

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ ЕКОНОМІКИ І ТЕХНОЛОГІЙ

ННІ/факультет	Інформаційних технологій
Кафедра	Інформатики і прикладного програмного забезпечення
Спеціальність	Інженерія програмного забезпечення
Форма навчання	Денна

**КВАЛІФІКАЦІЙНА
БАКАЛАВРСЬКА РОБОТА**

Москаленка Даніїла Вадимовича

(прізвище, ім'я, по батькові здобувача)

на тему Розробка програмного забезпечення для обліку операцій
купівлі-продажу валюти в пунктах обміну
на прикладі АТ "ПУМБ"

(повна назва теми)

за матеріалами праць провідних спеціалістів з розробки ПЗ та
проектування БД

(повна назва бази дослідження)

науковий керівник	<u>д.т.н., професор</u>	<u>Зеленський О.С.</u>
	<i>(наук. ступінь, вчене звання)</i>	<i>(підпис)</i>
		<i>(прізвище, ініціали)</i>

Робота допущена до захисту в ЕК

Протокол засідання кафедри

від 11.06.2025 № 12

Завідувач кафедри

(підпис)

д.т.н., професор

Наук. ступінь, вчене звання

Зеленський О.С.

Ініціали, прізвище

ЗГОДА здобувача вищої освіти

Державного університету економіки і технологій про перевірку кваліфікаційної роботи на прояви академічного плагіату та розміщення в Репозитарії Університету

Я, Москаленко Данііл Вадимович (ППП), підтримую політику Державного університету економіки і технологій з академічної доброчесності і відкритого доступу.

Засвідчую, що кваліфікаційна бакалаврська робота з теми “Розробка програмного забезпечення для обліку операцій купівлі-продажу валюти в пунктах обміну на прикладі АТ "ПУМБ"”

виконана самостійно та не містить академічного плагіату. Я не надавав і не одержував недозволену допомогу під час підготовки цієї роботи. Робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

Із чинним Положенням про запобігання та виявлення академічного плагіату в роботах здобувачів вищої освіти Державного університету економіки і технологій ознайомлений. Чітко усвідомлюю, що в разі виявлення у кваліфікаційній роботі порушення норм академічної доброчесності робота не допускається до захисту або оцінюється незадовільно.

Також я поінформований, що відповідно до «Положення про Репозитарій (електронну базу даних) Державного університету економіки і технологій» зазначена робота буде розміщена в Електронному архіві Університету (Репозитарії ДУЕТ). З умовами такого розміщення ознайомлений.

Дата

підпис

ініціали, прізвище (власноруч)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ ЕКОНОМІКИ І ТЕХНОЛОГІЙ

ННІ/факультет	Інформаційних технологій
Кафедра	Інформатики і прикладного програмного забезпечення
Спеціальність	Інженерія програмного забезпечення
Форма навчання	Денна

«ЗАТВЕРДЖУЮ»

Завідувач кафедри _____ Зеленський О.С.
(підпис) (Прізвище, ініціали)

«11» червня 2025 року

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ БАКАЛАВРСЬКУ РОБОТУ**

1. Тема роботи: «Розробка програмного забезпечення для обліку операцій купівлі-продажу валюти в пунктах обміну на прикладі АТ "ПУМБ"»

Керівник роботи д.т.н., професор Зеленський О.С.
затвердені наказом закладу вищої освіти від «04» квітня 2025 р. № 222-ст

2. Строк подання здобувачем роботи до «09» червня 2025 р.

3. Зміст кваліфікаційної роботи, об'єкт, предмет та мета дослідження:

Розділ 1. Дослідження предметної області

Розділ 2. Розробка алгоритму розв'язання задачі

Розділ 3. Організація інформаційного забезпечення

Розділ 4. Розробка програмного забезпечення задачі

Об'єкт дослідження: Діяльність пунктів обміну валют у банківських установах

Предмет дослідження: Автоматизації обліку валютних операцій у банках

Мета кваліфікаційної роботи: Розробка ПЗ для обліку операцій обміну іноземних валют в пунктах обміну, з урахуванням специфіки діяльності АТ «ПУМБ»

5. Дата видачі завдання «04» квітня 2025 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів МДР	Строк виконання етапів роботи	Відмітка керівника про виконання етапів (дата, підпис)
1	Підготовка розділу 1	04.04.2025-13.04.2025	
2	Підготовка розділу 2	14.04.2025-26.04.2025	
3.	Підготовка розділу 3	27.04.2025-15.05.2025	
4	Підготовка розділу 4	16.05.2025-08.06.2025	
5	Реєстрація завершеної кваліфікаційної роботи	09.06.2025	Реєстраційний №____ «09»червня 2025 р.
6	Отримання відгуку від наукового керівника	10.06.2025	
7	Подання кваліфікаційної роботи на перегляд завідувачу кафедри	11.06.2025	
8	Отримання зовнішньої рецензії	12.06.2025	
9	Попередній захист кваліфікаційної роботи на кафедрі	13.06.2025	
10	Підготовка до захисту в ЕК	16.06.2025-21.06.2025	

Завдання підготував науковий керівник

(підпис)

О.С. Зеленський

(прізвище та ініціали)

Завдання одержав

(підпис)

Д.В. Москаленко

(прізвище та ініціали)

АНОТАЦІЯ

на кваліфікаційну бакалаврську роботу

**«Розробка програмного забезпечення для обліку операцій купівлі-продажу
валюти в пунктах обміну на прикладі АТ "ПУМБ"»**

Москаленка Данііла Вадимовича

Кваліфікаційна бакалаврська робота на здобуття освітньо-кваліфікаційного рівня бакалавра зі спеціальності F2 «Інженерія програмного забезпечення» – Державний університет економіки і технологій – Кривий Ріг, 2025.

У цій дипломній роботі було створено програмне забезпечення для обліку операцій купівлі-продажу валюти в пунктах обміну валют на прикладі АТ «ПУМБ». Для реалізації цього завдання було розроблено базу даних у середовищах MySQL та Microsoft Access, яка стала важливою складовою модуля програмного комплексу «Операційний день банку».

Розроблена база даних забезпечує можливість експорту та імпорту операцій купівлі-продажу валюти до системи «Операційний день банку», а також реалізацію самих операцій обміну. Було реалізовано двосторонній зв'язок між базою даних пункту обміну валют і програмним комплексом банку, що сприяло підвищенню ефективності роботи персоналу.

За допомогою створеної бази даних співробітники пункту обміну отримали змогу формувати точні звіти щодо обсягів купленої та проданої валюти, а також додаткові журнали з фіксацією часу проведення операцій. Це дозволило зменшити навантаження на програмний комплекс «Операційний день банку», оскільки облік вівся не за кожною окремою операцією, а за зведеними сумами по кожній валюті.

Програмне забезпечення було реалізоване мовою програмування C#, що забезпечило належну функціональність, інтеграцію з базами даних та відповідність вимогам автоматизації валютно-обмінних процесів.

Ключові слова: ПУМБ, КУПІВЛЯ, ПРОДАЖ, ВАЛЮТА, АЛГОРИТМИ, СУБД, ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

IDE	<p>Integrated Development Environment — це інтегроване середовище розробки програмного забезпечення, яке поєднує в собі набір інструментів для написання, редагування, компіляції та налагодження програмного коду. IDE забезпечує зручний інтерфейс для роботи програміста та включає редактор коду, систему підсвічування синтаксису, автодоповнення, вбудовану консоль, інструменти для тестування та управління проєктами. Прикладом популярного IDE є Visual Studio, яка була використана у даній дипломній роботі для розробки програмного забезпечення.</p>
СУБД	<p>Система управління базами даних — це програмне забезпечення, яке забезпечує створення, зберігання, редагування та обробку даних у структурованому вигляді. СУБД керує взаємодією між прикладними програмами та самою базою даних, дозволяючи користувачам ефективно здійснювати запити, фільтрацію, оновлення та збереження інформації. У роботі були використані такі СУБД як MySQL і Microsoft Access, які забезпечили збереження даних про валютні операції, касирів, клієнтів тощо.</p>
АТ «ПУМБ»	<p>Акціонерне товариство Перший Український Міжнародний Банк — це одна з провідних банківських установ України, що надає повний спектр фінансових послуг фізичним та юридичним особам. Банк має широку мережу відділень і спеціалізується, зокрема, на обслуговуванні клієнтів у пунктах обміну валют.</p>
SQL	<p>Structured Query Language — це мова структурованих запитів, яка використовується для взаємодії з базами даних.</p>

	<p>SQL дозволяє виконувати такі операції, як вибірка даних (SELECT), оновлення (UPDATE), вставка (INSERT) та видалення (DELETE). У дипломній роботі SQL активно застосовувався для обробки запитів до таблиць, що містять інформацію про валютні курси, операції купівлі-продажу, клієнтів і касирів у базі даних MySQL.</p>
ООП	<p>Об'єктно-орієнтоване програмування — це парадигма програмування, заснована на використанні об'єктів, які є абстракцією реальних сутностей, що мають властивості (поля) та поведінку (методи). Основними принципами ООП є інкапсуляція, наслідування, поліморфізм та абстракція. У дипломному проєкті ООП використовувалося під час створення структури додатку мовою С#, де класи відображали сутності, такі як Клієнт, Валюта, Операція тощо.</p>
IntelliSense	<p>Інтелектуальна система автодоповнення коду, яка є частиною сучасних IDE, зокрема Visual Studio. Вона дозволяє розробникам писати код швидше та з меншими помилками, автоматично пропонуючи можливі методи, змінні, параметри функцій та інші конструкції мови під час написання коду. У рамках даної дипломної роботи IntelliSense суттєво спростив процес розробки програмного забезпечення, допомагаючи ефективно працювати з великим обсягом класів і структур даних.</p>

ЗМІСТ

ВСТУП.....	8
РОЗДІЛ I. ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ.....	10
1.1. Характеристика задачі.....	10
1.2. Вивчення й аналіз оперативних первинних документів.....	11
1.3. Вивчення нормативно-довідкових документів.....	16
1.4. Вивчення процесів перетворення вхідних повідомлень у вихідні.....	18
РОЗДІЛ II. РОЗРОБКА АЛГОРИТМУ РОЗВ'ЯЗАННЯ ЗАДАЧІ.....	23
РОЗДІЛ III. ОРГАНІЗАЦІЯ ІНФОРМАЦІЙНОГО ЗАБЕЗПЕЧЕННЯ.....	28
3.1. Створення та опис інфологічної моделі бази даних.....	28
3.2. Побудова та опис зв'язків логічної моделі бази даних.....	32
3.3. Побудова та опис фізичної моделі бази даних.....	33
3.4. Розробка бази даних у СУБД MS Access.....	35
3.5. Розробка бази даних у СУБД MySQL.....	39
РОЗДІЛ IV. РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ЗАДАЧІ.....	43
4.1. Обґрунтування вибору засобів розробки.....	43
4.2. Реалізація програмного забезпечення на мові C#.....	52
ВИСНОВКИ.....	62
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	64
ДОДАТКИ.....	65

ВСТУП

У сучасних умовах розвитку банківських інформаційних систем особливу увагу приділяють автоматизації операцій купівлі-продажу валюти, що здійснюються в пунктах обміну валют. Це зумовлено потребою в оперативному обліку валютно-обмінних операцій, підвищенні точності ведення звітності, зменшенні ризиків людського чинника та інтеграції з центральними банківськими системами.

У даній дипломній роботі розглянуто можливість вирішення проблеми автоматизації обліку операцій купівлі-продажу валюти на прикладі діяльності пунктів обміну АТ «ПУМБ». Основна увага зосереджена на створенні бази даних, яка є невіддільною складовою модуля для програмного комплексу «Операційний день банку». Такий підхід дає змогу здійснювати експорт та імпорт валютно-обмінних операцій, реалізовувати безпосередній процес купівлі-продажу валюти, а також забезпечити двосторонній зв'язок між пунктом обміну валют і центральною банківською системою.

Інтеграція системи обліку з програмним комплексом «Операційний день банку» дозволяє не лише прискорити обробку даних, а й оптимізувати процес підкріплення кас, що особливо важливо для готівкових розрахунків, у яких бере участь служба інкасації. Це, у свою чергу, сприяє пришвидшенню обігу національної та іноземної валюти між відділеннями банку.

Розроблена база даних також забезпечує можливість формування додаткових звітів і журналів операцій купівлі-продажу із зазначенням часу проведення, а також створення зведених звітів у розрізі кожної валюти. Такий підхід зменшує навантаження на програмний комплекс, дозволяючи обліковувати не кожну операцію окремо, а загальні суми в межах кожного виду валюти.

Об'єктом дослідження є процес автоматизації обліку валютно-обмінних операцій у фінансових установах, зокрема в банківських пунктах обміну валют.

Предметом дослідження є методи та засоби розробки програмного забезпечення для обліку операцій купівлі-продажу валюти, а також специфіка впровадження такого ПЗ на прикладі АТ «ПУМБ».

Метою дослідження є розробка ефективного програмного забезпечення для обліку операцій купівлі-продажу іноземної валюти в пунктах обміну валют, з урахуванням специфіки діяльності АТ «ПУМБ», з метою автоматизації, підвищення точності обліку та оптимізації щоденних валютно-обмінних операцій.

Отже, дана робота є актуальною в контексті вдосконалення внутрішніх бізнес-процесів банківської установи, підвищення ефективності роботи персоналу та оптимізації документообігу.

РОЗДІЛ I

ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1. Характеристика задачі

Визначення функціональних задач предметної області у контексті розробки бази даних для автоматизованого вирішення функціональних задач «облік купівлі-продажу валюти» потребує ретельного аналізу процесів купівлі-продажу та банківської діяльності. Основна мета даної роботи полягає в автоматизації процесів купівлі-продажу валюти, контролю залишків валюти в касах ПОВ, оптимізації документообігу банку та ПОВ.

Для успішної автоматизації обліку купівлі-продажу необхідно визначити ряд основних функцій, які повинні бути включені в систему:

- Облік купленої валюти: функція, що дозволяє фіксувати надходження грошових коштів до каси ПОВ, відстежувати кількість кожного типу валюти.
- Облік проданої валюти: функція, яка потрібна для того, щоб автоматизувати процес реєстрації вказаних з каси ПОВ отриманих коштів.
- Моніторинг залишків валюти в ПОВ: функція спрямована на постійне оновлення інформації про наявність або відсутність валюти в кожному ПОВ та переміщенням її між Банком та ПОВ.
- Генерація звітів: важлива функція для забезпечення аналізу ефективності діяльності ПОВ. Також функція важлива для генерації звітів для різних потреб.
- Управління даними про валюту та ПОВ: ця функція передбачає створення та підтримання бази даних, що містить інформацію про валюту, вимагаючи її найменування та інші дані. Забезпечення точного та актуального обліку дозволить ефективно здійснювати управління ПОВ.
- Управління даними про касирів та клієнтів: для продуктивної роботи ПОВ необхідно мати можливість зберігати та обробляти інформацію про касирів і клієнтів. Ця функція вимагає створення бази даних, що містить інформація як про

касирів так і про клієнтів. Управління цими даними дозволяє прискорити процес купівлі-продажу та зробити його більш зручним.

Прописування основних функцій для бази даних та їхня важливість для бізнесу обумовлені кількома ключовими аспектами. По-перше, це сприяє оптимізації процесів купівлі-продажу, оскільки чітко визначені функції дозволяють автоматизувати ключові операції, такі як облік валюти, відстеження купівлі-продажу, а також управління базою ПОВ. В свою чергу це підвищує ефективність роботи організації. Точність та достовірність даних також є важливим аспектом, що забезпечуються певними функціями бази даних.

Таким чином, використання сучасних баз даних із певними функціями допомагає бути на крок попереду конкурентів, прискорюючи прийняття рішень та реагування на зміни.

1.2. Вивчення й аналіз оперативних первинних документів

Рух національної та іноземної валюти між ТО та ПОВ КФ АТ «ПУМБ».

На початку робочого дня служба інкасації АТ «ПУМБ» доставляє готівку в іноземній та національній валюті в касу пункту обміну валют, таким чином здійснювати поновлення каси пункту обміну валют. Касир, отримуючи готівку, ставить свій особовий штамп і підпис. З цього починається робочий день касира. В програмі ПОВ касир вносить суми валют, які він отримав і курси валют, які вставив Криворізький філіал АТ «ПУМБ» на основі курсу валют НБУ (табл. 1.2). Наприклад, початок дня (табл. 1.1).

Таблиця 1.1

Початок дня

Код валюти цифровий	Назва валюти літерний	Сума
840	USD	500

978	EUR	500
980	UAH	10 000

Таблиця 1.2

Курс валют

Код валюти цифровий	Назва валюти літерний	Курс купівлі	Курс продажу
840	USD	40,46	43,45
978	EUR	41,05	42,25

Після цього відкривається робочий день пункту обміну валют АТ «ПУМБ». Впродовж робочого дня касир здійснює операції з купівлі-продажу валюти. Після кожної операції в програмі формується квитанція про операцію купівлі валюти (табл. 1.3).

Таблиця 1.3

Квитанція №1

Купівля валюти				
Сума в ін. валюті	Код ін. валюти	Курс	Код нац. валюти	Сума в нац. валюті
100 USD	40	40,46	980	4046,00 грн.
Підпис касира М.П.			Підпис клієнта	

Після операції продажу валюти друкується квитанція про операцію продажу валюти (табл. 1.4).

Квитанція №2

Продаж валюти				
Сума в нац. валюті	Код нац. валюти	Курс	Код ін. валюти	Сума в ін. валюті
4345,00 грн.	980	43,45	840	100 USD
Підпис касира М.П.			Підпис клієнта	

Протягом робочого дня в базі записуються дані про операції пункту обміну валют про купівлю валюти (табл. 1.5).

Таблиця 1.5

Дані операцій купівлі валют пункту обміну валют №1

№ операції	Час операції	Код ін. валюти	Сума в ін. валюті	Код нац. валюти	Сума в нац. валюті
1	08:25	840	100	980	4046,00
2	9:30	840	100	980	4046,00
3	10:50	978	200	980	8210,00
4	11:50	978	200	980	8210,00
5	14:50	840	100	980	4046,00
6	16:20	840	100	980	4046,00

В наступному журналі ми можемо побачити реєстр проданої валюти (табл. 1.6).

Таблиця 1.6

Дані операцій продажу валют пункту обміну валют №1

№ операції	Час операції	Код нац. валюти	Сума в нац. валюті	Код ін. валюти	Сума в ін. валюті
1	08:30	980	4345,00	840	100
2	9:50	980	4345,00	840	100
3	10:20	980	4225,00	978	100

4	14:30	980	4225,00	978	100
5	16:35	980	4345,00	840	100

В кінці робочого дня касир формує звіт для бухгалтерії валютного відділу банку. У касира є можливість сформулювати наступні звіти:

Журнал операцій купівлі іноземної валюти (табл. 1.7).

Таблиця 1.7

Журнал купівлі валюти

№	Сума	Код валюти	Час операції
1	100	840	08:25
2	100	840	9:30
3	200	978	10:50
4	200	978	11:50
5	100	840	14:50
6	100	840	16:20

Журнал операцій продажу іноземної валюти. (табл. 1.8)

Таблиця 1.8

Журнал продажу валюти

№	Сума	Код валюти	Час операції
1	100	840	08:30
2	100	840	9:50
3	100	978	10:20
4	100	978	14:30
5	100	840	16:35

Загальний звіт продажу валюти у кожній валюті (табл. 1.9).

Таблиця 1.9

Звіт проданої валюти USD

Сума	Курс валюти	Сума в національній валюті
------	-------------	----------------------------

300,00	43,45	14 035,00
--------	-------	-----------

Залишки у касі кожного типу валют (табл. 1.10).

Таблиця 1.10

Залишки на кінець робочого дня

Код валюти	Сума
840	400
978	500
980	15 875,00

Ярлик інкасаторської сумки (табл. 1.11).

Таблиця 1.11

Ярлик інкасаторської сумки

Код валюти	Сума
840	400,00
978	500,00
980	15 875,00

Звіт для служби інкасації банку (табл. 1.12).

Таблиця 1.12

Відомість

Код валюти	Сума
840	400,00
978	500,00
980	15 875,00
Підпис касира М. П.	Підпис інкасатора М. П.

Після закінчення робочого дня служба інкасації вивозить кошти з кожного пункту обміну валют, тому що вони на ніч не залишаються в сейфі касира.

1.3. Вивчення нормативно-довідкових документів

Нормативно-правова база купівлі-продажу валют.

- закон про банки і банківську діяльність;
- нормативна база НБУ купівлі продажу валют;
- про організацію Пунктів обміну валют в Україні;
- про організацію АТ «ПУМБ»;
- дозвіл на купівлю-продаж валюти.

Далі розглянуто Закон України «Про банки і банківську діяльність» [1].

Стаття 1. Цей Закон визначає структуру банківської системи, економічні, організаційні і правові засади створення, діяльності, реорганізації і ліквідації банків.

Метою цього Закону є правове забезпечення стабільного розвитку і діяльності банків в Україні і створення належного конкурентного середовища на фінансовому ринку, забезпечення захисту законних інтересів вкладників і клієнтів банків, створення сприятливих умов для розвитку економіки України та підтримки вітчизняного товаровиробника.

Стаття 6. Банки в Україні створюються у формі акціонерного товариства, товариства з обмеженою відповідальністю або кооперативного банку.

Законодавство про господарські товариства поширюється на банки у частині, що не суперечить цьому Закону.

Стаття 15. Найменування банку

Банк має повне і скорочене офіційні найменування українською та іноземними мовами. Найменування банку має містити слово "банк", а також вказівку на організаційно-правову форму банку.

Банк має печатку зі своїм повним офіційним найменуванням.

Слово "банк" та похідні від нього дозволяється використовувати у назві лише тим юридичним особам, які зареєстровані Національним банком України як банк і мають банківську ліцензію. Виняток становлять міжнародні організації,

що діють на території України відповідно до міжнародних договорів, згода на обов'язковість яких надана Верховною Радою України, та законодавства України.

Не дозволяється використовувати для найменування банку назву, яка повторює вже існуючу назву іншого банку або вводить в оману щодо видів діяльності, які здійснює банк. Вживання у найменуванні банку слів "Україна", "державний", "центральний", "національний" та похідних від них можливе лише за згодою Національного банку України.

Національний банк України має право відмовити у використанні банком запропонованого найменування за наведених у цій статті підстав.

Підрозділ банку використовує найменування лише того банку, підрозділом якого він є. До назви структурного підрозділу банку може додаватися найменування місцезнаходження цього підрозділу.

Згідно закону про банки і банківську діяльність, який був прийнятий Верховною Радою 21.12.2000 №2121-III. Перший Український Міжнародний Банк був створений як закрите акціонерне товариство 20 листопада 1991 року 23 грудня 1991 року банк був зареєстрований Національним банком України. У квітні 1992 року ПУМБ почав здійснювати банківські операції в тому числі і валютні.

«Перший Український Міжнародний банк» (ПУМБ) входить до групи найбільших банків України і є універсальним банком, діяльність якого зосереджена на комерційних, роздрібних і інвестиційно-банківських операціях. Видами діяльності Першого Українського Міжнародного Банк у передбачено виконання валютно-фінансових операцій.

Сертифікати для виконання фінансових-валютних операцій видаються в центральному офісі банку м. Запоріжжя. Часткові, для виконання фінансових-валютних операцій видаються відділенням міст України, в яких вони знаходяться, в тому числі і в Кривому Розі.

До фінансових установ України, які здійснюють фінансово-валютну роботу та зберігання резервів державних коштів є банківські установи, в тому

числі і Перший Український Міжнародний Банк, який засновано 20 листопада 1991 року.

Купівля-продаж валюти, яка нас цікавить в даній роботі регламентується Постановою № 82 від 05.03.2003 Про затвердження Положення про оформлення та виконання документів на перерахування, зарахування, купівлю та продаж іноземної валюти або банківських металів.

Також, окрім регламентуючих документів НБУ існують внутрішні документи АТ «ПУМБ» на основі яких здійснюються операції купівлі-продажу валюти.

1.4. Вивчення процесів перетворення вхідних повідомлень у вихідні

Вивчення процесів перетворення вхідних повідомлень у вихідні в рамках розробки бази даних для вирішення комплексу функціональних задач «Облік купівлі-продажу валюти в ПОВ» є важливим етапом проектування системи.

Вхідними повідомленнями є дані про ПОВ, валюту, касирів та клієнтів, а також загальна інформація про них. Перетворення цих даних у вхідні та обчислювальні дані здійснюється через складання облікових записів, що містять інформацію про рух грошових коштів в ПОВ та ТО Банка за певний період часу. Для цього в базі даних створюються відповідні таблиці про купівлю та продаж, які дозволять у вихідних повідомленнях слідкувати за залишками коштів в ПОВ на початок та кінець періоду, наприклад, день, тиждень, місяць. Це дозволяє точно вести облік грошових коштів. Також розробка журналу операцій забезпечить доступ та концентрований звіт для зручності обліку. Такий підхід до перетворення даних дозволяє ефективно автоматизувати процес обліку в ПОВ.

Кожна установа, яка займається купівлею-продажем валют, керується інструкціями НБУ. Для того щоб отримати прибуток в банках щодня розраховується валютна маржа для кожного з типів валют.

Маржа – це величина яка виражає різницю між процентними ставками, курсами цінних паперів, ставками страхових внесків і відшкодувань. В нашому випадку це різниця між курсами НБУ і курсом купівлі або продажу [2].

Маржа в торгівлі (продаж товару/послуг) – це різниця між відпускною ціною товару та витратами на його виробництво чи закупівлю, що показано у формулі (1.1).

$$M = BC - C \quad (1.1)$$

де M – маржа в торгівлі;

BC – відпускна ціна, що ;

C – собівартість, що .

Розрахунок валютної маржі залежить від контексту – це може бути торгівля на біржі, маржа у торгівлі або просто різниця між ціною купівлі та продажу.

Маржа у торгівлі валютами (Forex) – це необхідний депозит для відкриття та підтримки позиції у брокера, що показано у формулі (1.2) [3].

$$M = \frac{OU}{KP} \quad (1.2)$$

де M – маржа у торгівлі валютами;

OU – обсяг угоди, що є кількістю валюти, яку потрібно купити чи продати;

KP – кредитне плече, що вказує на співвідношення маржі до обсягу угоди.

Для розуміння розрахунку курсів купівлі-продажу було визначено основні валюти які є на фінансовому ринку України, їх зазначено в табл. 1.13 [4].

Для розрахунків будуть потрібні наступні дані:

– валюти;

- курс НБУ основних валют, які знаходяться в банку;
- курс продажу;
- курс купівлі;
- курс пунктів обміну валют.

Таблиця 1.13

Опис основних валют

№ з/п	Літерний код валюти	Цифровий код	Назва валюти	Країна
1	UAH	980	Гривня	Україна
2	USD	980	Долар США	США
3	EUR	978	Євро	Євросоюз
4	GBP	926	Фунт стерлінгів	Велика Британія

Середньозважений курс розраховується за формулою (1.3).

$$C_{cp} = \frac{\sum_{i=1}^n C_i \cdot b_i}{\sum_{i=1}^n b_i} \quad (1.3)$$

де C_{cp} – середньозважений курс;
 n – загальна кількість значень;
 i – номер ітерації;
 C_i – значення курсу з певної ітерації торгів;
 b_i – значення розміру певної ітерації торгів.

Для правильної роботи ПЗ ПОВ та для формування курсів продажу та купівлі валюти потрібно визначити спосіб розрахунку націнки, що наведено у формулі (1.4).

$$BH = BK - \frac{PK}{PK} \cdot 100\% \quad (1.4)$$

де ВН – відсоток націнки;
 БК – банківський курс валюти;
 РК – ринковий курс валюти.

Після розрахунку курсів валют уповноважена особа передає інформацію в кожен пункт обміну валют. Дані показані в табл. 1.14 будуть потрібні при розрахунках та обчисленнях [5].

Таблиця 1.14

Опис курсу продажу валют

№ з/п	Цифровий код	Літерний код валюти	Кількість одиниць валюти	Назва валюти	Курс НБУ	Курс Продажу
1	980	USD	1	Долар США	41,55	41,75
2	978	EUR	1	Євро	47,50	47,70
3	926	GBP	1	Фунт стерлінгів	56,12	56,50

Після розрахунку та встановлення курсів продажу валют той самий розрахунок робиться і для купівлі, що наочно продемонстровано в табл. 1.15.

Таблиця 1.15

Опис курсу купівлі валют

№ з/п	Цифровий код	Літерний код валюти	Кількість одиниць валюти	Назва валюти	Курс НБУ	Курс Купівлі
1	980	USD	1	Долар США	41,55	41,25
2	978	EUR	1	Євро	47,50	47,10
3	926	GBP	1	Фунт стерлінгів	56,12	55,50

Після цього, ми отримуємо необхідні дані для модуля, який розраховує суми купівлі-продажу, які необхідні для функціонування пункту обміну валют, що наведено в табл. 1.16.

Таблиця 1.16

Опис курсу купівлі-продажу валюти

№ з/п	Цифровий код	Літерний код валюти	Курс Купівлі	Курс Продажу
1	980	USD	41,25	41,75
2	978	EUR	47,10	47,70
3	926	GBP	55,50	56,50

Таким чином було підготовлено методи, що дозволять сформувати алгоритм роботи програми.

Висновки до розділу I

На основі проведеного дослідження предметної області можна зробити висновок, що автоматизація обліку операцій купівлі-продажу валюти в пунктах обміну банку є актуальним завданням, яке безпосередньо впливає на ефективність щоденної діяльності фінансової установи. Аналіз процесів, що відбуваються в пунктах обміну валют АТ «ПУМБ», показав необхідність створення програмного забезпечення, яке дозволяє не лише зберігати інформацію про транзакції, а й забезпечувати інтеграцію з центральним банківським програмним комплексом «Операційний день банку».

Особливу увагу було приділено розробці зручного інтерфейсу користувача, гнучкому механізму налаштувань, формуванню звітів та реалізації двосторонньої взаємодії з базою даних. Таким чином, проведене дослідження предметної області дало змогу визначити ключові функціональні вимоги до майбутньої інформаційної системи, що ляжуть в основу проектування та реалізації програмного продукту.

РОЗДІЛ II

РОЗРОБКА АЛГОРИТМУ РОЗВ'ЯЗАННЯ ЗАДАЧІ

Кожен з нас постійно зустрічається з безліччю завдань від найпростіших і добре відомих до дуже складних. Для багатьох задач існують певні правила (інструкції, розпорядження), що пояснюють виконавцю, як вирішувати дану задачу. Ці правила людина може вивчити заздалегідь або сформулювати самостійно у процесі виконання завдання. Такі правила прийнято називати алгоритмами.

Написанню програми завжди передує розробка деякого плану розв'язання задачі. Поняття алгоритму – одне з основних у програмуванні та інформатики. Алгоритм – це певна послідовність дій, написана на зрозумілій виконавцю алгоритмічній мові і визначаюча процес переходу від вихідних даних до результату.

При створенні алгоритму ми будемо дотримуватись таких особливостей:

- алгоритм завершиться після виконання скінченної кількості кроків;
- процес, що визначається алгоритмом, розділимо на окремі елементарні етапи, кожен з яких називається кроком алгоритмічного процесу;
- кожен крок алгоритму буде точно визначений. Дії, які необхідно здійснити, будуть чітко та недвозначно визначені для кожного можливого випадку;
- алгоритм має деяку кількість вхідних даних, тобто, величин, заданих до початку його роботи або значення яких визначають під час роботи алгоритму;
- алгоритм має одне або декілька вихідних даних, тобто, величин, що мають досить визначений зв'язок із вхідними даними.

У цьому визначенні вже зазначені основні властивості алгоритму. По – перше, алгоритм складається з кінцевого набору інструкцій або кроків, по-друге, кожен крок трактується виконавцем єдиним чином, що дозволяє гарантовано отримати рішення для деякого набору вхідних даних, по-третє, алгоритм завжди зводиться до деякого перетворення вихідних даних у результат чи результати. Для

машини, зрозуміло, потрібна більш чітка формалізація завдання, ніж для людини, розуміти природну мову комп'ютери поки нездатні, звідси необхідність врахування при складанні алгоритму обмеженого набору інструкцій ЕОМ.

Перед тим, як почати створювати програму, необхідно точно визначити мету її написання. Так, коли мета точно визначена, слід розглянути більш докладно методи досягнення поставленої перед програмою мети, ці методи досягнення бувають представлені в наборі певних дій, при цьому кожна дія запускається по черзі, тобто у визначеному заздалегідь заданому порядку.

Для полегшення процесу реалізації та з метою представлення алгоритму функціонування програм, було розроблено алгоритм для програми (Visual C#).

Програма буде мати головне меню, за допомогою якого будуть реалізовуватись алгоритми вирішення задач та буде здійснюватись управління програмними комплексами.

Процес автоматизації обліку купівлі-продажу валюти у пунктах обміну банку потребує чітко визначеної послідовності дій, яка б забезпечувала надійність, швидкість та зручність обробки операцій. Алгоритм, що реалізується в програмному забезпеченні, має охоплювати основні функціональні компоненти системи: ініціалізацію та налаштування додатку, взаємодію з базою даних, операції з валютою, виведення та друк таблиць, а також забезпечення доступу до актуальних курсів валют.

Завдяки такому алгоритму, який наведено на рис. 2.1, забезпечується повний цикл обробки валютно-обмінних операцій у режимі реального часу, з урахуванням вимог безпеки, точності, інтеграції з основним банківським комплексом, а також зручності для користувача. Реалізація цього алгоритму дозволяє значно покращити ефективність роботи пунктів обміну валют АТ «ПУМБ» і є вагомим внеском у цифрову трансформацію банківських сервісів.

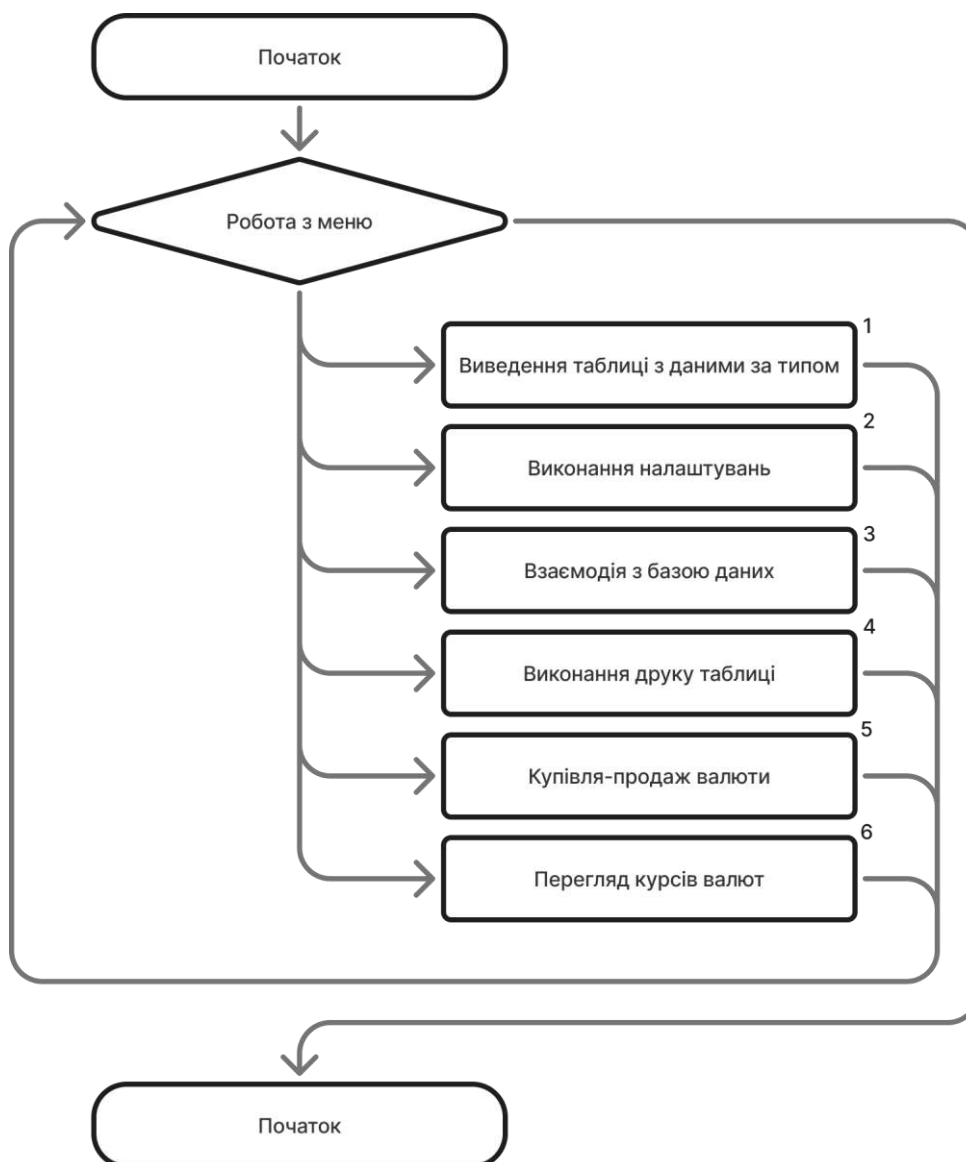


Рис. 2.1. Загальний алгоритм роботи програми

Далі буде розглянуто основні кроки роботи, які необхідно виконати для купівлі-продажу валюти.

Запуск та ініціалізація програми. Після запуску програма переходить у режим ініціалізації, де здійснюється первинне завантаження конфігураційних параметрів та виконуються налаштування інтерфейсу користувача. На цьому етапі перевіряється наявність доступу до бази даних, цілісність таблиць, зчитуються останні збережені налаштування користувача, що дозволяє адаптувати роботу системи до конкретного оператора пункту обміну.

Виконання налаштувань додатку (2). У цьому блоці реалізується функція конфігурації додатку. Користувач має змогу налаштувати права доступу, параметри друку, шаблони звітів, облікові параметри для кожної операції

(наприклад, граничні суми або валюти, що дозволені для обміну), встановити шлях з'єднання з центральною базою даних «Операційного дня банку». Цей етап є важливим з точки зору персоналізації системи та забезпечення її безпечної та ефективної експлуатації.

Взаємодія з базою даних (3). Після налаштування система активує постійне з'єднання з базою даних, що зберігає всю інформацію про операції купівлі-продажу, касові залишки, курси валют, змінні параметри, а також журнали транзакцій. Запити до бази здійснюються в режимі реального часу, що дозволяє актуалізувати дані миттєво після проведення кожної операції. Для оптимізації запитів використовуються SQL-команди з фільтрацією, індексами та транзакційною підтримкою.

Перегляд курсів валют (6). Оператор може звернутися до окремого модуля перегляду курсів валют. Дані можуть автоматично підвантажуватися із зовнішніх джерел (наприклад, внутрішньобанківських API) або задаватися вручну відповідно до рішення керівництва банку. Перегляд курсів реалізується у вигляді таблиці з можливістю сортування, фільтрації та оновлення. Цей етап важливий для попередньої оцінки умов обміну перед здійсненням операцій.

Купівля-продаж валюти (5). Це ключовий етап роботи програми. Оператор обирає тип операції (купівля або продаж), вказує валюту, суму, облікові дані клієнта (якщо потрібно), а також підтверджує операцію. Система автоматично розраховує еквівалент за поточним курсом, перевіряє наявність достатніх коштів у касі, проводить запис до бази даних, а також фіксує дату й час операції. Перед завершенням транзакції оператор має змогу переглянути короткий підсумок або відмінити дію.

Виведення таблиці з даними за типом (1). Після проведення кількох операцій користувач може сформулювати звіт у вигляді таблиці. Система надає змогу обирати тип даних — наприклад, операції лише з продажу, лише з купівлі, або всі транзакції за певний період. Таблиця формується автоматично із зазначенням дати, валюти, типу операції, суми, курсу та оператора. Цей звіт може

бути використаний як для внутрішнього контролю, так і для формування підсумкових звітів у рамках банківської документації.

Виконання друку таблиці (4). Для зручності документообігу реалізовано можливість друку сформованих таблиць. Користувач має змогу обрати формат виводу, кількість копій, додаткові параметри (наприклад, підпис касира або печатку). Друк виконується з попереднім переглядом, що забезпечує точність і відповідність банківським вимогам.

Таким чином було визначено алгоритм роботи користувача, для виконання операцій купівлі-продажу валют.

Висновки до розділу II

У результаті розробки алгоритму розв'язання задачі було сформовано чітку послідовність дій, що забезпечує ефективне функціонування програмного забезпечення для обліку валютно-обмінних операцій у пунктах обміну валют АТ «ПУМБ». Алгоритм охоплює всі основні етапи: ініціалізацію програми, налаштування параметрів, встановлення з'єднання з базою даних, перегляд курсів валют, виконання операцій купівлі-продажу, формування таблиць за типом операцій і здійснення друку. Кожен етап має чітко визначену логіку та послідовність виконання, що дозволяє зменшити кількість помилок під час обробки даних і скоротити час виконання стандартних операцій.

Важливою особливістю алгоритму є інтеграція з базою даних і підтримка обміну інформацією з комплексом «Операційний день банку», що забезпечує централізований контроль і облік. Алгоритм також передбачає можливість адаптації під різні сценарії використання, що дозволяє масштабувати рішення.

Таким чином, розроблений алгоритм став основою для подальшої реалізації функціонального, надійного та зручного інструменту автоматизації обліку в обмінних пунктах банку.

РОЗДІЛ III

ОРГАНІЗАЦІЯ ІНФОРМАЦІЙНОГО ЗАБЕЗПЕЧЕННЯ

3.1. Створення та опис інфологічної моделі бази даних

Інфологічна модель бази даних відіграє у процесі розробки інформаційних систем. Вона допомагає зрозуміти бізнес-логіку та основні об'єкти, з якими взаємодіятиме система.

Інфологічна модель — це концептуальне подання інформаційних потреб предметної області, що описує, які саме дані необхідно зберігати, їхні властивості, взаємозв'язки та правила обробки. Вона не залежить від конкретної СУБД і є основою для побудови логічної та фізичної моделей бази даних [6].

Створення інфологічної моделі дозволяє ідентифікувати сутності та зв'язки між ними, що важливо для правильного відображення вимог замовника у структурі бази даних. Більше того, така модель сприяє гнучкості та розширюваності бази даних, забезпечуючи можливість легкого додавання нових елементів без серйозних змін у її структурі. Уніфікація даних та стандартизація їх подання дозволяють підвищити якість інформації та забезпечити узгодженість даних у різних частинах системи. Попереднє створення інфологічної моделі також дозволяє оптимізувати запити та продуктивність бази даних, що є важливим для ефективного функціонування системи в цілому.

Таким чином, інфологічна модель бази даних не тільки є основою для проектування конкретної бази даних, а й є ключовим інструментом для розуміння бізнес-процесів, уніфікації даних та створення гнучких та ефективних інформаційних систем. Нижче наведена візуалізація даної моделі для кожного об'єкту у відповідних таблицях.

Першою сутністю буде розглянуто «Пункт обміну валют» (табл. 3.1), що відповідає за збереження інформації про місця, де виконується обмін валют.

Таблиця 3.1

Опис сутності «Пункт обміну валют»

Назва атрибута	Формат	Відсоток наявності	Обмеження на право звертання до зн. атрибута	Виводимість значень	Дублювання значень
Код пункту	9(8)	100%	Усі	-	Ні
Назва пункту	X(50)	100%	-	-	Ні
Адреса пункту	X(50)	100%	-	-	Ні
Код касира	9(8)	100%	-	-	Ні
Код клієнта	9(8)	100%	-	-	Ні

Сутність "Валюта" відповідає за інформацію про конкретну валюту, та використовується в операції купівлі-продажу (табл. 3.2).

Таблиця 3.2

Опис сутності «Валюта»

Назва атрибута	Формат	Відсоток наявності	Обмеження на право звертання до зн. атрибута	Виводимість значень	Дублювання значень
Код валюти	9(3)	100%	Усі	-	Ні
Назва валюти	X(50)	100%		-	Ні
Країна	X(50)	100%		-	Ні

Сутність "Клієнт" відповідає за інформацію про клієнтів, які використовують систему купівлі-продажу валюти (табл. 3.3).

Опис сутності «Клієнт»

Назва атрибута	Формат	Відсоток наявності	Обмеження на право звертання до зн.атрибута	Виводимість значень	Дублювання значень
Код клієнта	9(8)	100%	Усі	-	Ні
ПІБ клієнта	X(50)	100%	-	-	Так
Телефон клієнта	9(14)	100%	-	-	Ні

Сутність "Купівля" відповідає за надходження валюти в касу пункту обміну валют (табл. 3.4).

Таблиця 3.4

Опис сутності «Купівля»

Назва атрибута	Формат	Відсоток наявності	Обмеження на право звертання до зн.атрибута	Виводимість значень	Дублювання значень
Код купівлі	9(8)	100%	Касир та адміністратор	-	Ні
Код валюти	9(3)	100%		-	Так
Курс валюти	9(9).9(2)	100%		-	Так
Сума валюти	9(9).9(2)	100%		-	Так
Дата купівлі	X(8)	100%		-	Ні
Код пункту	9(8)	100%		-	Так

Сутність "Продаж" відповідає за інформацію про реєстрацію видачі валюти з каси пункту обміну валют (табл. 3.5).

Опис сутності «Продаж»

Назва атрибута	Формат	Відсоток наявності	Обмеження на право звертання до зн. атрибута	Виводимість значень	Дублювання значень
Код продажу	9(8)	100%	Касир та адміністратор	-	Ні
Код валюти	9(3)	100%		-	Так
Курс валюти	9(9).9(2)	100%		-	Так
Сума валюти	9(9).9(2)	100%		-	Так
Дата продажу	X(8)	100%		-	Ні
Код пункту	9(8)	100%		-	Так

Остання сутність це «Касир», що включає в себе дані про працівників пункту обміну валют, які працюють в банку (табл. 3.6).

Таблиця 3.6

Опис сутності «Касир»

Назва атрибута	Формат	Відсоток наявності	Обмеження на право звертання до зн. атрибута	Виводимість значень	Дублювання значень
Код касира	9(8)	100%	Усі	-	Ні
ПІБ касира	X(50)	100%	-	-	Так
Телефон касира	9(14)	100%	-	-	Ні

Таким чином було виконано формування інфологічної моделі бази даних, на основі якої можна створити фізичну модель.

3.2. Побудова та опис зв'язків логічної моделі бази даних

Створення логічної моделі для бази даних дозволяє організувати дані в структуровану форму, що полегшує їхнє управління, обробку та аналіз. Така модель допомагає розробникам зрозуміти структуру даних, визначити зв'язки між ними та спланувати ефективні методи доступу до інформації. Відповідно, завдяки логічній моделі можна проектувати базу даних таким чином, щоб вона відповідала бізнес-вимогам і була гнучкою для майбутніх змін.

Тому була створена логічна модель бази даних для обліку купівлі-продажу валюти на ERwin (рис. 3.1). У ній ми можемо побачити 6 сутностей та їх атрибути.

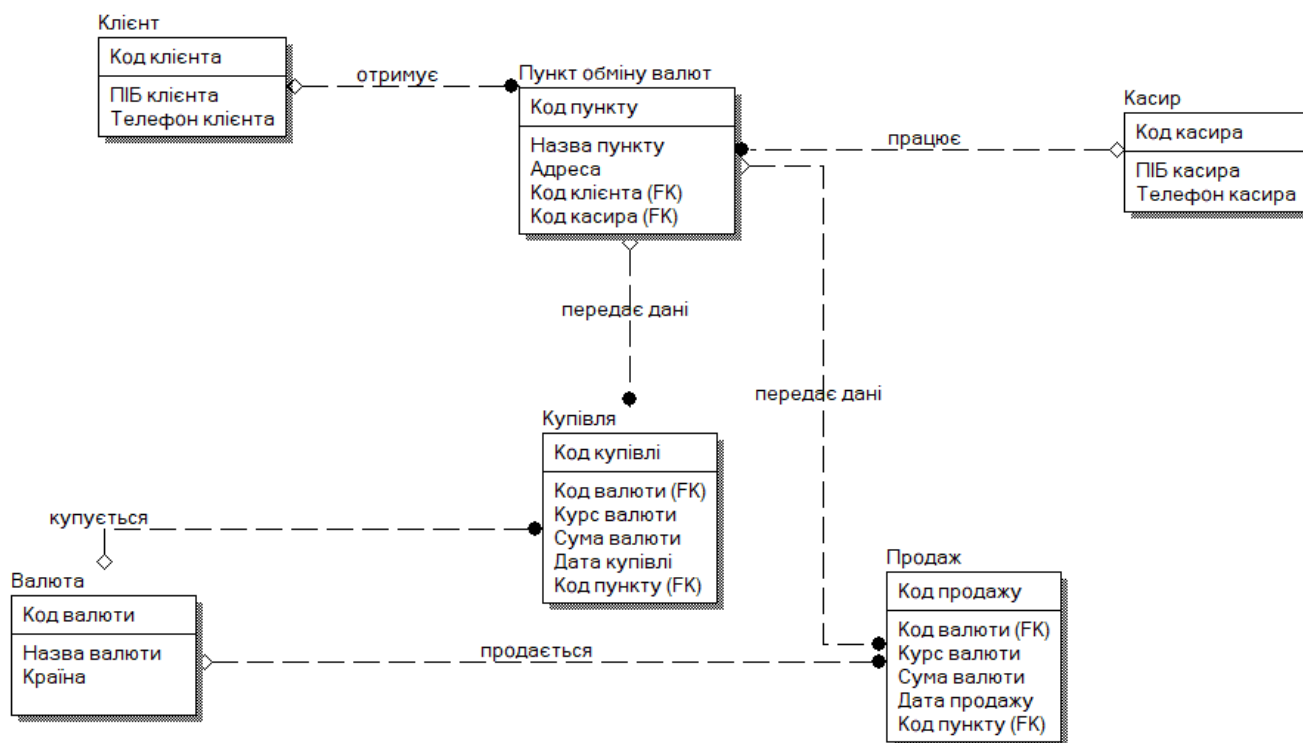


Рис. 3.1. Логічна модель даних для обліку продажу валюти

В цій логічній моделі сутностями є Клієнт, Касир, Пункт обміну валют, Купівля, Продаж і Валюта.

Категорія пункту обміну валют слугує для систематизації та передачі своїх даних сутностям Клієнт та Касир, також має зв'язок із сутністю Пункт обміну валют. Зв'язок між пунктом обміну валют та валютою має кардинальність «Zero, One or More».

Сутності Купівля і Продаж мають зв'язок із купівлею та продажем. Це дозволяє відстежити кількість операцій купівлі і продажу, дату купівлі-продажу, час операції.

Сутність Пункт обміну валют є центральною у базі даних. Вона пов'язана з сутностями Купівля і Продаж. Зв'язки мають кардинальність «One or More (P)». Nulls не допущено, щоб уникнути наявності записів про купівлю/продаж без валюти.

Клієнт вносить валюту в касу. Він пов'язаний із сутністю Пункт обміну валют і має кардинальність «Zero, One or More», що означає, що кожна операція Клієнта з одним Пунктом обміну валют. А в Пункті обміну валют можуть бути кілька операцій з конкретним клієнтом або може, взагалі, не бути операцій.

Продаж валюти є процесом продажу Касиром валюти Клієнту. Вона пов'язана із сутністю Продаж. Зв'язок клієнта з Пунктом обміну валют має кардинальність «Zero, One or More», тому що кожний процес купівлі-продажу валюти обов'язково пов'язаний із Касиром.

Атрибути, є характеристики чи властивості сутностей, які описані дані у базі даних. Вони є особливими елементами, за допомогою яких можна визначити та описати інформацію про об'єкт або сутність у базі даних. Атрибути визначають структуру даних, забезпечуючи її організацію, визначення та доступу до інформації. Атрибути всіх сутностей було розглянуто в попередньому розділі.

Таким чином було визначено зв'язки між сутностями бази даних, що дозволяє побудувати фізичну модель.

3.3. Побудова та опис фізичної моделі бази даних

Створення та опис фізичної моделі для бази даних - це ключовий етап у проектуванні бази даних, який має кілька важливих переваг.

По-перше, фізична модель дає точне уявлення про те, як дані зберігатимуться у базі даних. Це дозволяє розробникам та адміністраторам баз даних ефективно оптимізувати структуру бази даних для забезпечення

оптимальної продуктивності та ефективного використання ресурсів зберігання. Крім того, фізична модель є основою для створення схеми бази даних, яка визначає структуру таблиць, їх поля та зв'язки між ними. Це забезпечує чітке розуміння структури даних та їх взаємозв'язків, що спрощує розробку додатків та запитів до бази даних. Також фізична модель служить важливим інструментом для документування бази даних, що дозволяє легко передавати знання про структуру даних іншим членам команди розробки та забезпечує цілісність бази даних протягом її життєвого циклу. Зрештою, створення фізичної моделі допомагає виявити потенційні проблеми продуктивності чи цілісності даних на ранніх етапах проектування, що дозволяє уникнути їх у майбутньому та забезпечити стабільну роботу бази даних.

Фізична модель бази даних для обліку купівлі-продажу валюти на ERwin (рис. 3.2).

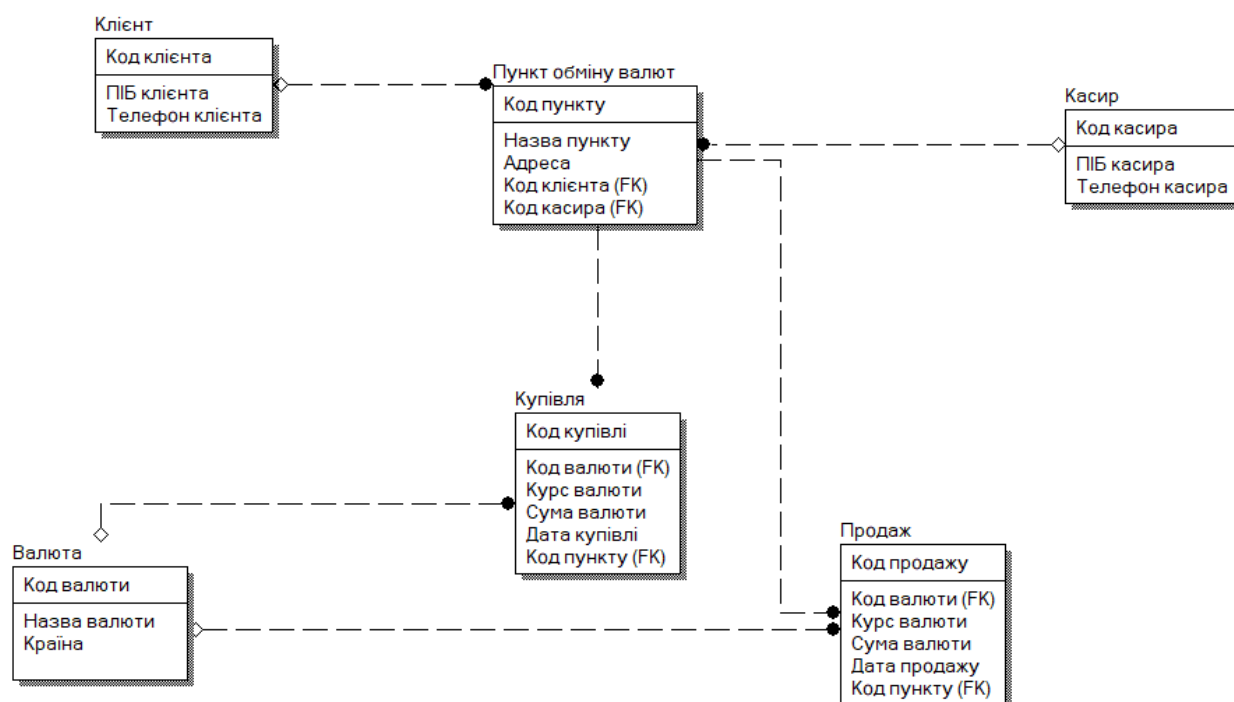


Рис. 3.2. Фізична модель даних для обліку продажу валюти

Розглянемо обмеження всіх зв'язків, щоб гарантувати цілісність даних:

Для зв'язків між усіма іншими таблицями, будь-яка спроба вставити або оновити дочірній елемент буде обмежена (RESTRICT). Це також стосується

батьківських операцій – спроба вставити або оновити данні буде обмежена. Жодна з операцій не буде автоматично здійснена. Це стосується таких зв'язків: «Клієнт» і «Пункт обміну валют», «Касир» і «Пункт обміну валют», «Пункт обміну валют» і «Купівля», «Пункт обміну валют» і «Продаж», «Валюта» і «Продаж», «Валюта» і «Купівля» (табл. 3.7).

Таблиця 3.7

Визначення обмежень посилальної цілісності

Дія	Дочірній елемент	Батьківський елемент
Видалення	NONE	RESTRICT
Вставка	RESTRICT	NONE
Оновлення	RESTRICT	RESTRICT

Таким чином було створено фізичну модель бази даних, що дозволяє сформулювати запити та команди на реалізацію цієї моделі в системі керування базами даних MS Access та MySQL.

3.4. Розробка бази даних у СУБД MS Access

Реалізація бази даних відіграє ключову роль у сучасних інформаційних технологіях, забезпечуючи ефективне зберігання, організацію та доступ до даних. Вона необхідна для структурованого зберігання інформації та забезпечення її доступності, обробки, аналізу та використання для прийняття рішень. Бази даних дозволяють зберігати різноманітні типи даних, включаючи текст, числа, зображення та інші, що робить їх універсальним інструментом роботи з інформацією.

У контексті використання Microsoft Access, реалізація бази даних набуває ряду переваг. Перш за все, Access має звичайно простий та інтуїтивно зрозумілий інтерфейс, що робить процес створення та управління базами даних доступним навіть для користувачів без спеціалізованих знань в області баз даних. Завдяки готовим шаблонам та інструментам для створення таблиць, запитів, форм та

звітів, Access дозволяє швидко створювати прототипи баз даних та проводити експерименти з їхньою структурою та змістом.

Ще однією перевагою Access є його інтеграція з іншими програмами Microsoft Office, такими як MS Excel та MS Word. Це забезпечує зручний доступ до даних із різних додатків, а також можливості для аналізу та звітності.

Початковим етапом при впровадженні бази даних є створення самої бази даних та її таблиць, що виконується автоматично через генерацію схеми ERwin. Внаслідок цього процесу створюється база даних для обліку купівлі продажу валют в СУБД MS Access, з таблицями та його зв'язкам (рис. 3.3).

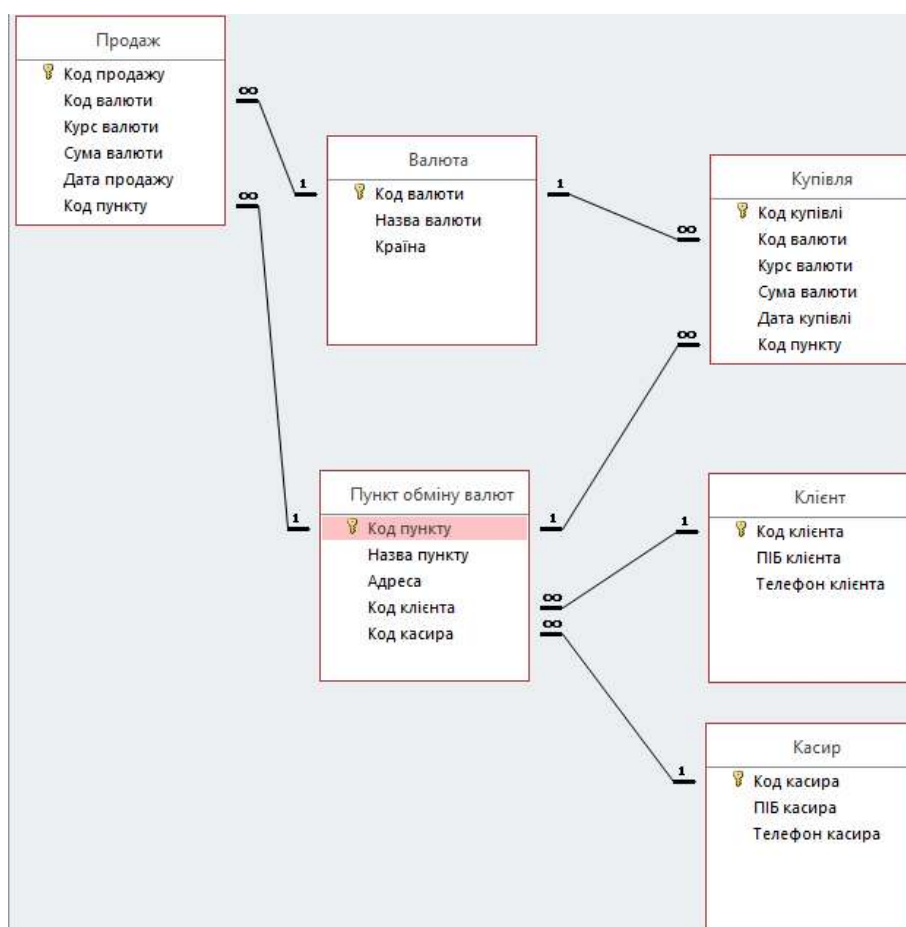


Рис 3.3. Схема бази даних в СУБД MS Access.

Як ми можемо побачити, схема була успішно згенерована з ERwin до СУБД MS Access. Всі таблиці, їх взаємозв'язки та стовпці ідентичні фізичній моделі, зображеній на рис. 3.2 у підрозділі 3.3 "Побудова та опис фізичної моделі бази даних" (додаток Б). Властивості полів представлені у вигляді таблиці опису (табл. 3.8).

Опис таблиць бази даних у СУБД MS Access

№ п/п	Назва таблиці	Назва стовпця	Тип даних	Властивості
1	Касир	Код касира	Лічильник	Primary key
2	Касир	ПІБ касира	Короткий текст	Розмір поля: 50; не може бути нулем
3	Касир	Телефон касира	Короткий текст	Розмір поля: 14; не може бути нулем
4	Валюта	Код валюти	Лічильник	Primary key
5	Валюта	Назва валюти	Короткий текст	Розмір поля: 50; не може бути нулем
6	Валюта	Країна	Короткий текст	Розмір поля: 50; не може бути нулем
7	Клієнт	Код клієнта	Лічильник	Primary key
8	Клієнт	ПІБ клієнта	Короткий текст	Розмір поля: 50; не може бути нулем
9	Клієнт	Телефон клієнта	Короткий текст	Розмір поля: 14; не може бути нулем
10	Пункт обміну валют	Код пункту	Лічильник	Primary key
11	Пункт обміну валют	Назва пункту	Короткий текст	Розмір поля: 50; не може бути нулем
12	Пункт обміну валют	Адреса пункту	Короткий текст	Розмір поля: 50; не може бути нулем
13	Пункт обміну валют	Код клієнта	Числовий	Foreign key; не може бути нулем
14	Пункт обміну валют	Код касира	Числовий	Foreign key; не може бути нулем
15	Купівля	Код купівлі	Лічильник	Primary key

16	Купівля	Код валюти	Числовий	Foreign key; не може бути нулем
17	Купівля	Курс валюти	Числовий	З плаваючою точкою Не може бути нулем
18	Купівля	Сума валюти	Числовий	З плаваючою точкою Не може бути нулем
19	Купівля	Дата купівлі	Дата і час	Не може бути нулем
20	Купівля	Код пункту	Числовий	Foreign key; не може бути нулем
21	Продаж	Код продажу	Лічильник	Primary key
22	Продаж	Код валюти	Числовий	Foreign key; не може бути нулем
23	Продаж	Курс валюти	Числовий	З плаваючою точкою Не може бути нулем
24	Продаж	Сума валюти	Числовий	З плаваючою точкою Не може бути нулем
25	Продаж	Дата продажу	Дата і час	Не може бути нулем
26	Продаж	Код пункту	Числовий	Foreign key; не може бути нулем

3.5. Розробка бази даних у СУБД MySQL

Після попередніх етапів реалізації бази даних для обліку купівлі-продажу валюти в Microsoft Access і MS SQL, здійснення розробки в СУБД MySQL є ще одним важливим сходинкою в оптимізації та вдосконалення. Перехід до MySQL обумовлений прагненням до подальшого покращення продуктивності, масштабованості та гнучкості бази даних, а також потенційним зниженням витрат на обслуговування та ліцензування.

Однією з ключових переваг MySQL є його відкритість та вільний доступ. Це означає, що організації можуть уникнути значних витрат на ліцензування, що є особливо актуальним для підприємств, які прагнуть скоротити витрати без шкоди функціональності та надійності системи. Крім того, доступність MySQL забезпечує доступ до широкого співтовариства розробників і ресурсів, сприяючи обміну знань і досвіду, а також швидкому вирішенню проблем, що виникають [7].

Ще однією перевагою MySQL є його висока продуктивність та масштабованість. СУБД володіє ефективною архітектурою, здатною обробляти великі обсяги даних з високою швидкістю, що робить його привабливим вибором для обліку грошових коштів в пунктах обміну валют. Більш того, MySQL має механізми реплікації та кластеризації, що дозволяють горизонтально масштабувати систему у разі необхідності збільшення її продуктивності або розширення функціональності.

Тому ми здійснили впровадження бази даних у MySQL (табл. 3.9). На рисунку 3.4 можна побачити організацію таблиць та їх взаємозв'язки, які були перенесені з ERwin (додаток А).

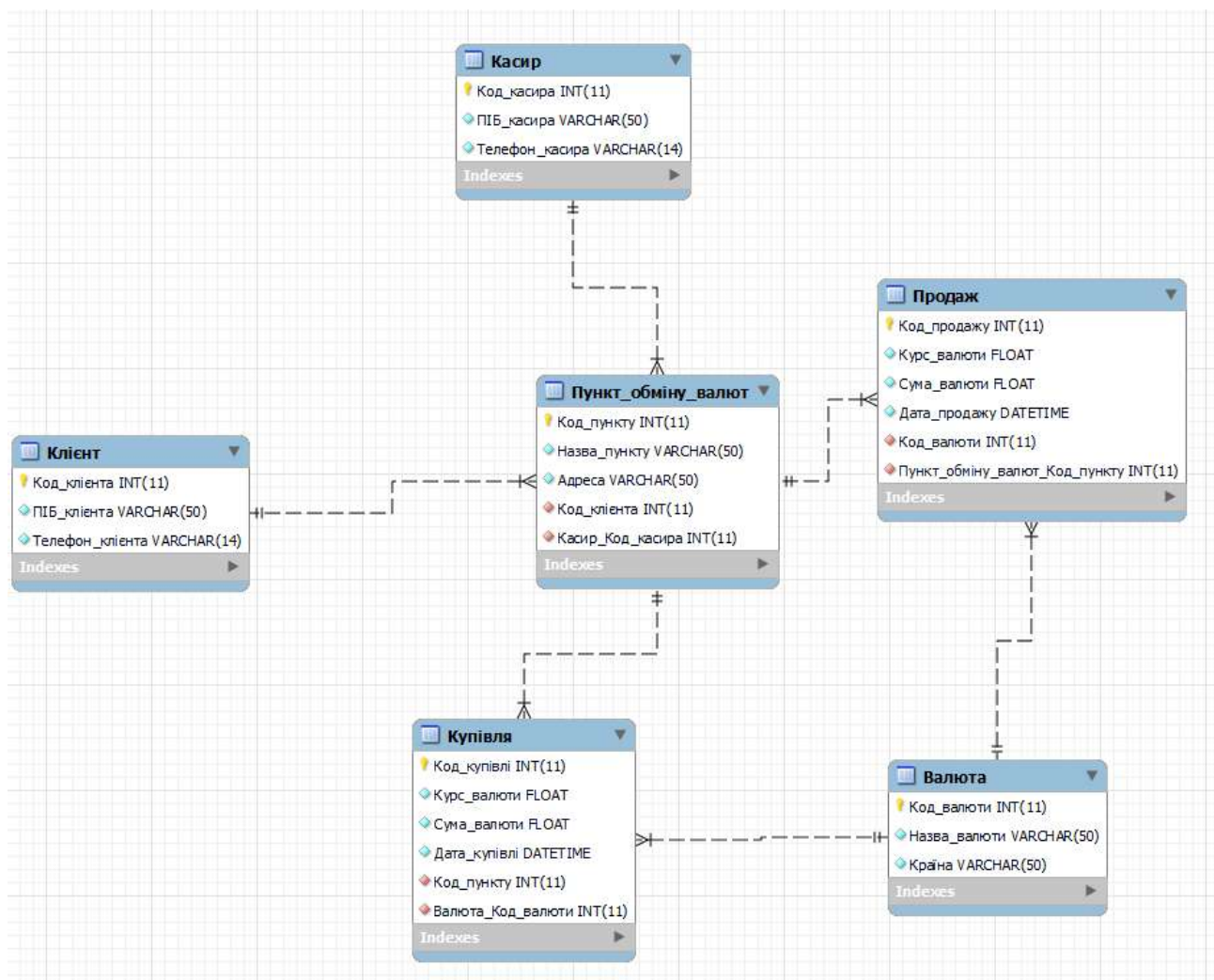


Рис. 3.4. Схема бази даних у СУБД MySQL

Таблиця 3.9

Опис таблиць бази даних у СУБД MySQL

№ п/п	Назва таблиці	Назва стовпця	Тип даних	Властивості
1	Касир	Код_касира	INT	Primary key
2	Касир	ПІБ_касира	VARCHAR(50)	NOT NULL
3	Касир	Телефон_касира	VARCHAR(14)	NOT NULL
4	Валюта	Код_валюти	INT	Primary key
5	Валюта	Назва_валюти	VARCHAR(50)	NOT NULL
6	Валюта	Країна	VARCHAR(50)	NOT NULL
7	Клієнт	Код_клієнта	INT	Primary key
8	Клієнт	ПІБ_клієнта	VARCHAR(50)	NOT NULL

9	Клієнт	Телефон_клієнта	VARCHAR(14)	NOT NULL
10	Пункт_обміну_валют	Код_пункту	INT	Primary key
11	Пункт_обміну_валют	Назва_пункту	VARCHAR(50)	NOT NULL
12	Пункт_обміну_валют	Адреса_пункту	VARCHAR(50)	NOT NULL
13	Пункт_обміну_валют	Код_клієнта	INT	Foreign key; NOT NULL
14	Пункт_обміну_валют	Код_касира	INT	Foreign key; NOT NULL
15	Купівля	Код_купівлі	INT	Primary key
16	Купівля	Код_валюти	INT	Foreign key; NOT NULL
17	Купівля	Курс_валюти	FLOAT	NOT NULL
18	Купівля	Сума_валюти	FLOAT	NOT NULL
19	Купівля	Дата_купівлі	DATETIME	NOT NULL
20	Купівля	Код_пункту	INT	Foreign key; NOT NULL
21	Продаж	Код_продажу	INT	Primary key
22	Продаж	Код_валюти	INT	Foreign key; NOT NULL
23	Продаж	Курс_валюти	FLOAT	NOT NULL
24	Продаж	Сума_валюти	FLOAT	NOT NULL
25	Продаж	Дата_продажу	DATETIME	NOT NULL
26	Продаж	Код_пункту	INT	Foreign key; NOT NULL

Таким чином розробка бази даних в СУБД MySQL є переходом до сучасного, потужного та економічно ефективного рішення для обліку валюти. Її відкрита природа забезпечує доступ до широкого співтовариства розробників та

ресурсів, а також дозволяє скоротити витрати на ліцензування. Висока продуктивність і масштабованість MySQL роблять його ідеальним вибором.

Висновки до розділу III

У розділі було розглянуто ключові аспекти, що забезпечують ефективне функціонування програмного комплексу для обліку операцій купівлі-продажу валюти в пунктах обміну АТ «ПУМБ». Інформаційне забезпечення виступає фундаментальною складовою всієї системи, оскільки визначає структуру, обсяг, форму подання, зберігання й обробки даних, необхідних для підтримки основних бізнес-процесів.

На основі аналізу предметної області було сформовано інфологічну модель, яка відобразила основні інформаційні об'єкти (операції купівлі-продажу, касові залишки, курси валют, звіти, користувачі) та взаємозв'язки між ними. Ця модель дозволила створити структуру бази даних, оптимізовану для виконання запитів, генерації звітів і забезпечення надійного зберігання інформації.

Особлива увага приділялася питанням цілісності, актуальності та безпеки даних. У рамках розробки були визначені правила ведення журналів транзакцій, контролю доступу до інформації, а також забезпечення синхронізації даних з центральною системою «Операційний день банку».

Таким чином, організація інформаційного забезпечення є критичним етапом у створенні ефективного, масштабованого та надійного програмного продукту. Від якості реалізації цього етапу залежить стабільність, точність і швидкість виконання всіх функцій системи, а також зручність її використання в реальних умовах роботи пункту обміну валют.

РОЗДІЛ IV

РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ЗАДАЧІ

4.1. Обґрунтування вибору засобів розробки

Обґрунтування вибору засобів розробки програмного забезпечення є важливим етапом реалізації дипломного проєкту, оскільки саме ці засоби визначають технічні можливості, продуктивність, зручність розробки, а також подальшу підтримку і масштабованість створеної системи. У даній роботі для створення програмного забезпечення обліку операцій купівлі-продажу валюти в пунктах обміну АТ «ПУМБ» було обрано середовище розробки Visual Studio 2019, платформу Windows Forms та мову програмування C#. Такий вибір має низку технічних, економічних і практичних переваг, які детально обґрунтовуються нижче.

Visual Studio 2019 (рис. 4.1) є однією з найпотужніших та найпопулярніших інтегрованих середовищ розробки (IDE) від компанії Microsoft. Вона підтримує широкий спектр мов програмування, включаючи C#, має вбудовані інструменти для розробки, налагодження, тестування та розгортання програм [8].

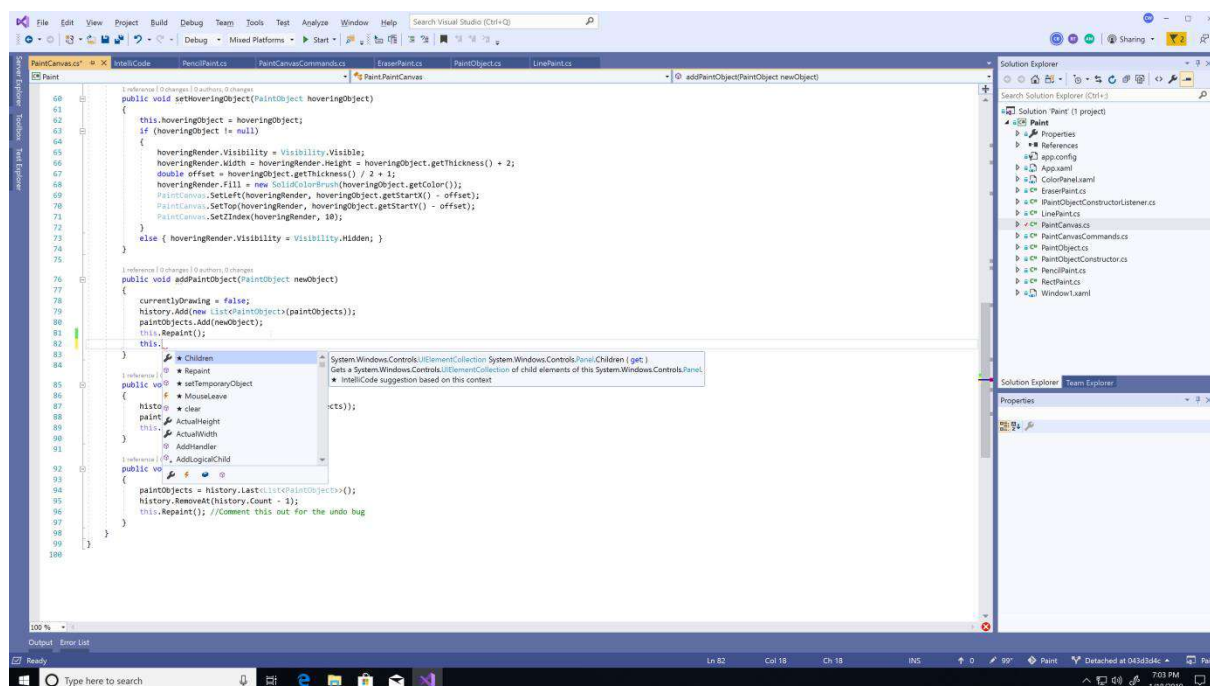


Рис. 4.1. Графічний інтерфейс IDE Visual Studio 2019

До переваг Visual Studio 2019 можна віднести наступне:

- зручний графічний інтерфейс, який дозволяє швидко створювати форми, налаштовувати події та обробники;
- підтримка IntelliSense – автоматичне доповнення коду, що прискорює процес розробки та зменшує ймовірність помилок;
- Інтеграція з системами контролю версій (наприклад, Git), що полегшує колективну розробку;
- потужні засоби налагодження з підтримкою точок зупину, аналізу змінних, відстеження викликів тощо;
- широке співтовариство та підтримка, наявність документації, форумів і прикладів.

Visual Studio 2019 також надає змогу легко інтегруватися з SQL Server, Microsoft Access або іншими СУБД, що особливо важливо у контексті розробки програм з доступом до бази даних.

Windows Forms (WinForms, рис. 4.2) – це платформа для створення настільних застосунків з графічним інтерфейсом користувача, яка є частиною Microsoft .NET Framework [9].

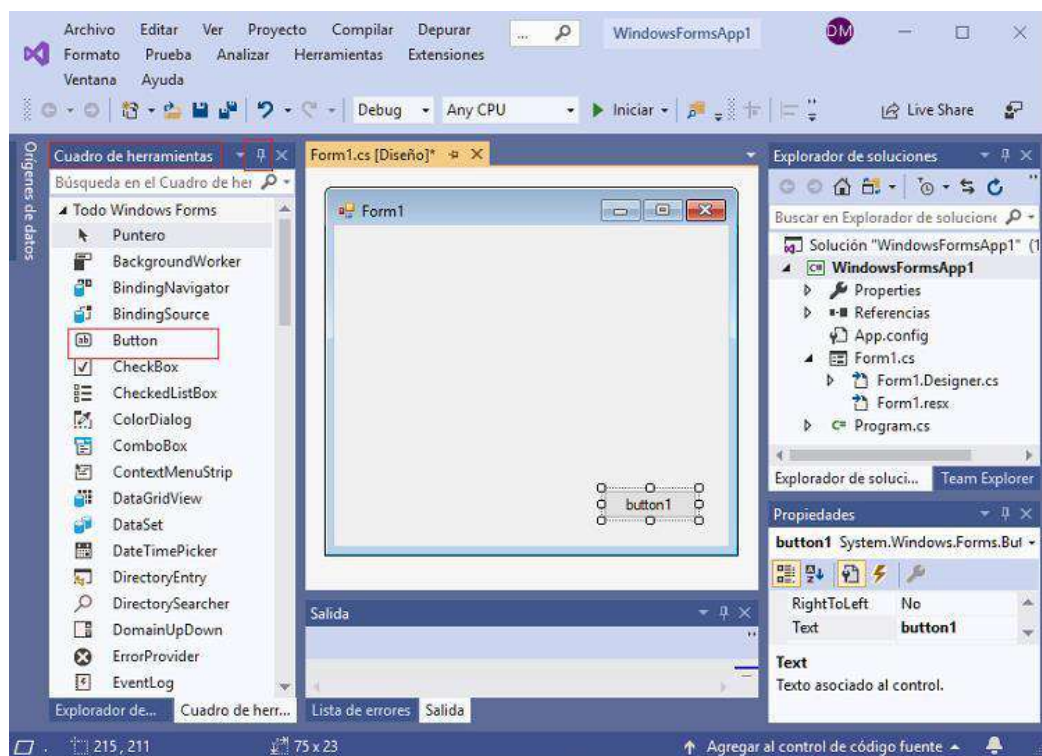


Рис. 4.2. Графічний інтерфейс засобів розробки WinForms

До переваг WinForms можна віднести наступне:

- Простота створення інтерфейсу завдяки "drag-and-drop" механізму елементів керування;
- Швидке навчання та розробка, особливо для невеликих і середніх систем, що не потребують високої складності;
- Наявність широкого набору стандартних компонентів (таблиці, кнопки, поля вводу, вкладки тощо), які легко налаштовуються;
- Легка інтеграція з базами даних, зокрема Access, MySQL, SQL Server;
- Можливість реалізації друку, експорту таблиць, діалогів налаштувань та інших функцій без потреби використання сторонніх бібліотек.

Для цілей розробки автоматизованого робочого місця оператора обмінного пункту, Windows Forms є цілком достатнім і оптимальним варіантом.

C# є сучасною, об'єктно-орієнтованою мовою програмування, яка розроблена Microsoft спеціально для платформи .NET. Вона поєднує у собі простоту синтаксису, потужність інструментів та високу продуктивність. Архітектуру .NET Framework, складовою частиною якої є мова програмування C#, показано на рис. 4.3 [10].

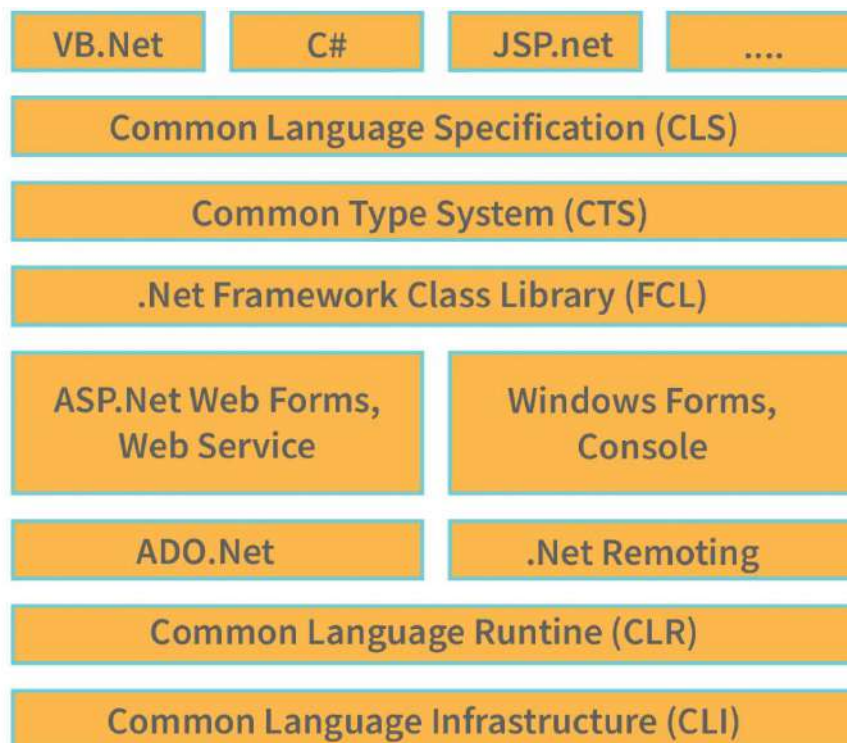


Рис. 4.3. Схематичне представлення архітектури .NET Framework

До переваг C# відноситься:

- Гарна підтримка ООП, що дозволяє будувати гнучку та масштабовану архітектуру програми;
- Потужна стандартна бібліотека, яка охоплює всі основні задачі: робота з файлами, базами даних, мережею, шифруванням, багатопотоковістю тощо;
- Безпечне управління пам'яттю, що мінімізує ризики помилок, пов'язаних із виділенням та очищенням пам'яті;
- Швидке виконання завдяки компіляції у проміжний код (IL), який виконується в середовищі CLR (Common Language Runtime);
- Підтримка LINQ, яка значно полегшує роботу з колекціями та базами даних;
- Актуальність мови та активна підтримка з боку спільноти й Microsoft.

Завдяки використанню C# розробник отримує можливість швидко створювати зручні у використанні додатки, що відповідають сучасним вимогам до надійності, продуктивності та безпеки.

Комбінація Visual Studio 2019, Windows Forms і мови C# дозволяє ефективно реалізувати поставлені завдання в рамках дипломної роботи. Такий вибір забезпечує оптимальне співвідношення між функціональністю, стабільністю, простотою розробки та технічною підтримкою. Обрані інструменти повністю відповідають вимогам предметної області та дозволяють створити зручне, продуктивне й масштабоване програмне забезпечення для автоматизації валютно-обмінних операцій у банківських установах.

У процесі розробки програмного забезпечення для автоматизації обліку операцій купівлі-продажу валюти в пунктах обміну АТ «ПУМБ» було застосовано принципи об'єктно-орієнтованого програмування (ООП) та концепції SOLID [11]. Це дозволило створити гнучку, масштабовану й підтримувану архітектуру програмного продукту, яка легко адаптується до змін функціональних вимог та підтримує розширення функціоналу.

ООП — це парадигма програмування, яка моделює програмні сутності як об'єкти, що мають стан (властивості) та поведінку (методи) [12]. Основні принципи ООП, застосовані в розробці програмного продукту, наведено нижче:

Інкапсуляція. У розробці додатку всі сутності (наприклад, «Операція обміну», «Курс валют», «Користувач», «Каса») були реалізовані у вигляді класів, які приховують внутрішні механізми обробки даних та надають лише публічні методи для взаємодії. Це дозволяє зменшити кількість помилок, пов'язаних з некоректним використанням об'єктів, та забезпечує гнучкість у зміні внутрішньої реалізації.

Наслідування. Для зменшення дублювання коду були створені базові класи. Наприклад, загальний клас `Operation` може бути базовим для таких класів, як `BuyOperation` та `SellOperation`, що реалізують специфіку купівлі та продажу валюти відповідно. Це дозволяє ефективно розширювати функціонал без переписування базової логіки.

Поліморфізм. Поліморфізм дозволив створювати єдині інтерфейси для роботи з різними типами об'єктів. Наприклад, метод друку або збереження операцій реалізується спільно для різних типів валютних транзакцій. Це полегшує додавання нових типів операцій без зміни існуючої архітектури.

Абстракція. Система оперує абстрактними сутностями, такими як «операція», «звіт», «користувач», не зосереджуючись на технічних деталях збереження або обробки. Такий підхід спрощує структуру програми та дозволяє реалізувати її частини незалежно одна від одної.

SOLID — це набір п'яти принципів об'єктно-орієнтованого проектування, який покликаний зробити систему легкою для супроводу, масштабування та повторного використання [13]. У рамках цієї дипломної роботи ці принципи реалізовані наступним чином:

Single Responsibility Principle (Принцип єдиної відповідальності). Кожен клас виконує лише одну чітко визначену функцію. Наприклад, клас `CurrencyRateService` відповідає виключно за завантаження та оновлення курсів валют, тоді як клас `ExchangeOperationManager` — за обробку валютних операцій.

Open/Closed Principle (Принцип відкритості/закритості). Програма легко розширюється без зміни існуючого коду. Наприклад, для додавання нових типів операцій достатньо реалізувати новий клас, що наслідує базовий інтерфейс, без зміни існуючих класів.

Liskov Substitution Principle (Принцип підстановки Лісков). Усі підкласи можуть замінювати базовий клас без порушення коректності роботи програми. Наприклад, класи BuyOperation та SellOperation можуть використовуватись у контексті базового типу ExchangeOperation без змін у логіці обробки.

Interface Segregation Principle (Принцип розділення інтерфейсу). Було створено вузькоспеціалізовані інтерфейси. Наприклад, окремі інтерфейси для операцій друку (IPrintable), роботи з БД (IDatabaseEntity) чи журналювання (ILogable), щоб класи реалізували лише той функціонал, який їм дійсно потрібен.

Dependency Inversion Principle (Принцип інверсії залежностей). Замість створення залежностей від конкретних реалізацій у коді використовувалися абстракції (інтерфейси). Це дозволило використовувати впровадження залежностей (Dependency Injection), що особливо зручно для тестування та модульності коду.

Застосування принципів ООП та SOLID дало змогу побудувати логічну, гнучку та масштабовану структуру програмного забезпечення, яке легко підтримувати та розвивати. Такий підхід дозволив чітко розділити відповідальності між компонентами системи, уникнути дублювання коду, полегшити процес тестування і розширення функціоналу. В результаті розроблене програмне забезпечення не лише відповідає вимогам дипломного проєкту, а й демонструє приклад якісного та професійного проєктування сучасних інформаційних систем.

У контексті розробки програмного забезпечення для обліку операцій купівлі-продажу валюти в пунктах обміну валют АТ «ПУМБ» особливу роль відіграють технології для зберігання, обробки та візуалізації даних. Дві з таких ключових технологій у середовищі С# — це ADO.NET та OpenGL. Перша забезпечує зручну та ефективну взаємодію з базами даних, а друга — дає змогу

реалізувати графічне подання інформації у вигляді діаграм, графіків та інтерактивних візуальних елементів.

ADO.NET є невіддільною частиною платформи .NET і використовується для доступу до реляційних баз даних. У системах банківського обліку це критично важливо, оскільки всі фінансові операції мають бути точно збережені, доступні для фільтрації, звітності та аудиту. Основне призначення ADO.NET — це виконання SQL-запитів, зчитування результатів, керування транзакціями та організація зв'язку між додатком і базою даних. У розробці ПЗ для обміну валют за допомогою ADO.NET реалізується імпорт та експорт даних про валютні операції між обліковою системою пункту обміну валют і основною банківською системою, наприклад, модулем «Операційний день банку».

Однією з важливих функцій ADO.NET у цій системі є підтримка параметризованих запитів, що дозволяє уникнути SQL-ін'єкцій та забезпечити безпечну обробку даних. Це особливо важливо у фінансовому секторі, де захист даних є пріоритетом. Крім того, ADO.NET забезпечує підтримку роботи в автономному режимі за допомогою об'єктів DataSet і DataTable. Завдяки цьому система може продовжувати роботу навіть при короткочасному відключенні від головного сервера — дані кешуються, а після відновлення зв'язку синхронізуються з базою.

У банківському застосунку ADO.NET дозволяє реалізувати широкий спектр функціональностей: зберігання інформації про щоденні курси валют, облік кожної операції купівлі чи продажу, формування фінансових звітів за різні періоди, а також зручне сортування та фільтрацію записів за типом валюти, відділенням, оператором чи періодом. Крім того, бібліотека підтримує масштабування — її можна використовувати як для локальних, так і для мережеских баз даних (наприклад, на MySQL чи MS SQL Server), що дає змогу легко адаптувати рішення для різних розмірів мережі банківських відділень.

Іншою важливою складовою системи є OpenGL — API для рендерингу графіки, який у середовищі C# зазвичай реалізується через бібліотеку OpenTK. У межах дипломної роботи OpenGL може бути використаний для побудови

графіків, які відображають зміну курсів валют у часі. Це може бути як окреме аналітичне вікно для внутрішнього користування працівників банку, так і частина звітнього модуля, що дозволяє візуально оцінити коливання курсів та кількість проведених операцій за певний період.

OpenGL дозволяє створювати як прості 2D-графіки (лінійні, стовпчикові, кругові), так і складніші 3D-візуалізації. У банківській сфері найчастіше використовуються 2D-інструменти для візуального аналізу, зокрема динаміка курсу долара, євро чи інших валют за день, тиждень чи місяць. Графічне представлення значно підвищує розуміння ситуації та прийняття управлінських рішень, порівняно зі звичайними числовими таблицями.

Використання OpenGL у цьому контексті має кілька переваг. По-перше, це висока продуктивність і можливість апаратного прискорення рендерингу — що дозволяє швидко обробляти великі об'єми даних. По-друге, це широкі можливості налаштування: колір, масштабування, інтерфейс взаємодії користувача. Це відкриває шлях до створення більш дружніх та інформативних інтерфейсів, що можуть відображати, наприклад, загальну суму операцій купівлі-продажу у вигляді діаграми з можливістю вибору типу валюти або діапазону дат.

Інтеграція OpenGL із Windows Forms або іншими віконними інтерфейсами в C# дозволяє вбудовувати графічні вікна прямо у візуальні елементи програми — форми, панелі тощо. Таким чином, система стає не лише інструментом для зберігання та обробки даних, а й потужним засобом візуального аналізу для керівного персоналу та аналітиків. Наприклад, керівник пункту обміну валют зможе наочно оцінити, в які години доби припадає найбільший обсяг операцій, яка валюта користується найбільшим попитом, чи як зміни курсу впливають на обсяг купівлі.

У підсумку, поєднання ADO.NET та OpenGL у C# створює ефективну архітектуру програмного забезпечення: ADO.NET відповідає за точну, безпечну та гнучку роботу з даними, тоді як OpenGL забезпечує потужні можливості для побудови інтерактивних і візуально привабливих інтерфейсів. Такий підхід відповідає сучасним стандартам фінансового програмного забезпечення, де

цінується не лише функціональність, а й зручність користування, швидкий доступ до візуалізованих даних і можливість оперативного аналізу. У поєднанні ці інструменти дозволяють створити комплексну систему, яка враховує як потреби оператора пункту обміну, так і вимоги адміністративного контролю з боку банку.

У межах розробки програмного забезпечення для обліку операцій купівлі-продажу валюти в пунктах обміну валют на прикладі АТ «ПУМБ» вибір засобів розробки був зумовлений необхідністю створення надійного, зручного, функціонального та масштабованого рішення. Виходячи з цього, як середовище розробки було обрано Visual Studio 2019, мову програмування С#, технологію створення інтерфейсу Windows Forms, а також ADO.NET для роботи з базами даних.

Visual Studio 2019 є потужним інтегрованим середовищем розробки (IDE), що забезпечує підтримку всіх етапів розробки — від проектування до компіляції, налагодження та тестування. Воно має широкий набір інструментів для створення як десктопних, так і серверних застосунків. Крім того, Visual Studio підтримує зручне керування версіями коду, інтеграцію з системами контролю версій та можливість роботи в команді.

Мова програмування С# була обрана з огляду на її сучасність, підтримку об'єктно-орієнтованого програмування (ООП), наявність широкої бібліотеки класів і глибоку інтеграцію з платформою .NET. Завдяки цьому С# дозволяє створювати структуроване, підтримуване і розширюване програмне забезпечення, яке легко адаптувати під вимоги банківської сфери.

Windows Forms — це технологія для побудови класичних десктопних інтерфейсів, що ідеально підходить для створення прикладних програм із графічним інтерфейсом користувача. Вона забезпечує просту реалізацію форм, діалогових вікон, таблиць, кнопок і панелей, що дозволяє розробити зрозумілий та інтуїтивний інтерфейс для працівників пунктів обміну валют.

Для реалізації збереження, пошуку та обробки даних застосовується ADO.NET, яке забезпечує ефективну роботу з базами даних (наприклад, MySQL, MS SQL Server, Access). Цей інструмент дозволяє виконувати SQL-запити,

обробляти транзакції, формувати звіти, а також реалізовувати імпорт та експорт даних між локальними системами та центральним банківським сервером.

Такий вибір інструментів дозволяє створити надійне програмне забезпечення, що враховує реальні потреби пункту обміну валют і забезпечує гнучкість, масштабованість та безпеку банківських операцій.

Таким чином було виконано обґрунтування засобів розробки програмного забезпечення.

4.2. Реалізація програмного забезпечення на мові C#

У межах дипломної роботи було створено повнофункціональне програмне забезпечення для обліку операцій купівлі-продажу валюти в пунктах обміну валют, яке відповідає потребам сучасного банківського відділення. Програма розроблена на прикладі АТ «ПУМБ» з урахуванням специфіки банківських процесів, регуляторних вимог, а також принципів зручності та швидкодії для щоденної роботи операторів. Результатом розробки стало інтуїтивно зрозуміле десктопне рішення (рис. 4.4) на основі C#, Windows Forms та ADO.NET.

Курси Валют

Курси валют пункту обміну валют АТ "ПУМБ"

Кількість: 1

Джерело: nbi

Дата: п'ятниця, 28 лютого 2025 г.

Перша валюта: RUB, USD, EUR, GBP, UAH

Друга валюта: RUB, USD, EUR, GBP, UAH

Сконвертувати

1 гривня (UAH) в доларах (USD)

Джерело: за курсом НБУ Дата оновлення: 2025-02-28 20:01 UTC

Я маю: 1 UAH Я отримаю: 0.02 USD

1 UAH = 0.0241 USD 1 USD = 41.514 UAH

1 долар США в гривні

Рис. 4.4. Робота з конвертором валют

Одним з центральних функціональних блоків створеного програмного забезпечення є модуль конвертації валюти, що реалізує важливу частину обслуговування клієнтів у пункті обміну валют. Цей модуль дозволяє не лише швидко і точно здійснювати обмін валют, а й інтегрує актуальні курси з різних джерел, таких як НБУ та внутрішній курс АТ «ПУМБ».

Після запуску додатку користувач потрапляє до головного вікна, де через верхнє меню або навігаційну панель обирає вкладку чи кнопку «Конвертація валюти». У результаті відкривається спеціальне вікно для здійснення операцій із перерахунку грошових коштів з однієї валюти в іншу. Дизайн вікна витриманий у стриманому стилі, що відповідає корпоративному стилю банківської установи: переважають спокійні кольори, логічне групування елементів, читабельні шрифти.

У верхній частині вікна розташовані основні поля для введення вихідних даних, що дозволяють оператору задати всі необхідні параметри для виконання конвертації:

- Поле введення суми — дозволяє ввести кількість валюти, яку необхідно конвертувати. Це поле підтримує лише числові значення, реалізовано перевірку коректності введення (валидацію) для запобігання введенню символів чи негативних значень.

- Спадаючий список джерел курсу (рис. 4.5) — дає можливість обрати джерело, з якого буде братись курс для обміну. Доступні варіанти: Національний банк України (НБУ) та внутрішній курс АТ «ПУМБ». Цей функціонал забезпечує гнучкість налаштування політики обміну залежно від банківських умов, а також відкриває можливість подальшого розширення (наприклад, інтеграція з міжнародними джерелами курсів).

- Поле вибору дати курсу — користувач має змогу обрати актуальну або попередню дату, за якою буде застосовано валютний курс. Це корисно для обробки операцій, що відбулися раніше, або при формуванні звітності. Обираючи іншу дату, система автоматично запитує курс.

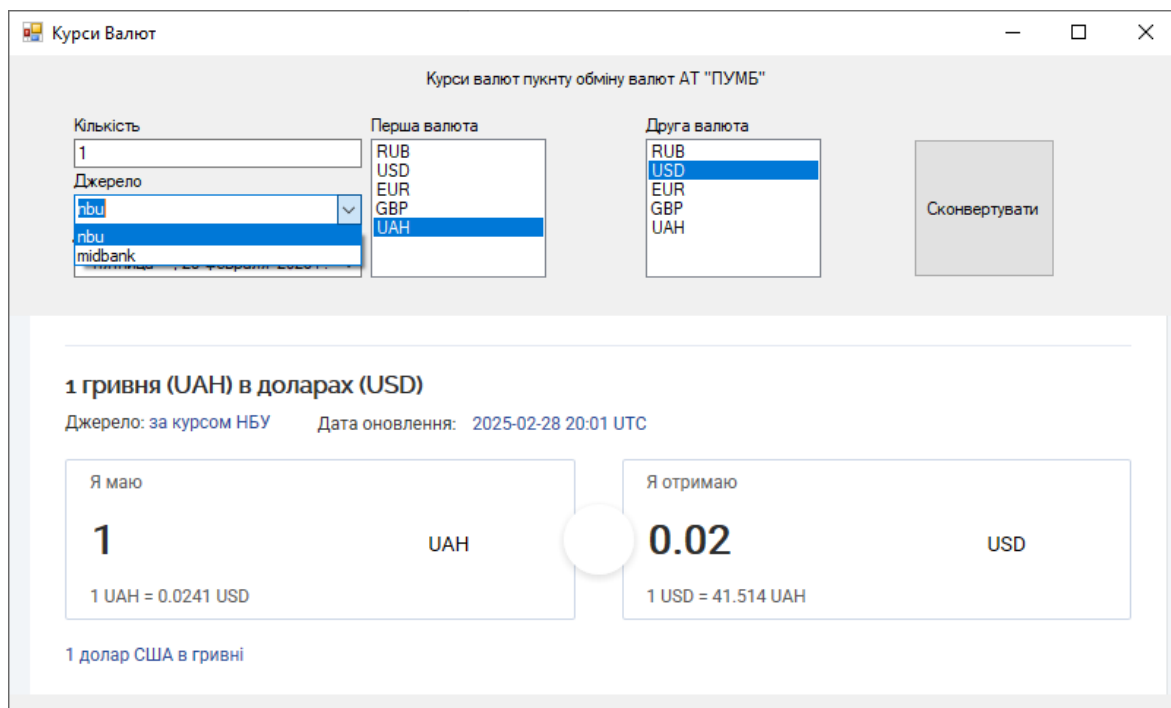


Рис. 4.5. Керування інтерфейсом для обрання джерела інформації

– Вибір валюти джерела — у наступному спадному списку оператор зазначає, з якої саме валюти буде здійснюватися конвертація. Наприклад, USD, EUR, GBP або UAH. Доступний перелік динамічно оновлюється, що дозволяє підтримувати лише ті валюти, з якими реально працює відділення.

– Вибір цільової валюти — у сусідньому спадному списку обирається валюта, в яку необхідно здійснити обмін. Механізм не дозволяє обрати однакові валюти для вихідної та цільової позиції, а у разі помилки система виводить відповідне повідомлення.

– Кнопка "Сконвертувати" — активує обчислення результату. При натисканні система зчитує введені дані, звертається до обраного джерела курсу, проводить математичну операцію та виводить результат у нижньому інформаційному блоці.

У нижній частині форми після натискання кнопки з'являється блок, у якому представлено результат операції. Він формується у вигляді звітного повідомлення, що включає кілька важливих інформаційних компонентів:

– вказується сума, яку було введено, та валюта, з якої здійснювався обмін;

- показується одержана сума у цільовій валюті, обрахована з точністю до двох знаків після коми;
- наводиться інформація про джерело курсу (наприклад, «Курс згідно з НБУ» або «Курс банку ПУМБ»);
- вказується дата курсу (рис. 4.6), на основі якого виконано операцію;
- відображається повне формулювання дії, наприклад: «Було конвертовано 100 USD у 3921.40 UAH за курсом НБУ на 10.06.2025».

Крім текстового опису, є можливість роздрукувати чек операції або зберегти його у PDF-форматі, що реалізовано через інтеграцію з функціоналом Windows Print та PDF Writer. Це дозволяє оператору зберігати звітність або видавати клієнтам підтвердження проведеної операції.

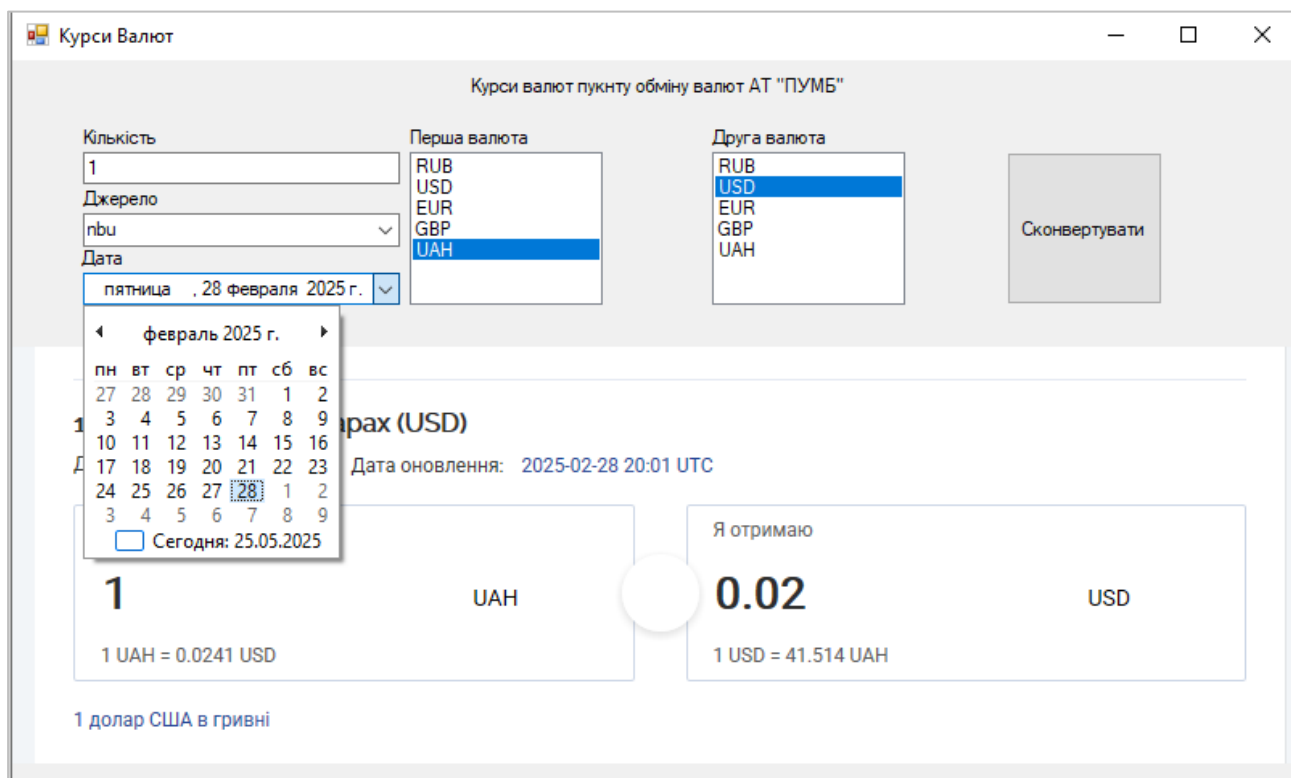


Рис. 4.6. Керування інтерфейсом для обрання дати курсу

Оновлення курсів відбувається за допомогою інтеграції з базою даних та/або веб-інтерфейсом (у майбутніх версіях можлива інтеграція з API НБУ). При виборі джерела «НБУ» система автоматично виконує запит до сервера для отримання останніх офіційних курсів. У разі відсутності підключення —

відображається попереднє збережене значення з повідомленням про його дату та необхідність оновлення.

Операція конвертування іноземної валюти в національну (рис. 4.7) є однією з ключових функцій програмного забезпечення, розробленого в межах даної дипломної роботи. Вона дозволяє швидко, точно та зручно здійснити обмін обраної іноземної валюти (наприклад, доларів США або євро) на українську гривню з урахуванням актуального курсу. Користувач вводить суму, обирає джерело курсу (Національний банк України або внутрішній курс ПУМБ), вказує дату курсу, а також валюту, з якої здійснюється конвертація. Після натискання кнопки «Сконвертувати» система обчислює результат з урахуванням усіх введених параметрів і відображає повну інформацію: з якої у яку валюту здійснено обмін, яка саме сума була сконвертована, курс обміну, дата оновлення курсу та джерело інформації. Такий підхід забезпечує прозорість операцій, спрощує облік валютних операцій у пунктах обміну та підвищує довіру клієнтів до сервісу.

The screenshot displays a web application window titled "Курси Валют" (Currency Rates). The main heading is "Курси валют пункту обміну валют АТ 'ПУМБ'" (Currency rates of the currency exchange point PJSC 'PUMB').

The interface includes several input fields and dropdown menus:

- Кількість (Quantity):** A text input field containing the value "100".
- Джерело (Source):** A dropdown menu with "nbu" selected.
- Дата (Date):** A date picker showing "воскресенье, 25 мая 2025 г." (Sunday, 25 May 2025).
- Перша валюта (First currency):** A dropdown menu with "USD" selected from a list including RUB, EUR, GBP, and UAH.
- Друга валюта (Second currency):** A dropdown menu with "UAH" selected from a list including RUB, USD, EUR, GBP, and UAH.
- Кнопка:** A button labeled "Сконвертувати" (Convert).

Below the input fields, the application displays the conversion result:

100 доларів (USD) в гривні (UAH)
Джерело: за курсом НБУ | Дата оновлення: 2025-05-25 07:01 UTC

The result is visualized in two boxes:

- Я маю (I have):** A box showing "100" in large font, with "USD" to its right. Below it, the text "1 USD = 41.4999 UAH" is displayed.
- Я отримаю (I receive):** A box showing "4 149.99" in large font, with "UAH" to its right. Below it, the text "1 UAH = 0.0241 USD" is displayed.

At the bottom left, there is a small text element: "100 гривень в доларах" (100 hryvnia in dollars).

Рис. 4.7. Конвертування іноземної валюти в національну

Конвертування національної валюти в іноземну (рис. 4.8) здійснюється за аналогічним принципом: користувач обирає гривню як вихідну валюту, вказує

суму, джерело курсу, дату та бажану іноземну валюту для обміну. Після цього програма обчислює відповідну суму в обраній валюті та надає детальну інформацію про проведену операцію.

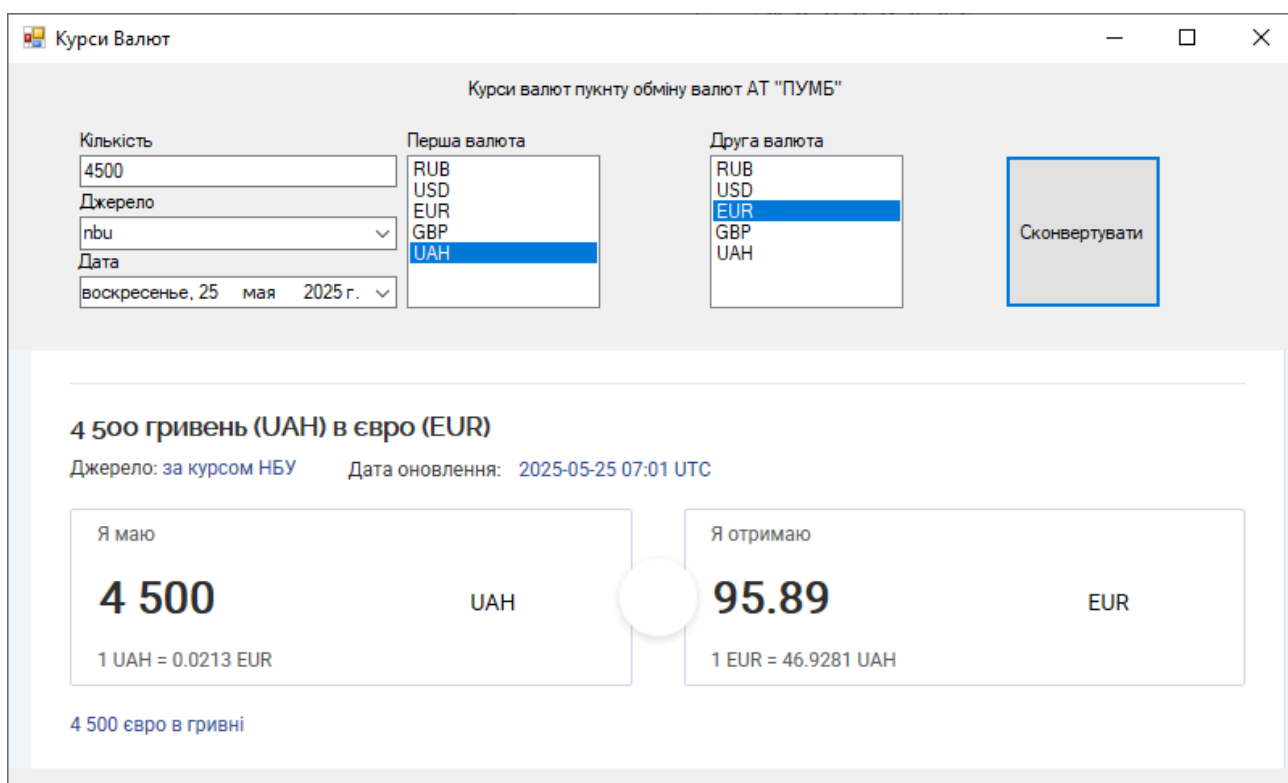


Рис. 4.8. Конвертування національної валюти в іноземну

Опис вкладок додатку (рис. 4.9) є важливою частиною розуміння структури та функціональності розробленого програмного забезпечення, що реалізоване в межах дипломної роботи на тему «Розробка програмного забезпечення для обліку операцій купівлі-продажу валюти в пунктах обміну валют на прикладі АТ "ПУМБ"». У програмі реалізовано інтуїтивно зрозумілу навігацію за допомогою вкладок, кожна з яких виконує окрему функцію і забезпечує взаємодію користувача з тими чи іншими аспектами системи. Серед ключових вкладок особливу увагу слід приділити вкладці «Налаштування» та вкладці «Робота з базою даних», яка розділена на підвкладки відповідно до логічної структури бази даних.

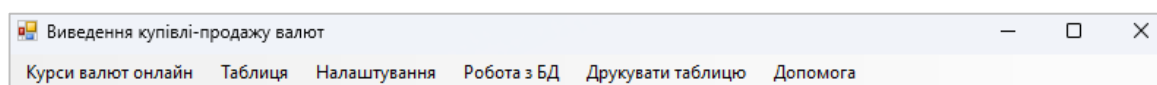


Рис. 4.9. Головне меню програми

Вкладка «Налаштування» реалізує персоналізацію інтерфейсу програми, дозволяючи користувачу адаптувати зовнішній вигляд системи до своїх візуальних вподобань. Подібна можливість позитивно впливає на користувацький досвід і сприяє зменшенню втомлюваності під час тривалої роботи.

Налаштування кольорової схеми: користувач може обрати з кількох попередньо визначених тем оформлення — світла, темна, корпоративна. Кольорова схема застосовується до фону форм, кнопок, таблиць і панелей. Зміна кольору відбувається в реальному часі, без потреби перезапуску програми.

Вибір шрифту: реалізовано механізм зміни типу та розміру шрифту інтерфейсу користувача. Залежно від уподобань або зору користувача можна обрати стандартний, збільшений або моноширинний шрифт. Шрифт застосовується до всіх форм, включаючи таблиці, кнопки та інформаційні блоки.

Попередній перегляд: у нижній частині вкладки розміщено панель попереднього перегляду, що дає змогу одразу побачити, як виглядатимуть зміни. Це дозволяє уникнути помилкових налаштувань та підвищує зручність.

Кнопка «Застосувати зміни»: лише після натискання цієї кнопки нові налаштування зберігаються у конфігураційному файлі користувача (наприклад, у форматі JSON або XML). При наступному запуску програми інтерфейс автоматично завантажується з урахуванням обраних параметрів.

Ця вкладка демонструє застосування принципів гнучкості і зручності інтерфейсу, що відповідає сучасним стандартам розробки програмного забезпечення.

Окрема вкладка, присвячена взаємодії з базою даних, структурована у вигляді підвкладок, кожна з яких відповідає за роботу з окремою таблицею. Програма використовує технології ADO.NET для підключення до бази даних, обміну даними та збереження змін. Всі зміни проходять через інтерфейс, який реалізує перевірку на коректність введених даних та відповідність структурі таблиці.

Підвкладка «Валюта» призначена для перегляду, додавання, редагування та видалення записів про доступні валюти. У таблиці відображаються такі дані:

назва валюти, код (наприклад, USD, EUR), країна походження, дата останнього оновлення курсу.

На вкладці «Валюта» користувач може:

- Додати нову валюту з унікальним кодом;
- Змінити назву або країну;
- Оновити дату актуальності;
- Видалити непотрібну валюту (лише якщо вона не задіяна в операціях).

Ця функціональність дозволяє гнучко керувати валютним довідником без потреби у прямому доступі до бази.

На підвкладці «Касир» адміністратор має змогу управляти даними касирів, які працюють у пунктах обміну валют. Інтерфейс дозволяє ввести повне ім'я, ідентифікатор, дату початку роботи, рівень доступу та контактну інформацію.

Реалізовано пошук касира за ПІБ або ID. Додатково можливе тимчасове деактивування облікового запису у випадку відпустки чи звільнення.

Підвкладка «Клієнт» призначена для зберігання інформації про клієнтів банку, які здійснюють операції обміну. Поля включають ПІБ, ідентифікаційний код, тип документа, серію та номер документа, а також історію операцій.

Завдяки цій вкладці можна швидко знайти клієнта, побачити його історію угод, що особливо корисно при розгляді суперечливих ситуацій або при необхідності повторного підтвердження даних.

Підвкладка «Купівля» містить усі дані про операції купівлі валюти. Таблиця включає ідентифікатор операції, клієнта, касира, дату і час, валюту, курс, суму, а також пункт обміну, де відбулася операція.

Підвкладка «Продаж» аналогічна попередній, ця вкладка призначена для обліку продажу валюти клієнтам. Відображає повний набір параметрів, включаючи курс продажу, дату, суму, ідентифікатори касира та клієнта. Є можливість анулювання або повторної генерації квитанції.

Підвкладка «Пункт обміну валют» дозволяє керувати довідником відділень, де здійснюються операції обміну валют. У ній зберігаються такі дані:

унікальний код пункту, адреса, номер телефону, графік роботи, статус (активний/неактивний).

Інтерфейс дає змогу змінювати адресу, додавати нові пункти, а також тимчасово деактивувати відділення.

Загальні можливості вкладки «База даних», де кожна підвкладка має стандартний набір функцій:

- таблиця даних з можливістю фільтрації, сортування та пошуку;
- кнопки CRUD (Create, Read, Update, Delete);
- перевірка даних перед збереженням;
- повідомлення про помилки (наприклад, при дублюванні записів);
- журнал змін для відстеження дій користувачів.

Крім того, реалізована автоматична синхронізація з базою даних після кожної операції. Зміни набувають чинності негайно, а також зберігаються у відповідних логах. Це забезпечує цілісність та актуальність інформації, а також дозволяє вести аудит.

Таким чином, вкладки «Налаштування» та «Робота з базою даних» утворюють потужний адміністративний інструментарій у структурі програмного забезпечення. Вони забезпечують зручний доступ до ключових функцій системи, дають змогу персоналізувати інтерфейс і контролювати критично важливі компоненти бази даних без потреби звертання до адміністратора БД або зовнішніх засобів. Це сприяє підвищенню продуктивності працівників, покращенню якості обслуговування клієнтів та ефективному управлінню інформацією в межах діяльності пункту обміну валют.

Висновки до розділу IV

Таким чином, вікно конвертації валют, яке є складовою частиною розробленого програмного забезпечення, реалізує всі необхідні функції для ефективної роботи працівника пункту обміну валют. Його дизайн, функціонал та логіка враховують потреби реальної бізнес-практики, забезпечуючи швидке

обслуговування клієнтів, точність обчислень, актуальність курсів та можливість документального оформлення результатів. Це дозволяє банку підвищити ефективність обслуговування, скоротити час на рутинні операції та забезпечити повну прозорість та контроль у роботі з грошовими коштами.

Програма успішно демонструє реалізацію теоретичних знань та практичних навичок програмування у середовищі .NET, використання принципів ООП, взаємодії з базами даних, реалізації інтерфейсів користувача, обробки введення та виведення інформації. Все це забезпечує високу якість і стабільність програмного продукту, що може бути використаний у реальному банківському середовищі.

ВИСНОВКИ

У процесі виконання дипломної роботи було всебічно досліджено особливості функціонування пунктів обміну валют на прикладі діяльності АТ «ПУМБ», а також проаналізовано існуючі проблеми, пов'язані з автоматизацією та обліком операцій купівлі-продажу іноземної валюти. На основі вивченого матеріалу було запропоновано варіант програмного забезпечення, здатного ефективно вирішити наявні проблеми, що виникають у процесі обслуговування клієнтів у валютних пунктах банку. Зокрема, особливу увагу було приділено побудові логічної структури бази даних, яка лягла в основу модуля автоматизації «Операційного дня банку».

Реалізоване програмне забезпечення забезпечує виконання широкого спектру функцій: від зберігання та обробки операцій купівлі-продажу валюти до автоматичного формування звітів та забезпечення двостороннього обміну даними з банківською інформаційною системою. Особливістю реалізації є можливість експорту та імпорту валютних операцій, що дозволяє значно підвищити швидкість і точність облікових процесів, а також мінімізувати втручання оператора. Програма враховує потреби як працівників банку, так і клієнтів, пропонуючи зручний інтерфейс для здійснення валютних операцій, зокрема — конвертації між різними валютами на основі актуальних курсів НБУ або внутрішнього курсу ПУМБ.

У межах роботи було розроблено базу даних, яка охоплює ключові об'єкти предметної області — валюту, касирів, клієнтів, точки обміну, операції купівлі та продажу. Завдяки використанню таких СУБД як MySQL і MS Access було досягнуто гнучкості та універсальності розгортання програмного забезпечення. Інтерфейс програми дозволяє користувачам не лише переглядати та редагувати дані, а й адаптувати вигляд інтерфейсу під індивідуальні уподобання — змінюючи кольори, шрифти тощо.

Крім того, реалізована система надає можливість покращеного звітування — у тому числі з фіксацією часу операцій та класифікацією за валютами, що

значно підвищує прозорість та контроль усіх процесів. Такі функціональні можливості дозволяють скоротити час обробки інформації, знизити навантаження на персонал та мінімізувати ймовірність помилок.

Загалом, дипломна робота повною мірою реалізувала поставлену мету — розробити ефективне програмне забезпечення для обліку операцій купівлі-продажу валюти в пунктах обміну валют. Об'єктом дослідження виступили інформаційні системи банківських обмінних операцій, предметом — процеси автоматизації обліку таких операцій. Отримані результати свідчать про доцільність впровадження розробленої системи в банківську практику, зокрема в інфраструктуру АТ «ПУМБ».

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Закон України «Про банки і банківську діяльність» від 7 грудня 2000 р. № 2121-III // Відомості Верховної Ради України. – 2001. – № 5–6. – Ст. 30.
2. Гайдаржи В.І. Основи проектування та використання баз даних: Навч. посіб. / В.І. Гайдаржи, О.А. Дацюк. – 2-ге вид. – К.: ІВЦ "Видавництво «Політехніка»". ТОВ "Фірма «Періодика»", 2004. - 256 с.
3. Голицина О.Л. Базы данных: учебное пособие / О.Л. Голицина, Н.В. Максимов, И.И. Попов. – М.: Форум: Инфра-М, 2004. – 352 с.
4. Грофф Дж. SQL: Полное руководство / Дж. Грофф, П. Вайнберг; пер с англ. – 2-е изд., перераб и доп. – К.: ВНУ, 2001. – 816 ст.
5. Дейт К.Дж. Введение в системы баз данных / Дейт К.Дж.; пер. с англ. – 7-е издание. – М.: Издательский дом "Вильямс", 2001. – 1072 с.
6. Клайн К. SQL. Справочник / Клайн К.; пер. с англ. – 2-е издание. – М.: КУДИЦ-ОБРАЗ, 2006. – 832 с.
7. Компания MySQL AB. MySQL. Справочник по языку. / пер. с англ.— М.: Издательский дом "Вильямс", 2005. – 432 с.
8. Конноли Т. Базы данных. Проектирование, реализация и сопровождение. Теория и практика / Т. Конноли, К. Бегг.; пер. с англ. – 3-е издание. – М.: Издательский дом «Вильямс», 2003. – 1440 с.
9. Крёнке Д. Теория и практика построения баз данных / Крёнке Д. – 8-е издание. – СПб.: Питер, 2003. – 800 с.
10. Access help & learning. Microsoft Support. URL: <https://support.microsoft.com/en-us/access>.
11. Microsoft SQL documentation - SQL Server. Microsoft Learn: Build skills that open doors in your career. URL: <https://learn.microsoft.com/en-us/sql/?view=sql-server-ver16>.
12. MySQL documentation. MySQL. URL: <https://dev.mysql.com/doc/>.
13. SQL Server 2019 Administration Inside Out / W. Assaf et al. Pearson Education, 2019. 864 p.

ДОДАТКИ

Лістинг SQL коду для створення бази даних у СКБД MySQL

```
CREATE TABLE Валюта
(
    Код_валюти          INTEGER NOT NULL,
    Назва_валюти       varchar(50) NOT NULL,
    Країна              varchar(50) NOT NULL
);
ALTER TABLE Валюта
ADD PRIMARY KEY (Код_валюти);
CREATE TABLE Касир
(
    Код_касира         INTEGER NOT NULL,
    ПІБ_касира         varchar(50) NOT NULL,
    Телефон_касира     varchar(14) NOT NULL
);
ALTER TABLE Касир
ADD PRIMARY KEY (Код_касира);
CREATE TABLE Клієнт
(
    Код_клієнта        INTEGER NOT NULL,
    ПІБ_клієнта        VARCHAR(50) NOT NULL,
    Телефон_клієнта    varchar(14) NOT NULL
);
ALTER TABLE Клієнт
ADD PRIMARY KEY (Код_клієнта);
CREATE TABLE Купівля
(
    Код_купівлі        INTEGER NOT NULL,
    Курс_валюти        FLOAT NOT NULL,
    Сума_валюти        FLOAT NOT NULL,
    Дата_купівлі       DATETIME NOT NULL,
    Код_пункту         integer NOT NULL,
    Код_валюти         integer NOT NULL
);
ALTER TABLE Купівля
ADD PRIMARY KEY (Код_купівлі);
CREATE TABLE Продаж
(
    Код_продажу        INTEGER NOT NULL,
    Курс_валюти        FLOAT NOT NULL,
    Сума_валюти        FLOAT NOT NULL,
    Дата_продажу       DATETIME NOT NULL,
    Код_пункту         integer NOT NULL,
    Код_валюти         integer NOT NULL
);
ALTER TABLE Продаж
ADD PRIMARY KEY (Код_продажу);
CREATE TABLE Пункт_обміну_валют
(
    Код_пункту         INTEGER NOT NULL,
    Назва_пункту       varchar(50) NOT NULL,
    Адреса             varchar(50) NOT NULL,
    Код_клієнта        integer NOT NULL,
```

```
        Код_касира            integer NOT NULL
);
ALTER TABLE Пункт_обміну_валют
ADD PRIMARY KEY (Код_пункту);
ALTER TABLE Купівля
ADD FOREIGN KEY R_8 (Код_пункту) REFERENCES Пункт_обміну_валют
(Код_пункту);
ALTER TABLE Купівля
ADD FOREIGN KEY R_16 (Код_валюти) REFERENCES Валюта (Код_валюти);
ALTER TABLE Продаж
ADD FOREIGN KEY R_11 (Код_пункту) REFERENCES Пункт_обміну_валют
(Код_пункту);
ALTER TABLE Продаж
ADD FOREIGN KEY R_14 (Код_валюти) REFERENCES Валюта (Код_валюти);
ALTER TABLE Пункт_обміну_валют
ADD FOREIGN KEY R_7 (Код_клієнта) REFERENCES Клієнт (Код_клієнта);
ALTER TABLE Пункт_обміну_валют
ADD FOREIGN KEY R_15 (Код_касира) REFERENCES Касир (Код_касира);
```

Лістинг VBA коду для створення бази даних у СКБД MS Access

```

Dim ERwinWorkspace As Workspace
Dim ERwinDatabase As Database
Dim ERwinTableDef As TableDef
Dim ERwinQueryDef As QueryDef
Dim ERwinIndex As Index
Dim ERwinField As Field
Dim ERwinRelation As Relation
Set ERwinWorkspace = DBEngine.Workspaces(0)
Set ERwinDatabase = ERwinWorkspace.OpenDatabase("sERwinDatabase")
' CREATE TABLE Валюта
Set ERwinTableDef = ERwinDatabase.CreateTableDef("Валюта")
Set ERwinField = ERwinTableDef.CreateField("Код валюти", DB_LONG)
ERwinField.Required = True
ERwinTableDef.Fields.Append ERwinField
Set ERwinField = ERwinTableDef.CreateField("Назва валюти", DB_TEXT, 50)
ERwinField.Required = True
ERwinTableDef.Fields.Append ERwinField
Set ERwinField = ERwinTableDef.CreateField("Країна", DB_TEXT, 50)
ERwinField.Required = True
ERwinTableDef.Fields.Append ERwinField
ERwinDatabase.TableDefs.Append ERwinTableDef
Set ERwinField = ERwinTableDef.Fields("Код валюти")
Call SetFieldProp(ERwinField, "Caption", DB_TEXT, "Код валюти:")
Set ERwinField = ERwinTableDef.Fields("Назва валюти")
Call SetFieldProp(ERwinField, "Caption", DB_TEXT, "Назва валюти:")
Set ERwinField = ERwinTableDef.Fields("Країна")
Call SetFieldProp(ERwinField, "Caption", DB_TEXT, "Країна:")
' CREATE INDEX ХПКВалюта
Set ERwinTableDef = ERwinDatabase.TableDefs("Валюта")
Set ERwinIndex = ERwinTableDef.CreateIndex("ХПКВалюта")
Set ERwinField = ERwinIndex.CreateField("Код валюти")
ERwinIndex.Fields.Append ERwinField
ERwinIndex.Primary = True
ERwinTableDef.Indexes.Append ERwinIndex
' CREATE TABLE Касир
Set ERwinTableDef = ERwinDatabase.CreateTableDef("Касир")
Set ERwinField = ERwinTableDef.CreateField("Код касира", DB_LONG)
ERwinField.Required = True
ERwinTableDef.Fields.Append ERwinField
Set ERwinField = ERwinTableDef.CreateField("ПІВ касира", DB_TEXT, 50)
ERwinField.Required = True
ERwinTableDef.Fields.Append ERwinField
Set ERwinField = ERwinTableDef.CreateField("Телефон касира", DB_TEXT, 14)
ERwinTableDef.Fields.Append ERwinField
ERwinDatabase.TableDefs.Append ERwinTableDef
Set ERwinField = ERwinTableDef.Fields("Код касира")
Call SetFieldProp(ERwinField, "Caption", DB_TEXT, "Код касира:")
Set ERwinField = ERwinTableDef.Fields("ПІВ касира")
Call SetFieldProp(ERwinField, "Caption", DB_TEXT, "ПІВ касира:")
Set ERwinField = ERwinTableDef.Fields("Телефон касира")
Call SetFieldProp(ERwinField, "Caption", DB_TEXT, "Телефон касира:")

```

Продовження Додатку Б

```

' CREATE INDEX XPKКасир
Set ERwinTableDef = ERwinDatabase.TableDefs("Касир")
Set ERwinIndex = ERwinTableDef.CreateIndex("XPKКасир")
Set ERwinField = ERwinIndex.CreateField("Код касира")
ERwinIndex.Fields.Append ERwinField
ERwinIndex.Primary = True
ERwinTableDef.Indexes.Append ERwinIndex
' CREATE TABLE Клієнт
Set ERwinTableDef = ERwinDatabase.CreateTableDef("Клієнт")
Set ERwinField = ERwinTableDef.CreateField("Код клієнта", DB_LONG)
ERwinField.Required = True
ERwinTableDef.Fields.Append ERwinField
Set ERwinField = ERwinTableDef.CreateField("ПІБ клієнта", DB_TEXT, 50)
ERwinField.Required = True
ERwinTableDef.Fields.Append ERwinField
Set ERwinField = ERwinTableDef.CreateField("Телефон клієнта", DB_TEXT, 14)
ERwinField.Required = True
ERwinTableDef.Fields.Append ERwinField
ERwinDatabase.TableDefs.Append ERwinTableDef
Set ERwinField = ERwinTableDef.Fields("Код клієнта")
Call SetFieldProp(ERwinField, "Caption", DB_TEXT, "Код клієнта:")
Set ERwinField = ERwinTableDef.Fields("ПІБ клієнта")
Call SetFieldProp(ERwinField, "Caption", DB_TEXT, "ПІБ клієнта:")
Set ERwinField = ERwinTableDef.Fields("Телефон клієнта")
Call SetFieldProp(ERwinField, "Caption", DB_TEXT, "Телефон клієнта:")
' CREATE INDEX XPKКлієнт
Set ERwinTableDef = ERwinDatabase.TableDefs("Клієнт")
Set ERwinIndex = ERwinTableDef.CreateIndex("XPKКлієнт")
Set ERwinField = ERwinIndex.CreateField("Код клієнта")
ERwinIndex.Fields.Append ERwinField
ERwinIndex.Primary = True
ERwinTableDef.Indexes.Append ERwinIndex
' CREATE TABLE Купівля
Set ERwinTableDef = ERwinDatabase.CreateTableDef("Купівля")
Set ERwinField = ERwinTableDef.CreateField("Код купівлі", DB_LONG)
ERwinField.Required = True
ERwinTableDef.Fields.Append ERwinField
Set ERwinField = ERwinTableDef.CreateField("Код валюти", DB_LONG)
ERwinField.Required = True
ERwinTableDef.Fields.Append ERwinField
Set ERwinField = ERwinTableDef.CreateField("Курс валюти", DB_DOUBLE)
ERwinField.Required = True
ERwinTableDef.Fields.Append ERwinField
Set ERwinField = ERwinTableDef.CreateField("Сума валюти", DB_DOUBLE)
ERwinField.Required = True
ERwinTableDef.Fields.Append ERwinField
Set ERwinField = ERwinTableDef.CreateField("Дата купівлі", DB_DATETIME)
ERwinField.Required = True
ERwinTableDef.Fields.Append ERwinField
Set ERwinField = ERwinTableDef.CreateField("Код пункту", DB_LONG)
ERwinField.Required = True
ERwinTableDef.Fields.Append ERwinField
ERwinDatabase.TableDefs.Append ERwinTableDef
Set ERwinField = ERwinTableDef.Fields("Код купівлі")
Call SetFieldProp(ERwinField, "Caption", DB_TEXT, "Код купівлі:")
Set ERwinField = ERwinTableDef.Fields("Код валюти")
Call SetFieldProp(ERwinField, "Caption", DB_TEXT, "Код валюти:")

```

Продовження Додатку Б

```

Set ERwinField = ERwinTableDef.Fields("Курс валюти")
Call SetFieldProp(ERwinField, "Caption", DB_TEXT , "Курс валюти:")
Set ERwinField = ERwinTableDef.Fields("Сума валюти")
Call SetFieldProp(ERwinField, "Caption", DB_TEXT , "Сума валюти:")
Set ERwinField = ERwinTableDef.Fields("Дата купівлі")
Call SetFieldProp(ERwinField, "Caption", DB_TEXT , "Дата купівлі:")
Set ERwinField = ERwinTableDef.Fields("Код пункту")
Call SetFieldProp(ERwinField, "Caption", DB_TEXT , "Код пункту:")
' CREATE INDEX ХПККупівля
Set ERwinTableDef = ERwinDatabase.TableDefs("Купівля")
Set ERwinIndex = ERwinTableDef.CreateIndex("ХПККупівля")
Set ERwinField = ERwinIndex.CreateField("Код купівлі")
ERwinIndex.Fields.Append ERwinField
ERwinIndex.Primary = True
ERwinTableDef.Indexes.Append ERwinIndex
' CREATE TABLE Продаж
Set ERwinTableDef = ERwinDatabase.CreateTableDef("Продаж")
Set ERwinField = ERwinTableDef.CreateField("Код продажу", DB_LONG)
ERwinField.Required = True
ERwinTableDef.Fields.Append ERwinField
Set ERwinField = ERwinTableDef.CreateField("Код валюти", DB_LONG)
ERwinField.Required = True
ERwinTableDef.Fields.Append ERwinField
Set ERwinField = ERwinTableDef.CreateField("Курс валюти", DB_DOUBLE)
ERwinField.Required = True
ERwinTableDef.Fields.Append ERwinField
Set ERwinField = ERwinTableDef.CreateField("Сума валюти", DB_DOUBLE)
ERwinField.Required = True
ERwinTableDef.Fields.Append ERwinField
Set ERwinField = ERwinTableDef.CreateField("Дата продажу", DB_DATETIME)
ERwinField.Required = True
ERwinTableDef.Fields.Append ERwinField
Set ERwinField = ERwinTableDef.CreateField("Код пункту", DB_LONG)
ERwinField.Required = True
ERwinTableDef.Fields.Append ERwinField
ERwinDatabase.TableDefs.Append ERwinTableDef
Set ERwinField = ERwinTableDef.Fields("Код продажу")
Call SetFieldProp(ERwinField, "Caption", DB_TEXT , "Код продажу:")
Set ERwinField = ERwinTableDef.Fields("Код валюти")
Call SetFieldProp(ERwinField, "Caption", DB_TEXT , "Код валюти:")
Set ERwinField = ERwinTableDef.Fields("Курс валюти")
Call SetFieldProp(ERwinField, "Caption", DB_TEXT , "Курс валюти:")
Set ERwinField = ERwinTableDef.Fields("Сума валюти")
Call SetFieldProp(ERwinField, "Caption", DB_TEXT , "Сума валюти:")
Set ERwinField = ERwinTableDef.Fields("Дата продажу")
Call SetFieldProp(ERwinField, "Caption", DB_TEXT , "Дата продажу:")
Set ERwinField = ERwinTableDef.Fields("Код пункту")
Call SetFieldProp(ERwinField, "Caption", DB_TEXT , "Код пункту:")
' CREATE INDEX ХПКПродаж
Set ERwinTableDef = ERwinDatabase.TableDefs("Продаж")
Set ERwinIndex = ERwinTableDef.CreateIndex("ХПКПродаж")
Set ERwinField = ERwinIndex.CreateField("Код продажу")
ERwinIndex.Fields.Append ERwinField
ERwinIndex.Primary = True
ERwinTableDef.Indexes.Append ERwinIndex
' CREATE TABLE Пункт обміну валют
Set ERwinTableDef = ERwinDatabase.CreateTableDef("Пункт обміну валют")

```

Продовження Додатку Б

```

Set ERwinField = ERwinTableDef.CreateField("Код пункту", DB_LONG)
ERwinField.Required = True
ERwinTableDef.Fields.Append ERwinField
Set ERwinField = ERwinTableDef.CreateField("Назва пункту", DB_TEXT, 50)
ERwinField.Required = True
ERwinTableDef.Fields.Append ERwinField
Set ERwinField = ERwinTableDef.CreateField("Адреса", DB_TEXT, 50)
ERwinField.Required = True
ERwinTableDef.Fields.Append ERwinField
Set ERwinField = ERwinTableDef.CreateField("Код клієнта", DB_LONG)
ERwinField.Required = True
ERwinTableDef.Fields.Append ERwinField
Set ERwinField = ERwinTableDef.CreateField("Код касира", DB_LONG)
ERwinField.Required = True
ERwinTableDef.Fields.Append ERwinField
ERwinDatabase.TableDefs.Append ERwinTableDef
Set ERwinField = ERwinTableDef.Fields("Код пункту")
Call SetFieldProp(ERwinField, "Caption", DB_TEXT, "Код пункту:")
Set ERwinField = ERwinTableDef.Fields("Назва пункту")
Call SetFieldProp(ERwinField, "Caption", DB_TEXT, "Назва пункту:")
Set ERwinField = ERwinTableDef.Fields("Адреса")
Call SetFieldProp(ERwinField, "Caption", DB_TEXT, "Адреса:")
Set ERwinField = ERwinTableDef.Fields("Код клієнта")
Call SetFieldProp(ERwinField, "Caption", DB_TEXT, "Код клієнта:")
Set ERwinField = ERwinTableDef.Fields("Код касира")
Call SetFieldProp(ERwinField, "Caption", DB_TEXT, "Код касира:")
' CREATE INDEX ХРКПункт обміну валют
Set ERwinTableDef = ERwinDatabase.TableDefs("Пункт обміну валют")
Set ERwinIndex = ERwinTableDef.CreateIndex("ХРКПункт обміну валют")
Set ERwinField = ERwinIndex.CreateField("Код пункту")
ERwinIndex.Fields.Append ERwinField
ERwinIndex.Primary = True
ERwinTableDef.Indexes.Append ERwinIndex
' CREATE RELATIONSHIP R/8
Set ERwinRelation = ERwinDatabase.CreateRelation("R/8", "Пункт обміну валют", "Купівля")
Set ERwinField = ERwinRelation.CreateField("Код пункту")
ERwinField.ForeignName = Код пункту
ERwinRelation.Fields.Append ERwinField
ERwinDatabase.Relations.Append ERwinRelation
' CREATE RELATIONSHIP R/16
Set ERwinRelation = ERwinDatabase.CreateRelation("R/16", "Валюта", "Купівля")
Set ERwinField = ERwinRelation.CreateField("Код валюти")
ERwinField.ForeignName = Код валюти
ERwinRelation.Fields.Append ERwinField
ERwinDatabase.Relations.Append ERwinRelation
' CREATE RELATIONSHIP R/11
Set ERwinRelation = ERwinDatabase.CreateRelation("R/11", "Пункт обміну валют", "Продаж")
Set ERwinField = ERwinRelation.CreateField("Код пункту")
ERwinField.ForeignName = Код пункту
ERwinRelation.Fields.Append ERwinField
ERwinDatabase.Relations.Append ERwinRelation
' CREATE RELATIONSHIP R/14
Set ERwinRelation = ERwinDatabase.CreateRelation("R/14", "Валюта", "Продаж")

```

Продовження Додатку Б

```
Set ERwinField = ERwinRelation.CreateField(      "Код валюти" )
ERwinField.ForeignName = Код валюти
ERwinRelation.Fields.Append ERwinField
ERwinDatabase.Relations.Append ERwinRelation
' CREATE RELATIONSHIP R/7
Set ERwinRelation = ERwinDatabase.CreateRelation("R/7", "Клієнт", "Пункт
обміну валют")
Set ERwinField = ERwinRelation.CreateField(      "Код клієнта" )
ERwinField.ForeignName = Код клієнта
ERwinRelation.Fields.Append ERwinField
ERwinDatabase.Relations.Append ERwinRelation
' CREATE RELATIONSHIP R/15
Set ERwinRelation = ERwinDatabase.CreateRelation("R/15", "Касир", "Пункт
обміну валют")
Set ERwinField = ERwinRelation.CreateField(      "Код касира" )
ERwinField.ForeignName = Код касира
ERwinRelation.Fields.Append ERwinField
ERwinDatabase.Relations.Append ERwinRelation
ERwinDatabase.Close
ERwinWorkspace.Close
' Terminating Access Basic DAO Session...
```

Фрагмент програмного коду реалізованої програми

```
namespace Converter
{
    public partial class Form1 : Form
    {
        string from, to, dzherelo, datakurs;

        public Form1()
        {
            InitializeComponent();
            listBox1.SelectedIndex = 4;
            listBox2.SelectedIndex = 1;
        }

        private void button1_Click(object sender, EventArgs e)
        {
            from = listBox1.SelectedItem.ToString(); // обрана валюта у 1
            to = listBox2.SelectedItem.ToString(); // обрана валюта у 2
            dzherelo = comboBox1.SelectedItem.ToString();
            datakurs = dateTimePicker1.Value.ToString("dd-MM-yyyy");

            if (from == to)
            {
                MessageBox.Show("Це та сама валюта!", "Увага!");
            }
            else if (textBox1.Text == "")
            {
                MessageBox.Show("Введіть кількість валют!", "Увага!");
            }
            else
            {
                webBrowser1.Navigate("https://minfin.com.ua/ua/currency/converter/" +
                    textBox1.Text + "-" + from + "-to-" + to + "?converter-type=" + dzherelo +
                    "&date=" + datakurs);
            }
        }
    }
}
```