", textBox5.Text);
await command.ExecuteNonQueryAsync();

MessageBox.Show("Команда INSERT виконана!", "Підтвердження
виконання команди INSERT.", MessageBoxButtons.OK,
MessageBoxIcon.Information);
    }
    else
    {
        label7.Visible = true;
        label7.Text = "Поля 'Name', 'GRN', 'USD', 'EUR', 'PLN' повинні бути
заповнені!";
    }

}

private async void обновиТИToolStripMenuItem_Click(object sender,
EventArgs e)
{
    listBox1.Items.Clear();

    SqlDataReader sqlReader = null;
    SqlCommand command = new SqlCommand("SELECT * FROM [Table]",
sqlConnection);
    try
    {
        sqlReader = await command.ExecuteReaderAsync();

```

```

while (await sqlReader.ReadAsync())
{
    listBox1.Items.Add(Convert.ToString(sqlReader["Id"]) + " " +
Convert.ToString(sqlReader["Name"]) + " " + Convert.ToString(sqlReader["GRN"])
+ " " + Convert.ToString(sqlReader["USD"]) + " " +
Convert.ToString(sqlReader["EUR"]) + " " + Convert.ToString(sqlReader["PLN"]));
}
}
catch (Exception ex)
{
    MessageBox.Show(ex.Message.ToString(), ex.Source.ToString(),
MessageBoxButtons.OK, MessageBoxIcon.Error);
}
finally
{
    if (sqlReader != null)
        sqlReader.Close();
}

}

private async void button4_Click(object sender, EventArgs e)
{
    if (label8.Visible)
        label8.Visible = false;

    if (!string.IsNullOrEmpty(textBox17.Text) &&
!string.IsNullOrEmpty(textBox17.Text) &&
!string.IsNullOrEmpty(textBox10.Text) &&
!string.IsNullOrEmpty(textBox10.Text) &&

```

```

!string.IsNullOrEmpty(textBox9.Text) &&
!string.IsNullOrWhiteSpace(textBox9.Text) &&
!string.IsNullOrEmpty(textBox8.Text) &&
!string.IsNullOrWhiteSpace(textBox8.Text) &&
!string.IsNullOrEmpty(textBox7.Text) &&
!string.IsNullOrWhiteSpace(textBox7.Text) &&
!string.IsNullOrEmpty(textBox6.Text) &&
!string.IsNullOrWhiteSpace(textBox6.Text) )
    {
        SqlCommand command = new SqlCommand("UPDATE [Table] SET
[Name] = @Name, [GRN] = @GRN, [USD] = @USD, [EUR] = @EUR, [PLN] =
@PLN WHERE [Id] = @Id", sqlConnection);

        command.Parameters.AddWithValue("Id", textBox17.Text);
        command.Parameters.AddWithValue("Name", textBox10.Text);
        command.Parameters.AddWithValue("GRN", textBox9.Text);
        command.Parameters.AddWithValue("USD", textBox8.Text);
        command.Parameters.AddWithValue("EUR", textBox7.Text);
        command.Parameters.AddWithValue("PLN", textBox6.Text);
        await command.ExecuteNonQueryAsync();

        MessageBox.Show("Команда UPDATE виконана!", "Підтвердження
виконання команди UPDATE.", MessageBoxButtons.OK,
MessageBoxIcon.Information);
    }
    else if ( !string.IsNullOrEmpty(textBox17.Text) &&
!string.IsNullOrWhiteSpace(textBox17.Text) )
    {
        label8.Visible = true;
        label8.Text = "Id має бути заповнений!";
    }

```

```

    }
    else
    {
        label8.Visible = true;
        label8.Text = "Поля 'Name', 'GRN', 'USD', 'EUR', 'PLN' повинні бути
заповнені!";
    }

}

private async void button8_Click(object sender, EventArgs e)
{
    if (label20.Visible)
        label20.Visible = false;
    if (!string.IsNullOrEmpty(textBox16.Text) &&
!string.IsNullOrWhiteSpace(textBox16.Text))
    {
        SqlCommand command = new SqlCommand("DELETE FROM [Table]
WHERE [Id] = @Id", sqlConnection);

        command.Parameters.AddWithValue("Id", textBox16.Text);

        await command.ExecuteNonQueryAsync();
        MessageBox.Show("Команда DELETE виконана!", "Підтвердження
виконання команди DELETE.", MessageBoxButtons.OK,
MessageBoxIcon.Information);
    }
    else
    {
        label20.Visible = true;

```

```
label20.Text = "Поле Id має бути заповнене!";
```

```
}
```

```
}
```

```
}
```

```
}
```