

Державний вищий навчальний заклад
“Українська академія банківської справи
Національного банку України”

“ДОПУЩЕНА ДО ЗАХИСТУ”
Завідувач кафедри
економічної кібернетики

(підпис)
“ ____ ” _____ 2011 р.

ДИПЛОМНА РОБОТА

на здобуття освітньо-кваліфікаційного рівня магістра

Тема роботи: “Автоматизація внутрішнього аудиту кредитних операцій
комерційного банку”

Виконав студент 5 курсу, група МЕК-01, спеціальність 8.03050201 «Економічна
кібернетика

Тимошенко Євген Володимирович

Керівник дипломної роботи:
доц. Яровенко Г.М.

(підпис)
“ ____ ” _____ 2011 р.

Виконавець-випускник:
Тимошенко Є.В.

(підпис)
“ ____ ” _____ 2011 р.

ЗМІСТ

ВСТУП.....	3
1 ОБСТЕЖЕННЯ СТАНУ АВТОМАТИЗАЦІЇ ВНУТРІШНЬОГО АУДИТУ В КОМЕРЦІЙНИХ БАНКАХ.....	5
1.1 Теоретичні основи організації внутрішнього аудиту кредитних операцій в банках.....	5
1.2 Аналіз стану автоматизації внутрішнього аудиту в сучасних банківських інформаційних системах.....	11
1.3 Функції, що потребують автоматизації, та формування вимог до системи.....	22
2 РОЗРОБКА ПРОЕКТУ АВТОМАТИЗАЦІЇ ВНУТРІШНЬОГО АУДИТУ КРЕДИТНИХ ОПЕРАЦІЙ КОМЕРЦІЙНОГО БАНКУ.....	25
2.1 Функціональний склад корпоративної інформаційної системи в комерційному банку.....	25
2.2 Місце аудиту кредитних операцій в корпоративній інформаційній системі комерційного банку.....	30
2.3 Архітектура та технології створення автоматизованої системи внутрішнього аудиту в банках.....	34
2.4 Підсистеми забезпечення функціональної частини	39
3 РЕАЛІЗАЦІЯ ПРОГРАМНОГО РІШЕННЯ ВНУТРІШНЬОГО АУДИТУ КРЕДИТНИХ ОПЕРАЦІЙ КОМЕРЦІЙНОГО БАНКУ.....	46
3.1 Вхідні та вихідні дані системи.....	46
3.2 Структура та особливості реалізації інформаційного та алгоритмічного забезпечення.....	49
3.3 Інтерфейс користувача та інструкція по використанню.....	60
3.4 Оцінка ефективності від впровадження запропонованого програмного рішення.....	77
ВИСНОВКИ.....	80
ПЕРЕЛІК ПОСИЛАНЬ.....	82
ДОДАТКИ.....	85

ВСТУП

Аудит як вид діяльності і сегмент ринку в нашій країні ще зовсім молодий, особливо в порівнянні з бухгалтерським обліком. Однак, в умовах збільшення числа компаній, їх укрупнення, розширення спектру наданих ними послуг, частих змін законодавчої та нормативної бази, а також майже суцільної автоматизації бухгалтерського і податкового обліку, виникла необхідність автоматизації також і аудиторських процедур. При цьому в аудиторській діяльності досі відсутні чітко модифіковані автоматизовані технології, що негативно впливає на якість аудиторських послуг, оскільки аудитор повинен за регламентований час проаналізувати значний обсяг інформації та сформулювати на її основі точні висновки.

Причинами затримки розвитку ринку інформаційних технологій аудиторської діяльності є такі:

- низький рівень розвитку аудиторського ринку, викликаний початковим періодом його становлення;
- недостатній рівень комп'ютерної грамотності користувачів;
- специфіка аудиторської діяльності, що полягає у неможливості повної формалізації процесу аудиту з-за значної кількості логічних суджень.

У сфері банківської діяльності також спостерігається необхідність автоматизації аудиту в цілому. Зокрема, у конкурентній боротьбі важливим засобом є внутрішній аудит кредитних операцій, який також потребує автоматизації. Внутрішній аудит кредитних операцій допомагає утриматися банку на ринку, зменшити ризики несплати кредитів та покращити власний кредитний портфель. В банках з цією метою створюються відділи, що розробляють стратегії внутрішнього аудиту, формують і координують виконання планів внутрішнього аудиту.

Тому, метою даної дипломної роботи є розробка програмного рішення для внутрішнього аудиту кредитних операцій комерційного банку, яке дозволяє вирішити проблему ефективного управління кредитною діяльністю комерційних банків-філій, виявлення неефективних операцій, пов'язаних з незаконною видачею кредитів.

Об'єктом дипломної роботи виступає кредитна діяльність комерційного банку.

Предметом дипломної роботи є внутрішній аудит кредитних операцій та засоби автоматизації його процедур.

Об'єктом впровадження автоматизованої системи виступають головні відділення банків, що здійснюють внутрішній аудит.

Задачами дипломної роботи є:

- аналіз стану автоматизації внутрішньобанківського аудиту в Україні;
- визначення переліку функцій, що потребують автоматизації;
- визначення переліку вимог до автоматизованої системи;
- вибір архітектури та технології вирішення задач;
- визначення складу функціональної частини;
- визначення складу підсистем забезпечення функціональної частини, таких як програмне забезпечення, апаратне забезпечення та інші види забезпечення;
- визначення переліку вхідних та вихідних даних систем;
- визначення структури та особливості реалізації інформаційного та алгоритмічного забезпечення;
- розроблення інтерфейсу користувача;
- розроблення інструкції по використанню автоматизованої системи;
- визначення очікуваних ефектів від впровадження автоматизованої системи.

Інформаційно-фактологічну базу проведеного дослідження сформували законодавчі та нормативні документи з питань діяльності фінансових посередників та розвитку фінансового ринку; нормативні документи Національного банку України; нормативи та стандарти ведення аудиту у банках; кредитні договори; закони, положення та інструкції стосовно здійснення кредитних операцій; методики аудиту кредитних операцій в комерційному банку.

В процесі виконання даної курсової роботи були використані методи наукового дослідження такі як: аналіз і синтез, індукція і дедукція (для опрацювання фактичного і теоретичного матеріалу), також були використані й інші: економіко-математичний метод, системний аналіз та метод дослідження документів.

1 ОБСТЕЖЕННЯ СТАНУ АВТОМАТИЗАЦІЇ ВНУТРІШНЬОГО АУДИТУ В КОМЕРЦІЙНИХ БАНКАХ

1.1 Теоретичні основи організації внутрішнього аудиту кредитних операцій в банках

На сьогодні одним з найбільш складних і суперечливих питань українського аудиту є аудит кредитних операцій, що пов'язано з існуванням проблеми неплатежів. Суб'єкти господарювання на перший план висувають вирішення власних проблем, замість виконання фінансових зобов'язань по платежах перед банками.

Дослідженню даної проблеми присвячені роботи таких вітчизняних вчених, таких як Пантелеєв В.П., Корінько М.Д., Сарахман О.М., Івахненко С.В., Подольский В.И., Щербакова Н.С., Комисаров В.Л., але незважаючи на величезний внесок науковців у дослідження, низка нерозв'язаних проблем залишається.

Таким чином, в умовах формування і становлення соціально-економічних відносин актуальним є питання автоматизація внутрішнього аудиту в цілому та внутрішнього аудиту кредитних операцій зокрема, а також постійне вдосконалення структури внутрішнього аудиту.

Внутрішній аудит створюється в банку для перевірки контрольно-орієнтованих заходів і процедур. Щоб забезпечити ефективність внутрішнього аудиту, треба гарантувати незалежність аудиторських працівників шляхом їх прямого підпорядкування спостережній раді чи аудиторському комітету спостережної ради. Внутрішній аудит не тільки робить суттєвий внесок у зміцнення середовища контролю, а й зменшує витрати банку на зовнішній аудит. Якщо зовнішній аудитор отримує інформацію, яка свідчить про компетентність, чесність та об'єктивність внутрішніх аудиторів, то він може в багатьох ситуаціях покластись на їхню роботу та зменшити обсяг аудиторської

перевірки. Крім того, внутрішні аудитори можуть безпосередньо допомагати зовнішньому аудиторіві у здійсненні аудиторських процедур [5].

Внутрішній аудит – це незалежний підрозділ, який не бере участі у виконанні операцій, а періодично перевіряє та оцінює:

- адекватність і ефективність заходів та системи внутрішнього контролю;
- управління ризиками, кредитним та інвестиційним портфелями;
- повноту, своєчасність і достовірність фінансової та іншої звітності;
- дотримання принципів і внутрішніх процедур обліку;
- відповідність регулятивним вимогам.

В Україні постановою Правління НБУ від 20 березня 1998 р. (№ 114) затверджено «Положення про організацію внутрішнього аудиту в комерційних банках України». З того часу наявність служби внутрішнього аудиту в комерційних банках стала однією з обов'язкових умов отримання банком ліцензії на банківську діяльність.

Внутрішній аудит – це незалежна експертна діяльність аудиторських служб банку для перевірки та оцінки адекватності й ефективності системи внутрішнього контролю та якості виконання співробітниками банку своїх обов'язків [21].

Як видно з визначення, наявність внутрішнього аудиту не скасовує інших форм та функцій контролю. У цій же постанові вказується, що внутрішній аудит передбачає отримання в документальній формі процедур і опису методів, які використовуються банком для здійснення внутрішнього контролю. Таким чином, внутрішній аудит оцінює два аспекти банківської діяльності:

- банківські операції;
- системи внутрішнього контролю у структурних підрозділах банку.

Сутність внутрішнього аудиту виявляється у його функціях, що представлені на рисунку 1.1.

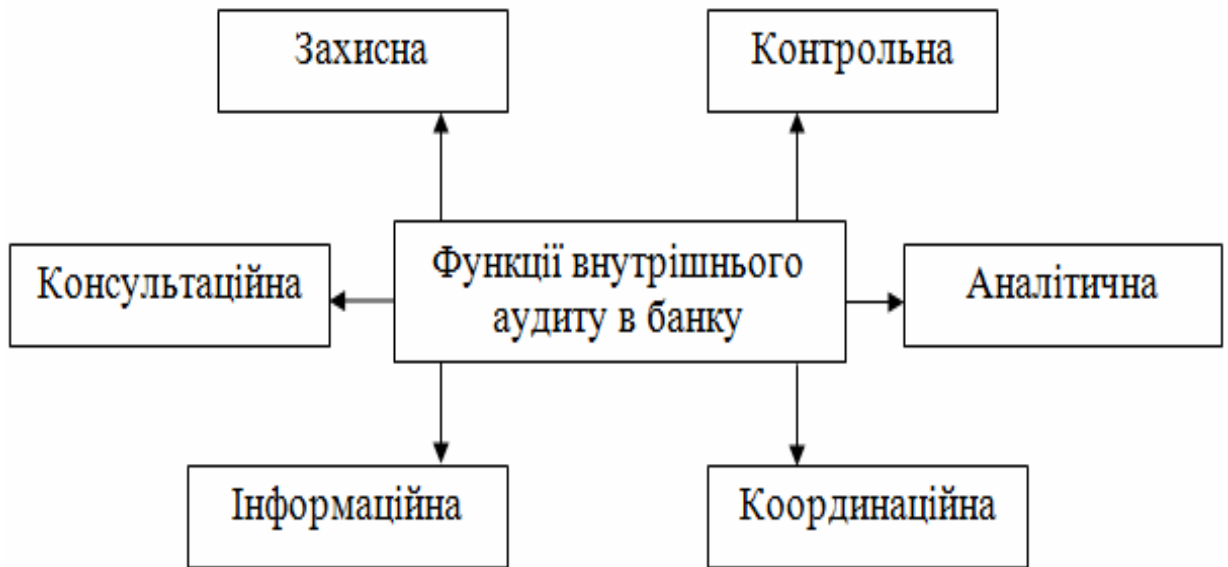


Рисунок 1.1 – Функції внутрішнього аудиту в комерційному банку

Функція координації полягає в плануванні роботи внутрішнього аудитора з погляду потреб акціонерів, інвесторів та правління банку.

Функція контролю – внутрішні аудитори перевіряють роботу окремих підрозділів банку з метою оцінки внутрішнього контролю, політики та процедур банку, оцінюють адекватність систем внутрішнього технологічного контролю та дисципліну виконання персоналом вказівок і наданих рекомендацій.

Захисна функція – це зусилля, спрямовані на охорону активів банку від потенційних крадіжок та фактів шахрайства, неефективного управління банківським бізнесом.

Аналітична функція – це аналіз ефективності та експертна оцінка різних аспектів (операцій) банківської діяльності.

Інформаційна функція – інформування керівництва банку про результати перевірок, виявлені факти порушень вимог законодавства або внутрішніх правил.

Консультаційна функція – надання поточних консультацій працівникам підрозділів.

Служба внутрішнього аудиту проводить перевірки діяльності та систем банку з метою:

- забезпечення дотримання банком вимог чинного законодавства України та нормативних актів НБУ;

- забезпечення повноти, своєчасності та достовірності відображення операцій у первинних документах, бухгалтерських реєстрах та системі бухгалтерського обліку відповідно до внутрішніх інструкцій та процедур банку і згідно з рішеннями керівників та акціонерів банку;

- виявлення недоліків у системах внутрішнього контролю та подання рекомендацій щодо шляхів їх подолання;

- виявлення сфер потенційних збитків для банку;

- виявлення умов, сприятливих для шахрайства, зловживань та незаконного привласнення коштів банку.

Служба внутрішнього аудиту банку – це самостійний структурний підрозділ банку, який створюється за рішенням виконавчого органу банку і підпорядковується безпосередньо спостережній раді банку. Це підпорядкування зменшує можливість тиску на службу внутрішнього аудиту з боку зацікавлених осіб та підвищує прозорість отриманих результатів перевірки та зроблених аудиторських висновків [11]. Дане підпорядкування представлено на рисунку 1.2.

Перевірки, що їх проводить у банку служба внутрішнього аудиту банку, здійснюються на підставі «Положення про службу внутрішнього аудиту банку» та відповідних регламентів, які кожний банк розробляє самостійно з дотриманням вимог Національного банку України, міжнародного досвіду та специфіки діяльності самого банку. «Положення про службу внутрішнього аудиту банку» затверджує спостережна рада.

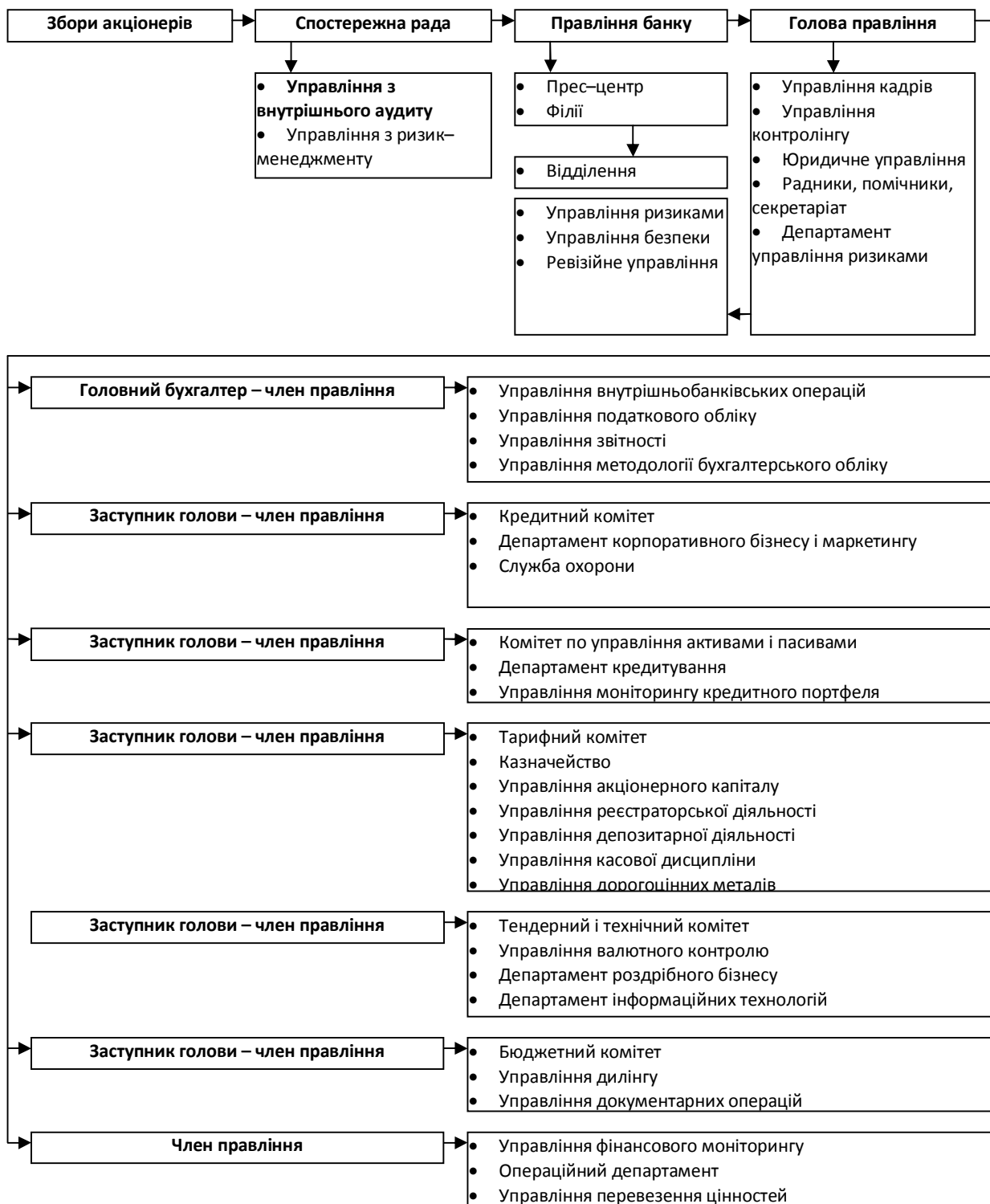


Рисунок 1.2 – Організаційна структура комерційного банку

В обов'язки спостережної ради банку входить:

- затвердження річного плану аудиторських перевірок;
- отримання звітів та висновків значних аудиторських перевірок;

– оцінка рекомендацій внутрішніх аудиторів щодо вдосконалення діяльності та процедур, де це виявилось необхідним;

– отримання річного звіту внутрішнього аудиту, в якому висвітлений загальний стан систем внутрішнього контролю банку, наводяться найважливіші аудиторські висновки та рекомендації. Цей звіт має також включати інформацію про подальший розвиток ситуацій, висвітлених у попередніх звітах.

Відповідно до Положення про службу внутрішнього аудиту банку аудиторська перевірка складається із стандартних етапів, які було проаналізовано на можливість застосування засобів автоматизації (рис. 1.3).

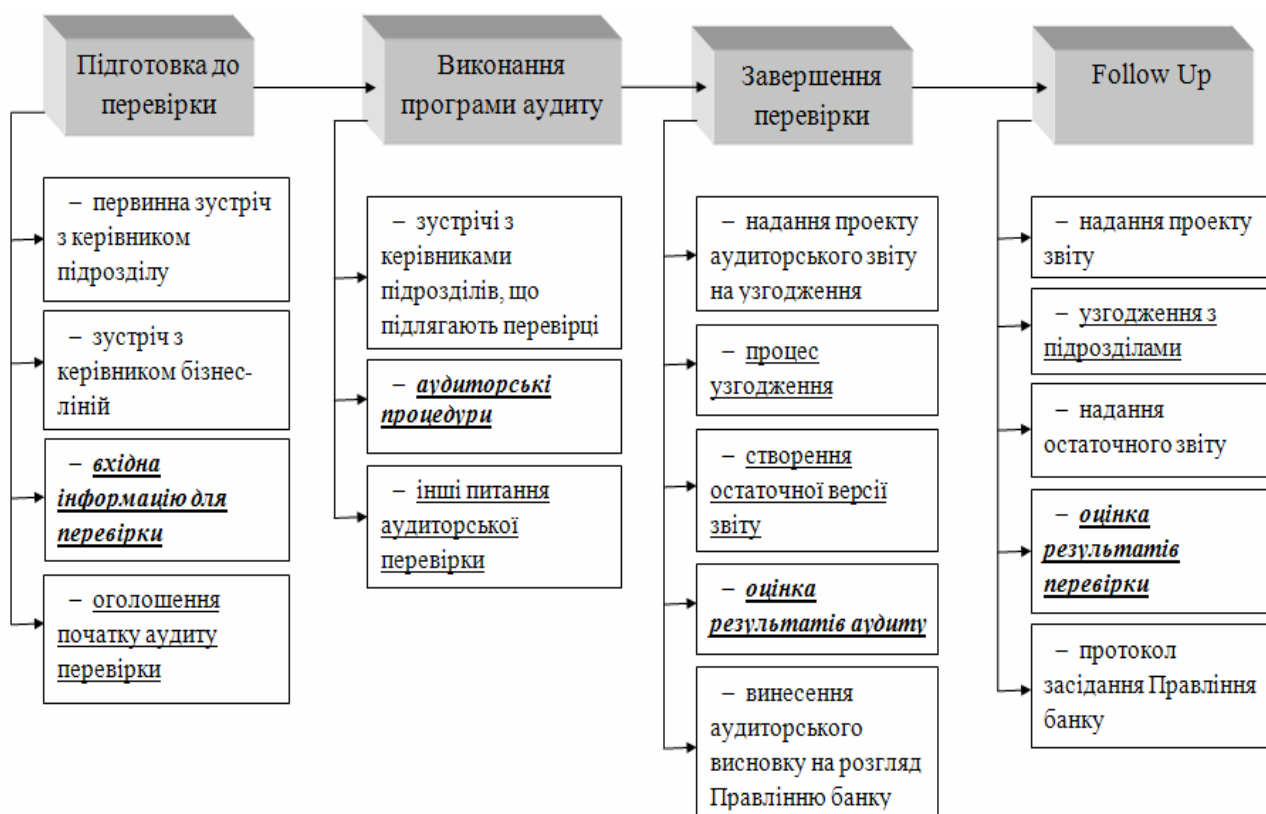


Рисунок 1.3 – Етапи внутрішнього аудиту в комерційному банку

На рисунку 1.3 представлені етапи, що є на даний момент часу автоматизовані (підкреслений текст), ті, що автоматизувати не можливо та не доцільно (звичайний текст) і ті, що потрібно автоматизувати (підкреслений курсивний текст).

Одним із трудомістких етапів проведення внутрішнього аудиту в банку є «Аудиторські процедури», на який аудитор витрачає найбільшу частину часу, що виділяється на перевірку, тому саме цьому етапу потрібно приділяти найбільшу увагу.

Таким чином, аудиторам потрібно зібрати, згрупувати, проаналізувати, перевірити та оцінити великий обсяг інформації у стислі строки, а також постійно додержуватись вимог нормативних документів, стежити за їх оновленням.

Використання комп'ютеризованих методів аудиту дозволяє значно скоротити витрати часу на виконання аудиторських завдань. Проте не можна стверджувати, що сьогодні робота аудитора автоматизована та комп'ютеризована на належному рівні. Тому, на наш погляд, ці аспекти проведення аудиту в комп'ютерному середовищі нині є актуальними.

1.2 Аналіз стану автоматизації внутрішнього аудиту в сучасних банківських інформаційних системах

Впровадження електронних систем обробки й передачі інформації здобувають універсальний і загальний характер, охопивши всі напрямки банківської діяльності. Сучасні інформаційні технології дозволяють координувати діяльність підрозділів банків, розширити міжбанківські зв'язки, діяти на фінансових ринках ряду країн. Нові можливості автоматизації банківських операцій робочих місць фахівців, інформаційних технологій банківських послуг дозволяють комплексно вирішувати проблеми аналізу банківської діяльності, розробки й створення регіональних, міжрегіональних і міжнародних банківських систем [18].

Автоматизація інформаційних й інших технологій банку сприяє рішенням стратегічних завдань. Головними напрямками автоматизації банків є зменшення витрат і збільшення якості обслуговування.

Рішення аналітичних завдань банківської діяльності диктується необхідністю визначення фінансових результатів, прогнозування напрямку розвитку, оцінкою економічної обґрунтованості й доцільності діяльності кожного банку. У вітчизняній практиці автоматизація аналітичних розрахунків втілюється поки що в окремих програмних продуктах, які ще не отримали широкого розповсюдження.

Питанню автоматизації банківської діяльності в сучасній науковій літературі приділяється багато уваги. Теоретичні аспекти розробки комплексних автоматизованих рішень для організації представлені в багатьох літературних джерелах, основні з яких представлені у таблиці 1.1.

Таблиця 1.1 – Літературні джерела по заданій тематиці

Місце аудиту	Автор	Назва праці	Мета написання
Аудит на підприємствах	Чемберс Е. [28]	«Комп'ютерний аудит»	Проведення аудиту в комп'ютерному середовищі
	Івахненко С.В. [9]	«Автоматизація аудиту в Україні та світі: підходи і програмне забезпечення»	Аналіз існуючого програмного забезпечення для проведення аудиту
		«Аудиторське програмне забезпечення для аналізу даних: перший український досвід»	Розробка програмного забезпечення для проведення аудиту в Україні, тенденції розвитку

Продовження таблиці 1.1

Місце аудиту	Автор	Назва праці	Мета написання
Аудит на підприємствах	Савченко В.Я. [1]	«Організація і методика аудиту»	Висвітлення процесу проведення аудиту та існуючих методик
	Кушнарєв А.С. [18]	«Проблеми автоматизації аудита»	Висвітлення існуючих проблем аудиту та шляхи їх подолання
	Давидов Г.М. [8]	«Аудит»	Теоретичні основи проведення аудиту
	Шевчук В.О. [5]	«Внутрішній аудит»	Проведення внутрішнього аудиту на підприємствах
Аудит у банках	О.М. Сарахман [2]	Аудит банків	Проведення контролю та аудиту у банках
Внутрішній аудит кредитних операцій у банках	Кіреєв О. І., Любунь О.С., Кравець В.М. [11]	Внутрішній аудит у банках	Проведення внутрішнього аудиту у банках

Таким чином, аналізуючи літературні джерела, що представлені в таблиці 1.1, можна сказати, що більшість літератури присвячена саме аудиту на підприємствах: теоретичні основи та підходи до автоматизації; менша увага приділяється аудиту в банках, а література про автоматизацію внутрішньобанківського аудиту кредитних операцій або взагалі відсутня, або представлена у вигляді наукових статей на задану тематику.

Щодо конкретної тематики праць, більшість авторів торкаються питань автоматизації проведення банківських операцій, дають різноманітні характеристики систем «Операційний день банку», систем міжбанківських розрахунків, систем електронних платежів. Серед проблем автоматизації банківських операцій найбільшої популярності набули такі види діяльності [26]:

- автоматизація розрахункових і касових операцій;
- автоматизація кредитних та депозитних операцій;
- автоматизація управління кредитними ресурсами;
- автоматизація та регулювання валютних операцій;
- інформаційні технології підтримання банківських операцій на фондовому ринку;
- автоматизація формування звітності комерційного банку.

Багато уваги приділяється електронній комерції, розвитку корпоративних банківських мереж.

В деяких наукових публікаціях вивчається питання автоматизації аналітичної роботи в комерційному банку, аналіз показників звітності, питання управлінського обліку [26].

Щодо банківських установ, то дане питання розглядається в рамках автоматизації аудиту документів у банку. На практиці аудит документів проводиться вручну з частковим використанням програмного забезпечення, в якості програмного забезпечення виступає звичайний текстовий редактор. Тому методами для реалізації технології аудиту документів, які повинні бути автоматизовані є:

- звірка документів всередині однієї бази даних;
- звірка документів з одного робочого місця по різних базах даних одночасно;
- звірка документів поза системою з «викачуванням» інформації в окрему базу (з їх порівнянням);
- вибіркова звірка по заданих параметрах;
- перехресна звірка з подальшим порівнянням результатів.

Якщо ж говорити про автоматизацію хоча б одного з етапів внутрішнього банківського аудиту, то під час проведення аналізу літературних джерел на задану тематику, було знайдено досить мала кількість публікацій. На основі чого можна зробити висновок про існування проблеми автоматизації внутрішнього аудиту в комерційному банку, його новизну і в той же час, проблему з точки зору відсутності наукових розробок та методик.

Основна маса автоматизованих банківських систем (АБС), що пропонуються на вітчизняному ринку, за вартістю й вимогами до обчислювальних засобів розрахована на середні й невеликі банки. Розширення систем відбувається, як правило, у тому випадку, якщо відповідні запити споживача носять потенційно масовий характер. Тому з ростом фінансових можливостей банків зростає й попит на багатоплатформенні системи. Майже всі фірми – розробники мають нові версії АБС, що узагальнюють попередній досвід, орієнтовані на ті ж програмно-активні засоби, але з поліпшеною архітектурою й більшим спектром можливостей. На ринку АБС немає єдиного магістрального напрямку їхнього розвитку, а поява нових класів систем значною мірою визначається інтенсивністю й особливостями розвитку банківської справи в країні.

Етапи проведення внутрішнього аудиту повно або частково можуть бути автоматизовані за допомогою стандартного набору програмного забезпечення комерційного банку. Основою застосування методик аудиту у більшості випадків є звичайні аналітичні пакети. Вони використовують інформацію з загальної бази даних корпоративної системи та стандартних баз фінансових даних. За допомогою програм-конверторів інформація потрапляє до аналітичного пакету. Надалі, в залежності від функціональності пакету або створюється аудиторський висновок, або видається певна інформація для його формування без використання автоматизованої системи. Автоматизація етапу встановлення адекватності, ефективності та своєчасності проведення рекомендованих заходів проводиться за допомогою моніторингу даних корпоративної бази даних. Автоматизація ж етапу документування аудиторської перевірки можлива за наявності програмних продуктів автоматизації документообігу.

Служба внутрішнього аудиту комерційного банку вже давно стала центральною ланкою банківського менеджменту. Діяльність відділів даної служби будується по спеціальній програмі конфіденційного характеру, затвердженій Правлінням банку. Із другорядного, залежного підрозділу, внутрішній аудит вже давно перетворився в одну із провідних структурних одиниць банку.

Одним з найбільш актуальних завдань, які аудитори ставлять перед розробниками програмного забезпечення в цей час, є автоматизація діяльності служби внутрішнього аудиту на всіх етапах аудиторської перевірки: від підготовки й планування до збору, систематизації й оформлення підсумкових документів відповідно до діючих стандартів. У той же час, сьогодення функціонування банківських установ відрізняється численністю й складністю фінансових операцій, тому для аудиторів стали практично незамінними комп'ютерні методи тестування числових даних у ході аудиторських процедур. Останні дозволяють виявити важливі тенденції в діяльності банку, виявити незвичайні відхилення й диспропорції показників (якщо вони мають місце). Застосування комп'ютерної обробки дає можливість підвищити надійність висновків і рентабельність аудиту.

Навряд чи знайдеться хоча б один банк, який сьогодні має комплексне рішення, якщо під цим розуміти рішення на базі єдиного програмного забезпечення (можливо, одного розробника). До цього потрібно прагнути, хоча не завжди. Так як для вузькоспеціалізованих задач, що мають свої особливості, існують кращі в своєму класі рішення. Кожен банк сам для себе повинен вирішувати, що йому краще використовувати.

В останній час все більше українських банків звертаються до західних рішень щодо банківської автоматизації. Ряд західних фірм пропонують свої рішення в Україні, й великі системні банки розглядають такі рішення. Деякі з них перейшли на стадію підписання договорів й реалізації. Загальновідомо, що в цьому напрямі працюють банки «Райффайзен банк Аваль», «Укрсоцбанк» та декілька інших.

При цьому потрібно мати на увазі, що придбання закордонного продукту потребує проведення внутрішнього реінжинірингу, відповідних змін в роботі. Вкладені в такого роду продукти кошти повинні окупити себе.

Група пакетів прикладних програм загального і проблемно-орієнтованого призначення включає широкий набір готових універсальних програм, з якими аудитор повинен вміти працювати. За функціональним призначенням і застосуванням їх в аудиті було виділено такі підгрупи, що представлені таблиці 1.2.

Таблиця 1.2 – Прикладні програми, які використовуються в аудиті

Тип програмного забезпечення	Приклад	Опис
Текстові редактори	Open Office Word, Microsoft Office Word	Призначені для створення та опрацювання текстів, з якими працює аудитор, наприклад створення аудиторського висновку.
Табличні редактори	Open Office Excel, Microsoft Office Excel	Призначені для створення і обробки документів з автоматичним розрахунком даних, що вносяться.
Правові бази даних і довідники	«Юрист+Закон», «ЛІГА»	Доступ до нормативно-правової бази у сфері банківської справи, методики аудиту, шаблони робочих документів.
Спеціалізовані статистичні та математичні пакети загального призначення	Statistica, MathCAD	Призначені для обробки статистичних даних, проведення аналізу даних.

Продовження таблиці 1.2

Тип програмного забезпечення	Приклад	Опис
Бухгалтерські програми та окремі модулі КІС	1С:Бухгалтерія, Турбо-Бухгалтер, Дебет-Кредит, Інфософт, Галактика, Парус	Призначені для ведення всіх видів бухгалтерського обліку
Програми електронного документообігу	Documentum 5	Призначені для роботи з документами, веб-сторінками, графічними зображеннями, кресленнями.
Експертні системи	DEDUCTOR studio, Prime Decision	Призначені для проведення вибору рішень у певних неструктурованих і слабкоструктурованих задачах, у тому числі й тих, що мають багато критеріїв.
	CLIPS, Dendral, R1 (експертна система)/XCon	Призначені для забезпечення вирішення специфічних завдань без присутності.
Програми автоматизації аудиторської діяльності	«ЕкспресАудит: ПРОФ», AuditXP, «Комплекс Аудит», Audit System/2tm, Audit Expert, «Аудит-майстер»	Підготовка, планування, здійснення аудиту та формування висновку.
	«ІТ Audit: Аудитор 2.0» і AuditNET	Комплексна автоматизація всіх процесів аудиторської діяльності.

Проведемо порівняльний аналіз програмного забезпечення за функціональними можливостями, що представлені в останній строчці таблиці 1.1, а саме «Програми автоматизації аудиторської діяльності». Порівняльний аналіз будемо проводити у програмному додатку «PRIME Decision», який дозволяє здійснювати вибір серед альтернатив.

Створення моделі починається з визначення головної мети, інших цілей і атрибутів. Побудова PRIME-моделі здійснюється у вікні Value Tree (Дерево значень), яке представлено на рисунку 1.4.

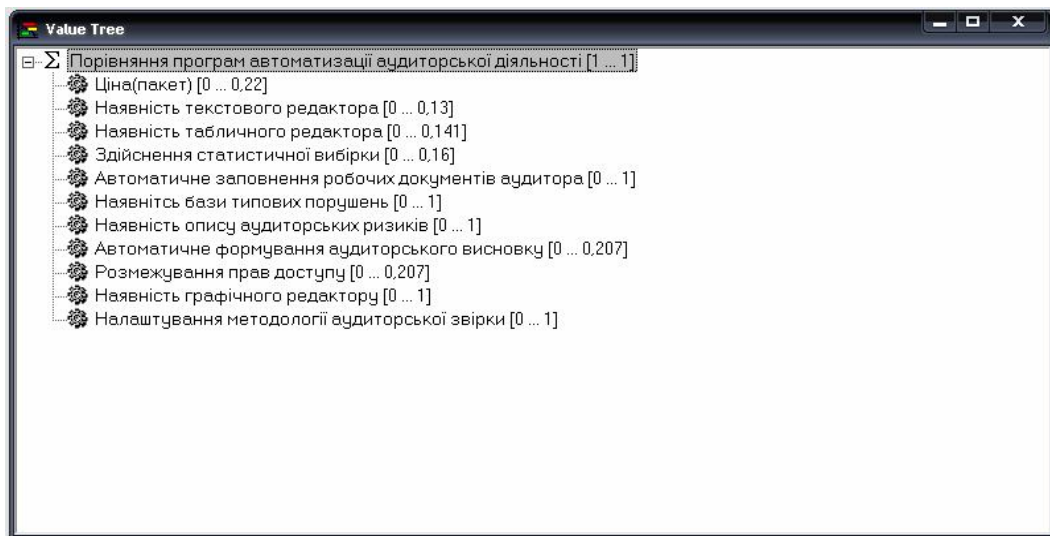


Рисунок 1.4 – Вікно Value Tree (Дерево значень)

У наведеному вище рисунку представлена мета створення моделі, а саме: порівняння програм автоматизації аудиторської діяльності. А також функціональні характеристики програмного забезпечення у аудиті, такі як:

- ціна;
- наявність текстового редактора;
- наявність табличного редактора;
- здійснення статистичної вибірки;
- автоматичне заповнення робочих документів аудитора;
- наявність бази типових порушень;
- наявність опису аудиторських ризиків;
- автоматичне формування аудиторського висновку;
- розмежування прав доступу;

- наявність графічного редактору;
- налаштування методології аудиторської звірки.

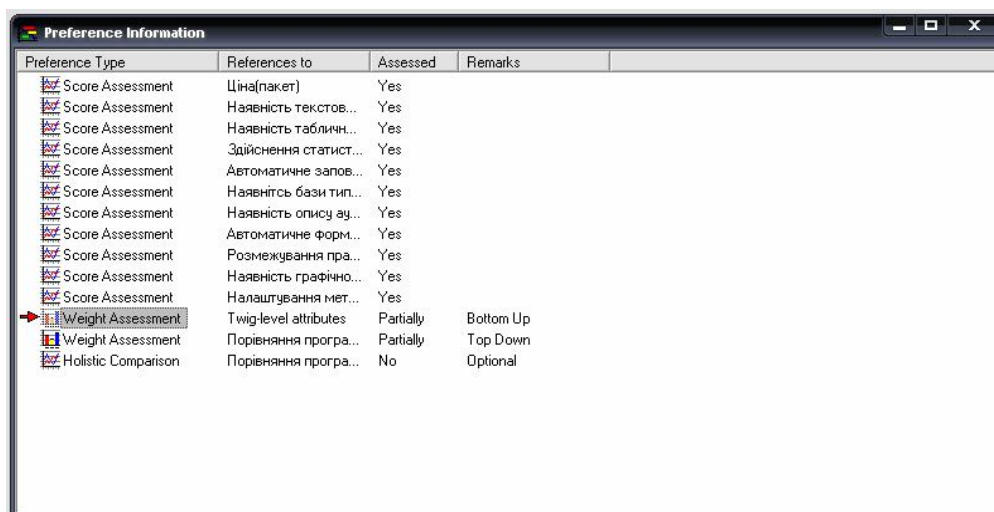
Після того, як було створено дерево значень, необхідно визначити альтернативи. Альтернативи визначаються у вікні Alternative, яке називається альтернативна матриця або сітка. Альтернативна матриця представлена на рисунку 1.5.



Name	Ціна(пакет)	Наявність текстов	Наявність табличн	Здійснення статист	Автоматичне запов	Наявність бази тип	Наявність опису ау	Автоматичне форм	Розмежування пра	Наявність графічн	Налаштування мет
Експрес Аудит. ПРОФ	1500	Так	Так	Так	Ні	Ні	Ні	Так	Так	Ні	Ні
IT Audit: Аудитор 2.0	20000	Так	Так	Так	Так	Так	Так	Так	Так	Так	Так
Аудит ХР	1200	Так	Так	Ні	Ні	Ні	Ні	Ні	Так	Ні	Ні
Комплекс аудит	900	Так	Ні	Ні	Ні	Так	Ні	Так	Ні	Ні	Ні
Audit System/2tm	2000	Так	Так	Так	Так	Ні	Ні	Так	Так	Ні	Ні
Аудит-майстер	300	Ні	Так	Так	Ні	Ні	Ні	Так	Так	Ні	Ні
Audit Net	11200	Так	Так	Так	Так	Так	Так	Так	Так	Ні	Так

Рисунок 1.5 – Вікно Alternative (Альтернативна матриця)

Наступний крок – визначення переваги для значень альтернатив. Переваги визначаються у вікні Preference Information. Є три типи елементів оцінки переваг: Score Assessment, Weight Assessment, Holistic Assessment. Вікно визначення переваг представлено на рисунку 1.6.



Preference Type	References to	Assessed	Remarks
Score Assessment	Ціна(пакет)	Yes	
Score Assessment	Наявність текстов...	Yes	
Score Assessment	Наявність табличн...	Yes	
Score Assessment	Здійснення статист...	Yes	
Score Assessment	Автоматичне запов...	Yes	
Score Assessment	Наявність бази тип...	Yes	
Score Assessment	Наявність опису ау...	Yes	
Score Assessment	Автоматичне форм...	Yes	
Score Assessment	Розмежування пра...	Yes	
Score Assessment	Наявність графічн...	Yes	
Score Assessment	Налаштування мет...	Yes	
Weight Assessment	Twig-level attributes	Partially	Bottom Up
Weight Assessment	Порівняння програ...	Partially	Top Down
Holistic Comparison	Порівняння програ...	No	Optional

Рисунок 1.6 – Вікно Preference Information (Визначення переваг)

В наведеному вище вікні визначаємо переваги для кожної функціональної характеристики програмного забезпечення у відповідності до потреб користувача, а також виставляємо ваги для кожної характеристики. Після введення вище наведених вхідних даних проводимо розрахунок створеної моделі і отримуємо результуючу матрицю правил рішення, за допомогою якої і буде визначатися найкращий програмний продукт. Вікно правил рішення представлено на рисунку 1.7.

	Maximax	Maximin	Central Values	Minimax Regret	Possible Loss
Експрес Аудит: ПРОФ	✓				1,000
IT Audit: Аудитор 2.0	✓	✓	✓	✓	0,220
Аудит ХР	✓				1,000
Комплекс аудит	✓				1,000
Audit System/2tm	✓				1,000
Аудит-майстер	✓				1,000
Audit Net	✓				1,000

Рисунок 1.7 – Вікно правил рішення

Таким чином, провівши порівняльну характеристику, можна зробити висновок, що найкращим програмним забезпеченням за функціональним наповненням є «IT Audit: Аудитор 2.0», розробником якого є компанія «IT Audit», але даний програмний продукт є найбільш дорогим серед інших.

Отже, можна зробити висновок, що на ринку програмного забезпечення представлено досить багато програмних комплексів, що можуть бути використані для автоматизації роботи внутрішнього аудитора. Всі вони є досить функціональними але, з огляду на те, що проведення внутрішньобанківського аудиту для кожної банківської установи є справою досить індивідуальною, а також специфіка роботи вимагає проведення внутрішнього аудиту зовсім різних відділів, філій, операцій для кожного банку і з різною періодичністю, можна стверджувати, що можлива поява нових програмних продуктів, які будуть користуватися попитом банківських установ.

Для багатьох інформаційних систем, які використовуються в західних банках, існують спеціальні програми комп'ютерного аудиту. Однак у цей час не відомо про будь-які російські чи українські програми, що є присутніми на ринку.

1.3 Функції, що потребують автоматизації та формування вимог до системи

Мета автоматизації внутрішнього аудиту кредитних операцій – вирішення проблеми ефективного управління кредитною діяльністю комерційних банків, виявлення неефективних операцій, пов'язаних з незаконною видачею кредитів.

Таким чином, для досягнення поставленої мети необхідно здійснити автоматизацію наступних задач аудиту кредитних операцій:

а) перевірка наявності усіх документів щодо оформлення кредитів – перевірка кредитного договору на наявність відповідних документів, що необхідні для отримання кредиту, такими документами можуть бути: довідка з місця постійної роботи за строк не менше 6 місяців, довідка про доходи за строк не менше 6 місяців, довідка про наявність рухомого та нерухомого майна, довідка про сімейний стан та інше; в автоматизованій системі вся ця інформація буде зберігатися у базі даних, як постійна інформація;

б) щомісячне нарахування відсотків – проведення розрахунку відсотку за користування кредитом в цілому за весь період та для щомісячної сплати зокрема, тобто складання графіку платежів на весь період користування кредитом; в автоматизованій системі дана задача вирішується шляхом автоматичного розрахунку відсотків для кожного кредитного договору та складання індивідуального графіку сплати відсотків по кожному кредитному договору;

в) правильність відображення відсотків за джерелами покриття – проводиться перевірка правильності відображення відсотків, тобто правильність розрахунку відсотків, перевірка процентної ставки у відповідності до політики банку з урахуванням джерел покриття; автоматизована система

відповідно до алгоритму проводить перевірку та визначає наявність помилок або їх відсутність;

г) дотримання встановлених правил з оцінювання та оформлення застави при отриманні кредиту – при наявності застави, кредитний інспектор, що надає кредит повинен дотримуватись всіх правил з оцінювання та оформлення застави; автоматизована система здійснює перевірку порядку оцінювання та оформлення застави та відображає порушення, якщо такі є;

д) цільове використання одержаних кредитів – постійний моніторинг цільового використання грошей, що були надані в кредит; в автоматизованій системі зберігається інформація на яке цільове використання були надані кошти, а кредитний інспектор повинен періодично перевіряти цільове використання одержаних кредитів;

е) своєчасність погашення основного боргу – перевірка сплати основного боргу відповідно до індивідуального графіку погашення; автоматизована система постійно контролює погашення та у разі не сплати автоматично з'являється заборгованість та відповідне повідомлення кредитному інспектору;

ж) своєчасність погашення нарахованих відсотків – перевірка сплати відсотків відповідно до індивідуального графіку погашення; автоматизована система постійно контролює погашення та у разі не сплати автоматично з'являється заборгованість та відповідне повідомлення кредитному інспектору;

з) наявність непогашеної заборгованості за позиками банку – у разі наявності непогашеної заборгованості клієнту надається період для погашення у 30 днів, у випадку, якщо клієнт не погасить заборгованість, він повинен буде сплати пеню у розмірі 0.01% за кожен прострочений місяць;

и) наявність та правильність оформлення всіх документів за отриманими кредитами – перевірка правильності оформлення кредитних договорів, тобто заповнення всіх обов'язкових полів, правильність заповнення даних полів; автоматизована система проводить перевірку по кожному кредитному договору та робить висновки у вигляді помилок та відображення полів де були зроблені помилки.

Користувачами автоматизованої системи будуть виступати співробітники відділу внутрішнього аудиту комерційного банку.

Система має можливість надавати користувачу отримувати важливу для роботи інформацію, при необхідності створюючи на її основі детальний звіт.

Проблема автоматизації аудиту ґрунтується на проблемі формалізації цього процесу. Аудит значною мірою є творчим процесом, особливості якого залежать як від конкретного кредитного договору, який перевіряють, так і від практичного досвіду та поглядів самого аудитора.

Тому, на відміну від програм бухгалтерського обліку, вимоги до аудиторського програмного забезпечення мають бути такі:

- розвинуті засоби контролю операцій;
- підвищена гнучкість, оскільки аудиторам доводиться працювати з різними кредитними договорами, кожен з яких має свої параметри, певні особливості та нюанси, то програма повинна мати можливість швидко переналагоджуватись на специфіку кожного клієнта; у комп'ютерних аудиторських системах повинні враховуватись загальні вимоги і стандарти аудиту;
- ергономічність, під якою розуміють наявність засобів для зручного введення великих обсягів інформації, оперативного і простого формування звітів;
- зв'язок на рівні баз даних, який дозволяє реалізувати додаткові засоби для введення і виведення даних, представлених у різних форматах.

Крім основних вимог, аудиторському програмному забезпеченню мають бути властиві такі ознаки:

- простота освоєння;
- швидкодія;
- професійна оперативна підтримка з боку розробників.

2 РОЗРОБКА ПРОЕКТУ АВТОМАТИЗАЦІЇ ВНУТРІШНЬОГО АУДИТУ КРЕДИТНИХ ОПЕРАЦІЙ КОМЕРЦІЙНОГО БАНКУ

2.1 Функціональний склад корпоративної інформаційної системи в комерційному банку

Автоматизована банківська система (далі АБС) повинна забезпечувати:

- автоматизацію внутрішньобанківської діяльності і в першу чергу внутрішньобанківських операцій по обробці платіжних і інших документів у тих підрозділах банківської установи, які працюють безпосередньо з клієнтами;
- автоматизацію виконання міжбанківських розрахунків і інших зовнішньобанківських операцій;
- автоматизацію фінансових операцій в рамках міжнародного банківського бізнесу.

Автоматизація цих процесів повинна доповнювати одна одну і бути взаємозв'язаною і, що автоматизація кожного з них має свою специфіку та особливості і є доволі складною проблемою з високим рівнем автономності.

Зосередимо увагу на автоматизації внутрішньобанківської діяльності на рівні комерційних банків. Огляд АБС показав, що вони відрізняються одна від одної як за структурою, так і за набором функціональних задач. Зараз не має типової структури АБС, якою б керувались банки при розробці своїх систем, практично відсутні елементи стандартизації та уніфікації банківських технологій.

Вивчення структур різних банківських систем та проведене певне їх узагальнення, дозволяє виділити такі основні функціональні підсистеми АБС: операційний день банку (ОДБ), управління кредитними ресурсами (Кредити), управління валютними операціями (Валютні операції), управління депозитами (Депозити), управління цінними паперами (Цінні папери), управління касою (Каса), внутрішньобанківський облік (Внутрішній облік), управління розрахунками з використанням пластикових карток (Карткові операції), звітність, аналіз діяльності банку (Аналіз).

АБС – це технологічна система, яка забезпечує функціонування банківського закладу. Ядром АБС є підсистема ОДБ, яка інформаційно зв'язана з іншими функціональними підсистемами. Крім внутрішніх інформаційних зв'язків АБС характеризується великим спектром інформаційних зв'язків з зовнішнім середовищем, в якості якого виступають клієнти банку, інші банки, фінансові та державні органи. Охарактеризуємо коротко кожен із перерахованих підсистем.

Базовою підсистемою, яка повинна обов'язково функціонувати в кожному комерційному банку є «Операційний день банку» (ОДБ). Основні функції цієї підсистеми такі:

- введення та обробка клієнтських платіжних документів;
- створення та ведення особових рахунків клієнтів та масивів нормативно-довідкової інформації;
- робота з картотеками;
- обробка особових та балансових рахунків;
- ведення аналітичного і синтетичного обліку, формування балансу за кожен банківський день та відповідних вихідних форм.

В рамках підсистеми «Управління кредитними ресурсами банку» працівники кредитного відділу банку мають можливість виконувати такі основні функції:

- аналіз фінансового стану позичальника, визначення його кредитоспроможності та оцінка ризику при кредитуванні;
- формування та облік кредитних договорів;
- ведення та коригування розпоряджень на оплату кредитів;
- ведення та коригування строкових зобов'язань на погашення кредиту;
- ведення та коригування процентних ставок та графіків оплати процентів по кредитному договору;
- нарахування процентів по кредиту та облік їх сплати;
- облік та контроль погашення кредитної заборгованості;
- аналіз кредитного портфеля, класифікація кредитів та визначення розміру резервування.

Підсистема «Управління кредитними ресурсами» повинна бути інтегрована з іншими функціональними підсистемами банку, зокрема з ОДБ, в якій виконують бухгалтерські проведення при наданні кредиту та при погашенні суми основного боргу і відсотків по ньому.

У комерційних банках, що отримали дозвіл на виконання операцій з іноземною валютою має функціонувати підсистема «Управління валютними операціями». У рамках цієї підсистеми обов'язково має функціонувати комплекс задач «Валютний операційний день», що забезпечуватиме увід та обробку валютних платіжних документів, відкриття та закриття валютних рахунків, конвертацію валют, ведення рахунків покриття, формування балансу та інші операції з іноземною валютою.

«Валютний операційний день» — не обов'язково реалізується окремим програмним комплексом. Операції з валютою можуть бути автоматизованими комплексним мультивалютним ОДБ, який може працювати як з національною та і з будь-якою іншою валютою.

Крім комплексу задач «Валютний операційний день» у комерційному банку, можуть бути автоматизовані такі задачі:

- прогнозування курсів валют;
- облік біржових валютних операцій;
- облік ділінгових операцій та оцінка ділінгових контрактів. Для проведення ділінгових операцій банком може використовуватись системи міжнародної фінансової інформації, наприклад система Reuters Dealing;
- робота з системою SWIFT, якщо банк є учасником цієї міжнародної міжбанківської телекомунікаційної мережі передачі банківських повідомлень. Використання SWIFT дозволяє обмінюватись фінансовими повідомленнями з банками нерезидентами та іншими міжнародними фінансовими установами.

Підсистема «Управління депозитами» має забезпечувати автоматизацію робіт по обслуговуванню фізичних та юридичних осіб, що відкрили депозитні (вкладні) рахунки в банку. В підсистемі повинні вестись депозитні (вкладні) рахунки клієнтів банку. Основними задачами підсистеми є: облік операцій з

готівкою, облік безготівкових операцій, облік цінних бланків, нарахування відсотків за депозитними рахунками, а також формування звітних форм для роботи з депозитними вкладками.

Існують два підходи до автоматизації робіт в даній підсистемі. Перший традиційний, коли вкладнику видається паперова ошадна книжка, яка відбиває всі операції по рахунку. Другий напрямок – вкладнику замість ошадної книжки видається пластикова картка з магнітною смужкою, на якій вміщується номер рахунку, дату його відкриття, термін, на який відкрито рахунок, та суму коштів на рахунку. У цьому разі власникові пластикової картки також надається персональний ідентифікаційний код, який є засобом доступу до депозитного рахунку.

У деяких АБС «Управління депозитами» не виділяється в окрему функціональну підсистему, а інтегрується комплекс, який носить назву «Управління кредитно-депозитними операціями».

Підсистема «Управління цінними паперами». Комерційні банки можуть виконувати наступні операції з цінними паперами:

- випускати і продавати власні акції;
- купувати, продавати і зберігати цінні папери (акції, облігації, векселі тощо);
- проводити операції з цінними паперами за дорученням клієнтів.

У підсистемі «Управління цінними паперами» виділяються такі основні напрямки автоматизації робіт:

- автоматизація обліку операцій з власними акціями банку;
- автоматизація обліку операцій з державними цінними паперами;
- автоматизація обліку операцій з іншими цінними паперами (акціями підприємств, векселями, сертифікатами і т.п.);
- автоматизація депозитарної та реєстраторської діяльності;
- автоматизація управління портфелем цінних паперів, моделювання та прогнозування стану фондового ринку.

Підсистема «Управління касою» необхідна для обліку готівки та організації роботи обмінних пунктів. Ця підсистема являє собою міні-банк, що має свій баланс, рахунки і документацію, в якій відображаються готівкові кошти. В підсистемі виконуються такі основні функції:

- ведення довідника касових символів;
- введення та обробка прибуткових касових документів;
- введення та обробка видаткових касових документів;
- формування та ведення касових журналів;
- формування звітних форм з обліку роботи каси;
- підсистема може працювати в одній локальній мережі з підсистемою «Операційний день банку» або може бути територіально віддаленою і взаємодіяти з банком за допомогою електронної пошти.

Підсистема «Каса» комерційного банку технологічно може складатись з наступних АРМів:

- задачі формування касового плану (АРМ бухгалтера-економіста каси);
- задачі обліку надходження коштів у касу (АРМ касира з обліку надходжень);
- задачі обліку видатку коштів з каси (АРМ касира з обліку видатків);
- задачі інкасації коштів у банку (АРМ нічного касира і бухгалтера з інкасації);
- задачі обліку роботи обмінних валютних пунктів (АРМ співробітника з контролю та нагляду за роботою обмінних валютних пунктів) ;
- задачі обліку наявності коштів у касі банку (АРМ завідуючого касою).

Підсистема «Внутрішньобанківський облік» включає до свого складу задачі, що пов'язані з обліком у самому банку. До цієї підсистеми відносяться такі основні задачі: облік праці і нарахування заробітної плати працівникам банку, облік власних основних засобів банку, облік нематеріальних активів, облік амортизації основних засобів та нематеріальних активів, облік господарських і експлуатаційних витрат.

В підсистемі «Звітність банку» формується бухгалтерська, фінансова та статистична звітність про діяльність комерційного банку. Джерелом інформації для даної підсистеми є база даних, сформована в ОДБ. У кожному комерційному банку встановлено АРМ – Статзвітність, який є програмним продуктом НБУ. За допомогою цього АРМу формується вся статистична звітність комерційних банків у вигляді окремих файлів показників і передається в депозитарій статистичної звітності в НБУ, де ці показники обробляються і формуються необхідні звітні форми.

Підсистема «Управління розрахунками з використанням пластикових карток». Метою функціонування цієї підсистеми є автоматизація безготівкових розрахунків з фізичними особами з використанням пластикових карток. Основними напрямками робіт в підсистемі є наступні:

- обслуговування локальних платіжних карткових систем;
- обслуговування міжнародних платіжних карткових систем;
- обслуговування національних платіжних карткових систем.

Враховуючи те, що картковий бізнес лише починає розвиватись у комерційних банках України, то в більшості АБС представлений лише перший напрямок, який пов'язаний з обслуговуванням локальних платіжних карткових систем. В переважній більшості це системи побудовані на картках з магнітною смужкою, які використовуються для виплати заробітної плати співробітникам банку через банкомат, для сплати послуг співробітників банків в «своїх» їдальнях та магазинах. Також картки з магнітною смужкою використовуються як еквівалент ощадної книжки при відкритті депозитних (ощадних) рахунків.

2.2 Місце аудиту кредитних операцій в корпоративній інформаційній системі комерційного банку

Місце системи внутрішнього аудиту кредитних операцій в інформаційній системі комерційного банку визначається її метою: незалежна оцінка проведення кредитних операцій в банку.

Як видно на рисунку 2.1 та 2.2, аудит кредитних операцій входить до системи внутрішньобанківського аудиту, який в свою чергу входить до модулю аналізу банківської інформації.

Даний модуль має зворотній зв'язок з системою забезпечення функціональної частини АБС. Це пояснюється тим, що до функцій системи внутрішнього банківського аудиту входять і такі, як:

- відповідність інформаційних систем, впроваджених у банку, положенням і вимогам внутрішньої політики, планам розвитку, процедурам, що діють, законодавчим і нормативним актам;
- засоби захисту й забезпечення збереження активів (у тому числі програмного забезпечення) і їхньої наявності;
- засоби забезпечення правильності й цілісності фінансової та операційної інформації, а також ряд загальних аспектів діяльності банку, зокрема: оцінка економічної обґрунтованості й ефективності використання ресурсів; перевірка діяльності банку й функціонування систем з метою з'ясування їхньої відповідності цілям, поставленим перед банком.

Загальна структурна схема побудови АБС представлена на рисунку 2.2.

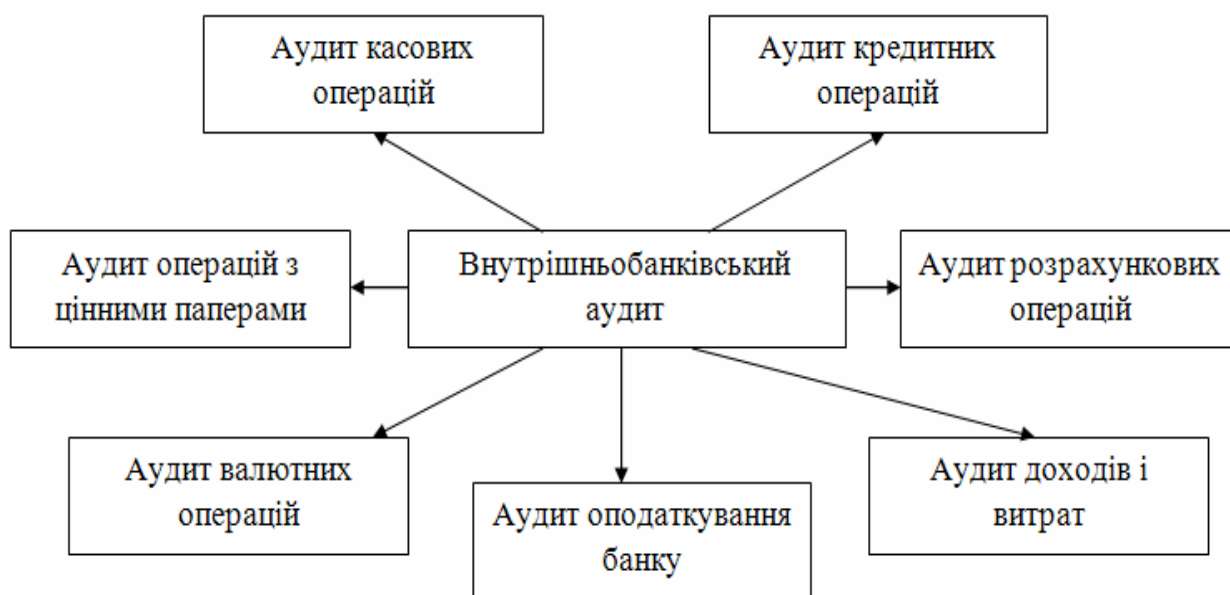


Рисунок 2.1 – Функціональна структура внутрішньобанківського аудиту



Рисунок 2.2 – Структура корпоративної інформаційної системи комерційного банку

Внутрішній аудит кредитних операцій займає досить специфічне місце в інформаційній системі банку. Це пояснюється тим, що внутрішній аудит одночасно є як частиною ІС так і незалежним органом, що перевіряє ІС на відповідність нормам. Цим пояснюється складність створення автоматизованої системи внутрішнього аудиту, і інтеграції її з АБС. Тому зараз на ринку програмного забезпечення для банків досить важко знайти серйозне рішення для автоматизації внутрішнього аудиту.

Отже, задача автоматизації аудиту кредитних операцій є складовою частиною задачі проведення аудиту і звичайно, проведення власне всього комплексу робіт з аудиту діяльності банківської установи. До того ж дана задача пов'язана з іншими задачами, що вирішуються у банку, включаючи і ті, що вирішуються за допомогою використання КІС банку.

Для задачі існує два джерела отримання вхідної інформації:

- дані, що надходять з корпоративної бази даних банку;
- дані первинних документів (анкети, опитувальні листи, журнали фіксації фактичних подій).

За встановленими бізнес правилами банку, Управління внутрішнього аудиту отримує інформацію з корпоративної бази даних (СУБД) у вигляді файлів Microsoft Access або Microsoft SQL. Структури даних первинних документів відповідають вимогам до документації банку.

Відображення інформаційного взаємозв'язку вхідної і вихідної інформації представлено на рисунку 2.3.

Використання автоматизованих банківських систем (АБС) дозволяє банкам скоротити витрати на обслуговування клієнтів, контролювати виконання банківських операцій у реальному масштабі часу, що значною мірою сприяє підвищенню якості прийняття оперативних, тактичних, а іноді і стратегічних фінансових рішень [26].



Рисунок 2.3 – Інформаційний зв’язок вхідної і вихідної інформації задачі «Аудит кредитних операцій»

Внутрішній аудит, будучи первинною формою банківського контролю, став основою всієї системи керування комерційними банками. Створюючи прозоре інформаційне поле, внутрішній аудит сприяє більш глибокому й обґрунтованому аналізу діяльності банку, створенню справедливої й ефективної системи стимулювання. Це відповідає загальноприйнятим міжнародним стандартам керування комерційними банками.

2.3 Архітектура та технології створення автоматизованої системи внутрішнього аудиту в банках

Для створення автоматизованої системи внутрішнього аудиту як складової частини КІС банку була обрана клієнт-серверна архітектура, тому що

вона є більш популярною і має меншу кількість вад у порівнянні з файл-серверною архітектурою. Детальніша інформація про зазначену архітектуру представлена нижче.

Базу знань, сформовану за допомогою методів навчання, функціонально можна представити як множину можливих станів середовища та оцінки різних можливих дій агента у цих станах. На основі таких оцінок агент вибирає певну дію для прийняття рішення, самі ж оцінки формуються на основі попереднього досвіду агента. Таким чином, у кожен момент часу активно використовується агентом певний невеликий фрагмент бази знань (поточний стан середовища). При цьому інші аналогічні агенти могли б використовувати (і навіть модифікувати) інші фрагменти цієї ж бази знань. У багатьох випадках такі централізовані бази знань можуть бути досить корисними, зокрема значно прискорити процес навчання.

Терміни «агент» та «середовище» розуміємо тут у тому значенні, яке в них вкладається у класичних працях про навчання [17]. Зокрема, під агентом розумітимемо елемент, який безпосередньо взаємодіє з середовищем: одержує інформацію із зовнішнього середовища, впливає на нього та одержує числові оцінки своїх дій. За такого вибору дій агент як керується інформацією з бази знань, так і модифікує базу знань на основі своїх оцінок.

Існує багато прикладів реалізації централізованих баз знань, але вони мають ті чи інші вади. Тому для ефективної реалізації таких баз знань потрібно обирати саме клієнт-серверну архітектуру.

Методи навчання дозволяють формувати базу знань за одночасного розв'язування агентами задач (на основі числових оцінок), а також дають можливість безконфліктного паралельного використання єдиної бази знань багатьма агентами, а основне – значно прискорюють навчання завдяки збільшенню інтенсивності потоку досвіду від багатьох клієнтів.

Сам сервер баз знань та API для написання клієнтського програмного забезпечення розроблені на базі технології MS Framework .Net [27].

Загальна схема клієнт-серверної архітектури інтелектуальних баз знань показана на рисунку 2.4.

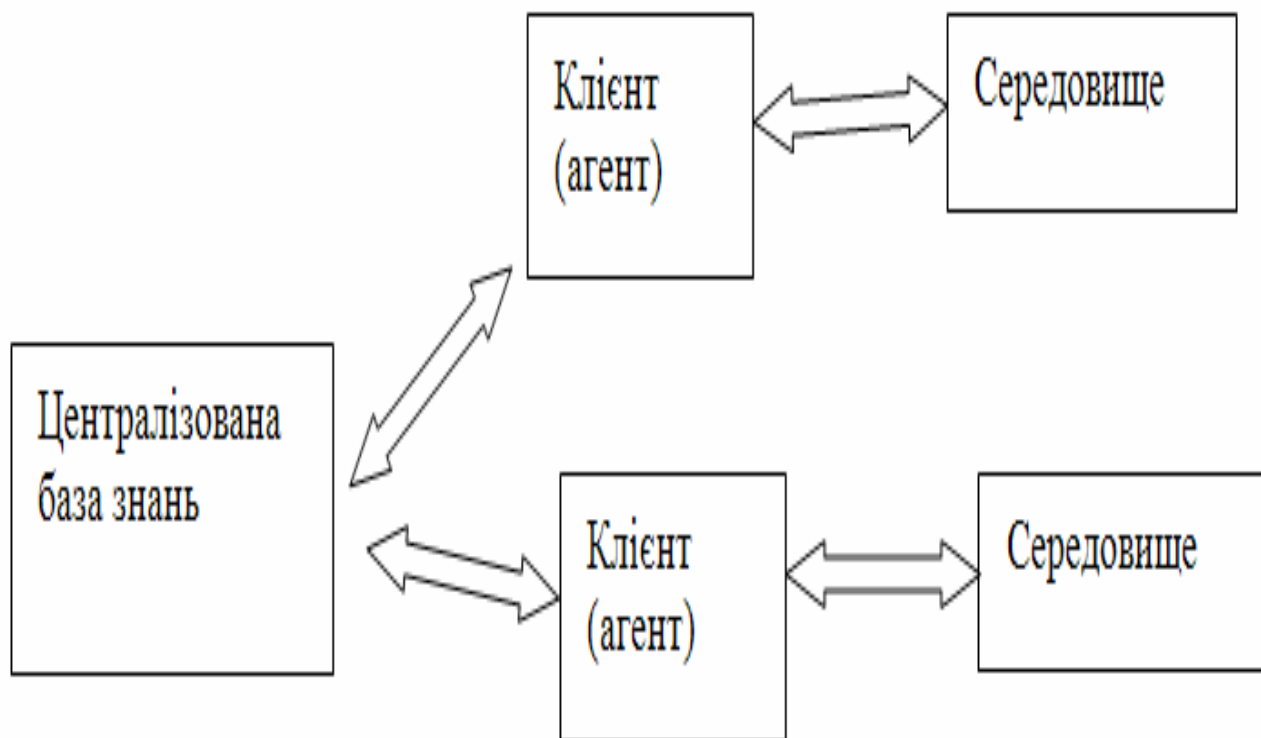


Рисунок 2.4 – Загальна схема клієнт-серверної архітектури інтелектуальних баз знань

Тут централізована база знань «консультує» клієнтів щодо поведінки у середовищі. Хоча кожен клієнт взаємодіє із своїм середовищем і нічого «не знає» про інших клієнтів, проте для всіх клієнтів, що під'єдналися до однієї й тієї ж бази знань, середовища мають бути еквівалентними. Взаємодія між агентом та середовищем відбувається на основі дискретного перетворювача. Агент взаємодіє із середовищем послідовно, надсилаючи та одержуючи фіксовані сигнали (символи із деякого фіксованого алфавіту A), графічно схему такої взаємодії зображено на рисунку 2.5.

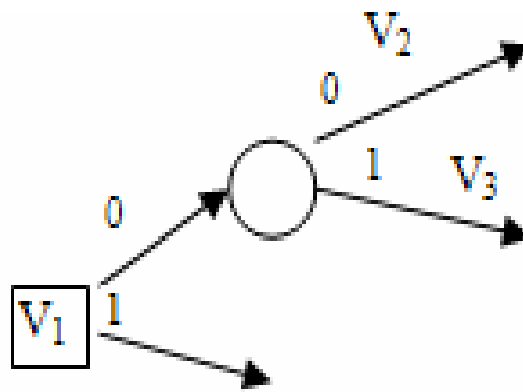


Рисунок 2.5 – Схема взаємодії між агентом та середовищем

Таким чином, для кожного клієнтського під'єднання у виділений момент часу сервер бази знань зберігає деяку маску, що ідентифікує певний стан (ідентифікатор). Одержуючи символ із середовища агент передає його серверу, відповідно до цього змінюється поточний стан даного клієнта. Аналогічно, коли сервер очікує символ від агента (прийняття рішення), агент звертається за рішенням до централізованої бази знань, яка формує рішення для даної ситуації – випробуване на основі досвіду багатьох клієнтів.

Загалом, як зазначалося вище, паралельне використання єдиної бази знань багатьма клієнтами не представляє ніяких проблем. Структура переходів при цьому не змінюється, а змінюються лише ціни станів (чи переходів), які є основними факторами для вибору рішення, проте у межах незначного проміжку часу вони змінюються не дуже сильно.

Переваги клієнт-серверної архітектури:

- мінімум затрат на обслуговування бізнес-процесів;
- максимальна оперативність при оперуванні даними;
- зручність в обслуговуванні, більшість операцій може виконуватись автоматично;
- один працівник може легко обслуговувати кілька процесів одночасно без особливих зусиль;
- мінімум затрат на комунікації між підрозділами компанії;

- оперативне і гнучке отримання звітів про діяльність компанії;
- роботу з системою можна здійснювати з будь якого комп'ютера що приєднаний.

Запропонована фірмою Microsoft технологія розробки програмного забезпечення на платформі .Net поєднує в собі максимальну простоту при розробці та максимальну ефективність програмного коду. Технологія Remoting.Net забезпечує зручну і в той же час безпечну взаємодію між різними програмами (в тому числі і на різних комп'ютерах) [27]. .NET Remoting робить посилання на об'єкт ремоутінгу доступним клієнтській програмі, потім створює екземпляр об'єкта та використовує об'єкт ремоутінгу, ніби він був локальним об'єктом. Тим не менше, фактично виконання коду відбувається на серверному боці. Об'єкт ремоутінгу ідентифікується за допомогою Activation URL, екземпляр об'єкта створюється при встановленні з'єднання з URL. Середовище ремоутінгу створює приймач запитів, коли сервер реєструє канал, який використовується для з'єднання з об'єктом ремоутінгу. На клієнтському боці середовище ремоутінгу створює проксі, який діє як псевдо-екземпляр об'єкта ремоутінгу. В ньому не реалізована функціональність об'єкта ремоутінгу, але в нього такий саме інтерфейс. Таким чином інфраструктура ремоутінгу повинна знати публічний інтерфейс об'єкта ремоутінгу наперед. Будь-який виклик методу об'єкта, включаючи ідентифікацію самого методу та всі параметри передані до нього, серіалізуються в байтовий потік та передаються через комунікаційний, залежний від протокола, канал до приймаючого проксі об'єкта на серверній стороні («маршалінг»). На серверній стороні проксі вичитує з потоку усі отримані дані та робить виклик методу на об'єкті ремоутінгу від імені клієнта. Результати серіалізуються та передаються на клієнт, де проксі вичитує результат та передає його до програми, що викликала об'єкт ремоутінгу.

Для написання клієнтів до сервера баз знань використовується бібліотека з проксі-класом, яка відкриває з'єднання до бази знань та працює із цим

з'єднанням. Перед початком роботи із з'єднанням клієнтська програма створює екземпляр класу з'єднання.

2.4 Підсистеми забезпечення функціональної частини

Для створення автоматизованої системи потрібно створити ефективну структуру бази даних, тобто визначення обсягів і типів даних, від яких безпосередньо залежить продуктивність бази даних. Об'єм інформації впливає на розмір бази даних, а типи даних виступають у ролі факторів, що визначають обмеження, які накладаються на структуру даних.

При розробці даного проекту будуть враховані такі найбільш розповсюджені елементи збору інформації, як:

- прізвище, ім'я, по батькові та дата народження;
- адреса проживання клієнтів;
- посада працівників;
- місце роботи та номер телефону клієнта;
- паспортні дані клієнтів;
- сума кредиту, відсоток по кредиту, дата отримання та погашення;
- вид та сума заставного майна;

Існують наступні підсистеми забезпечення: програмне забезпечення, апаратне забезпечення, організаційне забезпечення, інформаційне та алгоритмічне забезпечення.

Програмне забезпечення.

Необхідною частиною будь-якої комп'ютерної інформаційної системи є її програмне забезпечення. Програмне забезпечення – це комплекс програм, призначених для вирішення на комп'ютері інформаційної системи визначеного класу завдань. За своїм призначенням програмне забезпечення – невід'ємна частина будь-якої інформаційної системи, що відбиває принципову основу організації обчислювального процесу, програмного принципу обробки інформації комп'ютерною системою. Без відповідного програмного

забезпечення будь-яка, навіть найкращим чином розроблена апаратура, була б практично такою же непотрібною, як, наприклад, магнітофон без системи звукозапису. Але в наш час існує велика кількість програмного забезпечення і тому виникає проблема вибору. Для реалізації проекту автоматизації бібліотеки необхідно в першу чергу обрати гідну СУБД [12].

При виборі СУБД необхідно враховувати наступні фактори:

- максимальна кількість користувачів, які одночасно звертаються до бази;
- характеристики клієнтського персонального комп'ютера;
- апаратні компоненти сервера;
- серверна операційна система;
- рівень кваліфікації персоналу.

На сьогоднішній день відомою є велика кількість різноманітних серверів баз даних SQL. Зупинитися слід на двох СУБД – Oracle8i, Microsoft SQL Server. Порівняння проходить на кожному із основних етапів функціонування:

- конфігурування системи;
- моніторинг;
- настройка;
- обробка запитів;
- розробка серверних і клієнтських модулів.

Отже, пакет Oracle8i наділений найбільш розвиненим набором функцій для роботи з мовою Java й доступу до даних через Інтернет, системою оптимізації одночасного доступу. Єдиним недоліком даної СУБД є складність адміністрування, до того ж вона є достатньо дорогою.

Таким чином, серед основних властивостей СУБД Oracle слід виділити наступні:

- висока надійність;
- можливість розбиття великих баз даних на розділи;
- наявність універсальних засобів захисту інформації;

- ефективні методи максимального підвищення швидкості обробки запитів;
- наявність широкого спектру засобів розробки, моніторингу та адміністрування;
- орієнтація на Інтернет технології.

Також необхідно зазначити, що розроблена фірмою Oracle система оптимізації одночасного доступу виключає ситуацію, коли одному користувачу слід чекати, доки інший завершить зміни в змісті бази даних, тобто відсутнє блокування на читання.

Найважливішими характеристиками СУБД Microsoft SQL Server є наступні:

- простота адміністрування;
- можливість підключення до Web;
- швидкодія й функціональні можливості механізму сервера СУБД;
- наявність засобів віддаленого доступу.

Дана система має чудові засоби тиражування, які дозволяють синхронізувати дані персонального комп'ютера з інформацією бази даних і навпаки. Зрештою дана СУБД представляє собою сучасну повно функціональну базу даних, яка ідеально підходить для малих та середніх організацій. Недоліком даної системи є те, що загальноприйнятою тенденцією в XXI столітті став перехід на платформу LINUX, а SQL Server функціонує тільки в середовищі Windows [12].

Для створення клієнтської програми, яка взаємодіє з базою даних було обрано Microsoft Visual Studio.

Одним з найбільш поширених підходів, які використовують при розробці клієнтських прикладень для взаємодії з реляційними базами даних під ОС Windows, є використання інструментарію Visual Studio корпорації Microsoft. MS Visual Studio – це інтегроване середовище розробки (Integrated Development Environment (IDE)) для створення, документування, запуску та коректування програм, написаних на сучасних мовах створення програмних систем. Це потужний інструмент професійної розробки складних прикладень, один з

найкращих у світі. Також у рамках проекту необхідно встановити каркас Microsoft.NET Framework відповідної версії для роботи MS Visual Studio.

Апаратне забезпечення.

Функціонування інформаційної системи ґрунтується на використанні технічних засобів, які забезпечують збирання, реєстрацію, передачу, зберігання, обробку і тиражування інформації, а також процеси взаємодії людини і комп'ютера для вирішення задач управління об'єктом. Передбачається, що інформаційна система створюватиметься на базі впровадження: нових радіо-комунікаційних зв'язків, сучасних комп'ютерної та факсимільно-друкуючої техніки, локальної комп'ютерної мережі і відповідних програмних засобів. Для ефективного використання різноманітної номенклатури технічних засобів необхідно забезпечити їх інформаційну (єдині формати даних і агрегатів даних, спільність тезаурусу і класифікаторів, наявність інтерфейсів кодів і технічних пристроїв), програмну і технічну (пропускні здатності пристроїв і каналів, можливість спряження пристроїв і каналів штатними засобами) сумісність. З цією метою всі засоби інформаційної системи повинні бути об'єднані в єдиний комплекс технічних засобів інформаційної системи. Ця специфічна вимога є додатковою до звичайних вимог, яка охоплює виконання поставлених завдань з необхідною якістю, виконання вимог економіки, надійності, економічності і естетики.

Структурними елементами апаратного забезпечення є технічні засоби, методичні і управлінські матеріали, технічна документація, обслуговуючий персонал, які забезпечують виконання функціональних задач інформаційної системи.

Так як для реалізації проекту було обрано Microsoft Visual Studio, то апаратне забезпечення для даних програмних продуктів повинне відповідати наступним мінімальним вимогам, що представлені у таблиці 2.1.

Таблиця 2.1 – Мінімальні вимоги до сервера та середовища розробки

Пристрій	Мінімальні вимоги
Монітор	VGA або вища роздільність – як мінімум 1024x768
Вказівний пристрій	Пристрій типу «миша» або ж сенсорний навігатор, клавіатура
Дисковод	CD/DVD пристрій
Процесор	Pentium IV-сумісний або ж вище, мінімальна швидкість ядра – 2 GHz, рекомендовано – 2,5 GHz
Оперативна пам'ять	Мінімально 1 GB, рекомендовано - 2 GB

Вимоги до апаратного забезпечення клієнтського місця представлені у таблиці 2.2.

Таблиця 2.2 – Мінімальні вимоги до клієнтського місця

Пристрій	Мінімальні вимоги
Монітор	VGA або вища роздільність – як мінімум 1024x768
Вказівний пристрій	Пристрій типу «миша» або ж сенсорний навігатор, клавіатура
Дисковод	CD/DVD пристрій
Процесор	Pentium III-сумісний та вище, мінімальна швидкість ядра – 600 GHz, рекомендовано – 1 GHz або Athlon, мінімальна швидкість ядра – 1,2 GHz, рекомендовано – 2 GHz
Оперативна пам'ять	Мінімально 128 MB, рекомендовано - 1 GB

Організаційне забезпечення.

Під організаційним забезпеченням слід розуміти узгодження по місцю, часу і меті сумісне функціонування окремих виконавців, колективів і технічних засобів. Воно повинно здійснюватися і регулюватися деякими правилами взаємодії, які утворюють правовий та моральний кодекс і складають основу правового забезпечення. Тому організаційне забезпечення будується на нормативних актах правового забезпечення, а правове забезпечення знаходить своє втілення в організаційному забезпеченні.

Організаційне забезпечення інформаційної системи охоплює сукупність засобів, методів і відповідного персоналу. Воно повинно забезпечити проведення техніко-економічного аналізу існуючої системи управління, вибору і постановки задач побудови інформаційної системи на етапі розробки і впровадження [12].

В результаті впровадження автоматизованої системи у відділі внутрішнього аудиту в банку зміни у функціональній відповідальності працівників не відбудуться, аудиторам потрібно пройти навчальні курси поробі у програмному додатку.

Ергономічне забезпечення.

Ергономічне забезпечення охоплює сукупність методів і засобів, призначених для створення оптимальних умов для ефективної діяльності і навчання операторів з складу персоналу інформаційної системи. Ергономічне забезпечення включає в себе:

- комплекс документації, яка містить ергономічні вимоги до робочих місць і здійснює експертизу робочих місць;
- комплекс методів, учбово-методичних матеріалів і технічних засобів підготовки персоналу до роботи;
- комплекс методів і засобів, які забезпечують професійний відбір.

В плані ергономічного забезпечення на етапах проектування інформаційної системи визначається ступінь і рівень участі людини в системі управління, вимоги до форми представлення інформації, умови оточуючого середовища діяльності людини, порядок роботи і відпочинку персоналу,

нормативи навантаження і надійності персоналу; вимоги до технічних засобів, способи взаємодії персоналу і технічних засобів.

Інформаційне та алгоритмічне забезпечення розглянемо у наступному розділі.

Створений графічний інтерфейс автоматизованої системи позитивно впливає на швидкість освоєння програмного продукту та подальшої роботи з ним. В майбутньому це впливає на швидкість роботи користувача, що в свою чергу зменшує час обробки даних аудитором. Обрана кольорова гама не провокує швидку втоми очей і користувач може більше часу працювати за комп'ютером.

3 РЕАЛІЗАЦІЯ ПРОГРАМНОГО РІШЕННЯ ВНУТРІШНЬОГО АУДИТУ КРЕДИТНИХ ОПЕРАЦІЙ КОМЕРЦІЙНОГО БАНКУ

3.1 Вхідні та вихідні дані системи

Інформаційне забезпечення є технологічною основою автоматизованої системи обробки економічної інформації. Основою для його формування є вхідна, вихідна, умовно-постійна інформація, архівна інформація та база знань.

Вхідна інформація буде формуватися з особових даних клієнтів, яким надається кредит, всі ці дані будуть внесені до бази даних відповідно до об'єктів бази даних.

До вхідної інформації відносяться такі об'єкти бази даних:

а) об'єкт «Працівники» містить інформацію про персональні дані працівника, а саме: прізвище, ім'я, по-батькові, посаду та номер телефону;

б) об'єкт «Клієнти» призначений для зберігання інформації про персональні дані клієнтів, такі як: прізвище, ім'я, по-батькові, посаду та номер телефону, дохід, паспортні дані, дату народження;

в) об'єкт «Адреси» призначений для зберігання інформації про адреси проживання клієнтів і має такі атрибути, як: країна, область, населений пункт, вулиця, будинок, квартира;

г) об'єкт «Відділення» призначений для зберігання інформації про відділення, а саме назва відділення та юридична адреса;

д) об'єкт «Кредитна операція» містить інформацію про параметри кредиту, який отримує клієнт. Такими параметрами виступають вид кредиту, сума кредиту, відсоткова ставка по кредиту, строк надання кредиту, цільове використання коштів;

е) об'єкт «Забезпечення» призначений для зберігання інформації про вид забезпечення, який надає клієнт, для отримання кредиту.

У якості вихідної інформації системи буде виступати вікно оцінки фінансового стану позичальника, яке представлено на рисунку 3.1, вікно виявлених помилок в результаті перевірки, яке представлено на рисунку 3.2, а також вікно з аудиторським висновком, яке представлено на рисунку 3.3. Тобто система в результаті перевірки проведених кредитних операцій в банку виведе

повідомлення про дотримання (недотримання) процедури відкриття певного кредитного договору та вкаже помилки, які були допущені.

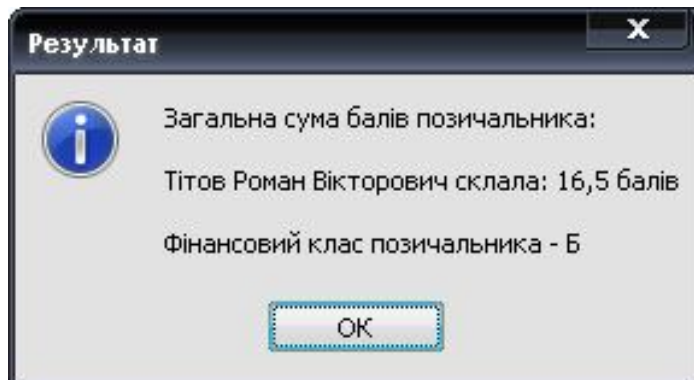


Рисунок 3.1 – Вікно оцінки фінансового стану позичальника

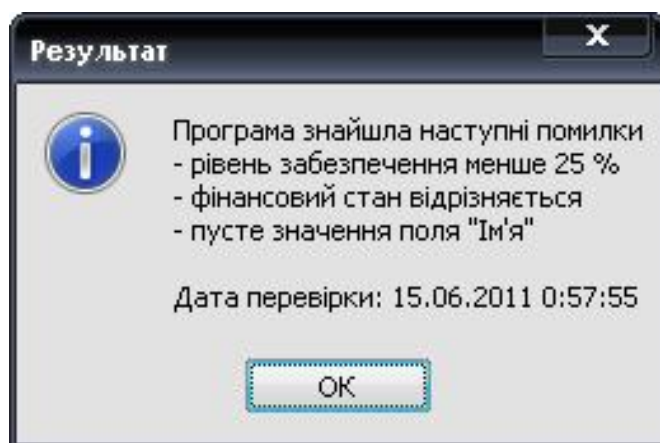


Рисунок 3.2 – Вікно результату перевірки кредитного договору

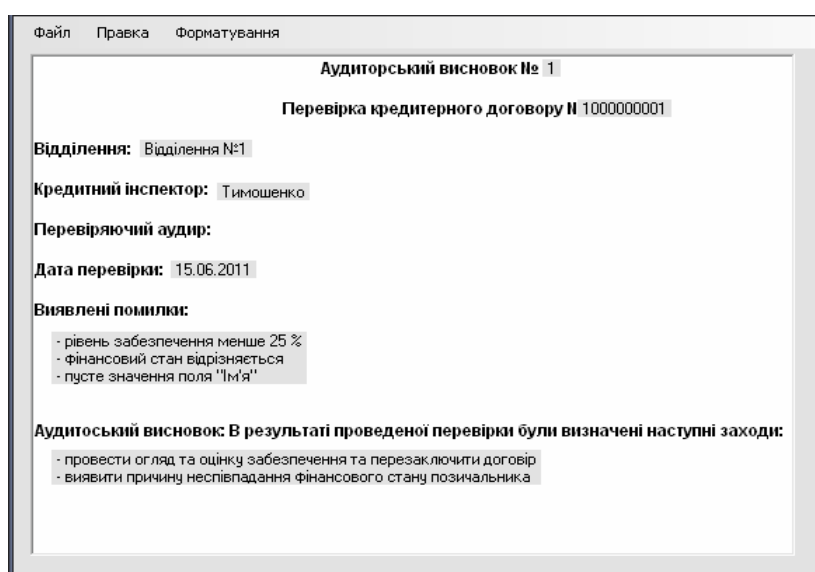


Рисунок 3.3 – Вікно аудиторського висновку

У якості умовно-постійної інформації виступає інформація про поточні нормативні значення, відносно яких аудитор проводить перевірку. Дана інформація представлена на рисунку 3.4.

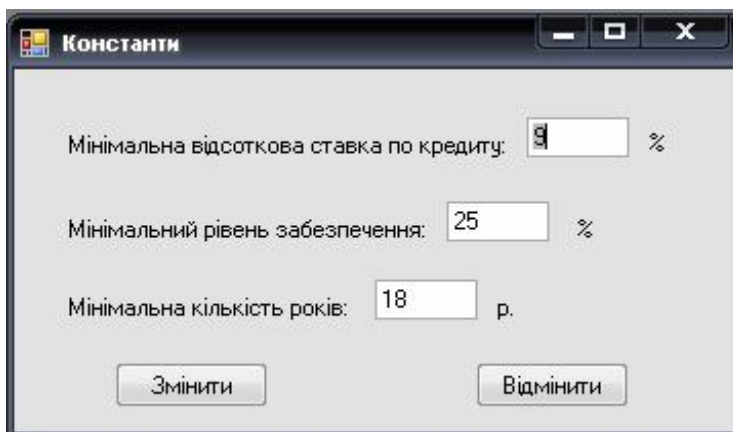


Рисунок 3.4 – Вікно умовно-постійної інформації

До архівної інформації відноситься журнал проведених перевірок, де зберігається інформація про дату перевірки, номери перевірених договорів та виявлені помилки. Вікно з архівною інформацією представлено на рисунку 3.5.

№	№ кредитної операції	Дата перевірки	Результат перевірки
93	1000000002	6/11/2011	є_помилки
94	1000000004	6/11/2011	є_помилки
95	1000000006	6/14/2011	є_помилки
96	1000000002	6/14/2011	не_знайдено
97	1000000001	6/14/2011	є_помилки
98	1000000001	6/14/2011	є_помилки
99	1000000004	6/14/2011	є_помилки
100	1000000004	6/14/2011	є_помилки
101	1000000003	6/14/2011	є_помилки

Рисунок 3.5 – Вікно архівної інформації

3.2 Структура та особливості реалізації інформаційного та алгоритмічного забезпечення

База даних складатиметься з базових шести таблиць. Розглянемо їх більш докладніше. Для зручності зобразимо об'єкти бази даних в табличному вигляді.

Таблиця “Адреси” зберігає масив адрес. В таблиці 3.1 розміщені атрибути об'єкта “Адреси”.

Таблиця 3.1 – Атрибути об'єкта “Адреси”

Назва атрибута	Тип даних	Обмеження	Призначення атрибута
АдресаІдн	Int (4)	NOT NULL	Виступає первинним ключем таблиці
КлієнтІдн	Int (4)	NOT NULL	Містить ідентифікатор клієнта
Країна	Varchar(30)	NOT NULL	Містить назву країни
Область	Varchar(30)	NOT NULL	Містить назву області
Населений пункт	Varchar(30)	NOT NULL	Містить назву населеного пункту
Вулиця	Varchar(30)	NOT NULL	Містить назву вулиці
Будинок	Varchar(10)	NOT NULL	Містить номер будинку
Квартира	Varchar(10)	NULL	Містить номер квартири

Таблиця “Відділення” містить інформацію про відділення банку. В таблиці 3.2 розміщені атрибути об'єкта “Відділення”.

Таблиця 3.2 – Атрибути об'єкта «Відділення»

Назва атрибута	Тип даних	Обмеження	Призначення атрибута
ВідділенняІдн	Int (4)	NOT NULL	Виступає первинним ключем таблиці
Назва відділення	Varchar(30)	NOT NULL	Містить назву відділення
Юридична адреса	Varchar(30)	NOT NULL	Містить юридичну адресу відділення

Таблиця «Забезпечення» містить інформацію про забезпечення, що надається клієнтом для отримання кредиту. В таблиці 3.3 розміщені атрибути об'єкта «забезпечення».

Таблиця 3.3 – Атрибути об'єкта «Забезпечення»

Назва атрибута	Тип даних	Обмеження	Призначення атрибута
ЗабезпеченняІдн	Int (4)	NOT NULL	Виступає первинним ключем таблиці
Кредитний ДоговірІдн	Int (4)	NOT NULL	Містить ідентифікатор кредитного договору
Вид забезпечення	Varchar(30)	NULL	Містить інформацію про вид забезпечення
Сума	Money	NULL	Містить суму відповідного забезпечення

Таблиця «Працівники» містить інформацію про працівників, що працюють у кредитному відділі банку. В таблиці 3.4 розміщені атрибути об'єкта «Працівники».

Таблиця 3.4 – Атрибути об'єкта «Працівники»

Назва атрибута	Тип даних	Обмеження	Призначення атрибута
ПрацівникІдн	Int (4)	NOT NULL	Виступає первинним ключем таблиці
ВідділенняІдн	Int (4)	NOT NULL	Містить ідентифікатор відділення
Прізвище	Varchar(30)	NOT NULL	Містить прізвище працівника
Ім'я	Varchar(15)	NOT NULL	Містить ім'я працівника
Побатькові	Varchar(30)	NULL	Містить побатькові працівника
Номер	Varchar(15)	NULL	Містить телефонний номер працівника
Посада	Varchar(15)	NOT NULL	Містить інформацію про посаду працівника

Таблиця «Кредитна_операція» містить інформацію про кредитні операції, що були укладені. В таблиці 3.5 розміщені атрибути об'єкта «Кредитна_операція».

Таблиця 3.5 – Атрибути об'єкта «Кредитна_операція»

Назва атрибута	Тип даних	Обмеження	Призначення атрибута
Кредитний ДоговірІдн	Int (4)	NOT NULL	Виступає первинним ключем таблиці
КлієнтІдн	Int (4)	NOT NULL	Містить ідентифікатор клієнта
ПрацівникІдн	Int (4)	NOT NULL	Містить ідентифікатор працівника

Продовження таблиці 3.5

Назва атрибута	Тип даних	Обмеження	Призначення атрибута
Вид кредиту	Varchar(15)	NOT NULL	Містить вид кредиту
Сума кредиту	Money	NOT NULL	Містить суму кредиту
Валюта	Varchar(10)	NULL	Містить валюту кредиту
Відсоток по кредиту	Money	NOT NULL	Містить відсоткову ставку по кредиту
Дата отримання	Date	NOT NULL	Містить дату отримання кредиту клієнтом
Строк	Int(4)	NOT NULL	Містить строк користування кредитом
Цільове використання	Varchar(30)	NULL	Містить інформацію про цільове використання кредиту

Таблиця «Клієнти» містить інформацію про клієнтів банку, що отримали кредит. В таблиці 3.6 розміщені атрибути об'єкта «Клієнти».

Таблиця 3.6 – Атрибути об'єкта «Клієнти»

Назва атрибута	Тип даних	Обмеження	Призначення атрибута
КлієнтІдн	Int (4)	NOT NULL	Виступає первинним ключем таблиці
Прізвище	Varchar(30)	NOT NULL	Містить прізвище клієнта
Ім'я	Varchar(15)	NOT NULL	Містить ім'я клієнта
Побатькові	Varchar(30)	NULL	Містить побатькові клієнта

Продовження таблиці 3.6

Назва атрибута	Тип даних	Обмеження	Призначення атрибута
Місце роботи	Varchar(30)	NOT NULL	Містить інформацію про місце роботи клієнта
Посада	Varchar(15)	NOT NULL	Містить інформацію про посаду клієнта
Телефон	Varchar(15)	NULL	Містить інформацію про контактний номер телефону клієнта
Щомісячний дохід	Money	NOT NULL	Містить інформацію про щомісячний дохід клієнта
Серія паспорту	Varchar(10)	NOT NULL	Містить серію паспорту
Ким виданий	Varchar(30)	NOT NULL	Містить інформацію про те, ким був виданий паспорт
Ідентифікаційний номер	Int(15)	NOT NULL	Містить персональний ідентифікаційний номер клієнта
Дата народження	Date	NOT NULL	Містить дату народження клієнта
Стать	Varchar(10)	NULL	Містить стать клієнта

Отже, після означення таблиць, полів, індексів і зв'язків між таблицями потрібно поглянути на базу даних, що проектується, в цілому і проаналізувати її, використовуючи правила нормалізації, з метою усунення логічних помилок. Важливість нормалізації полягає в тому, що вона дозволяє розбити великі відношення, що, як правило, містять великий надлишок інформації, на дрібніші

логічні одиниці, що групують лише дані, об'єднані «за природою». Отже, ідея нормалізації полягає в наступному. Кожна таблиця в реляційній базі даних задовольняє умову, згідно якій в позиції на перетині кожного рядка і стовпчика таблиці завжди знаходиться єдине значення, і ніколи не може бути множини таких значень.

Схема бази даних автоматизованої задачі представлена рисунку 3.6.

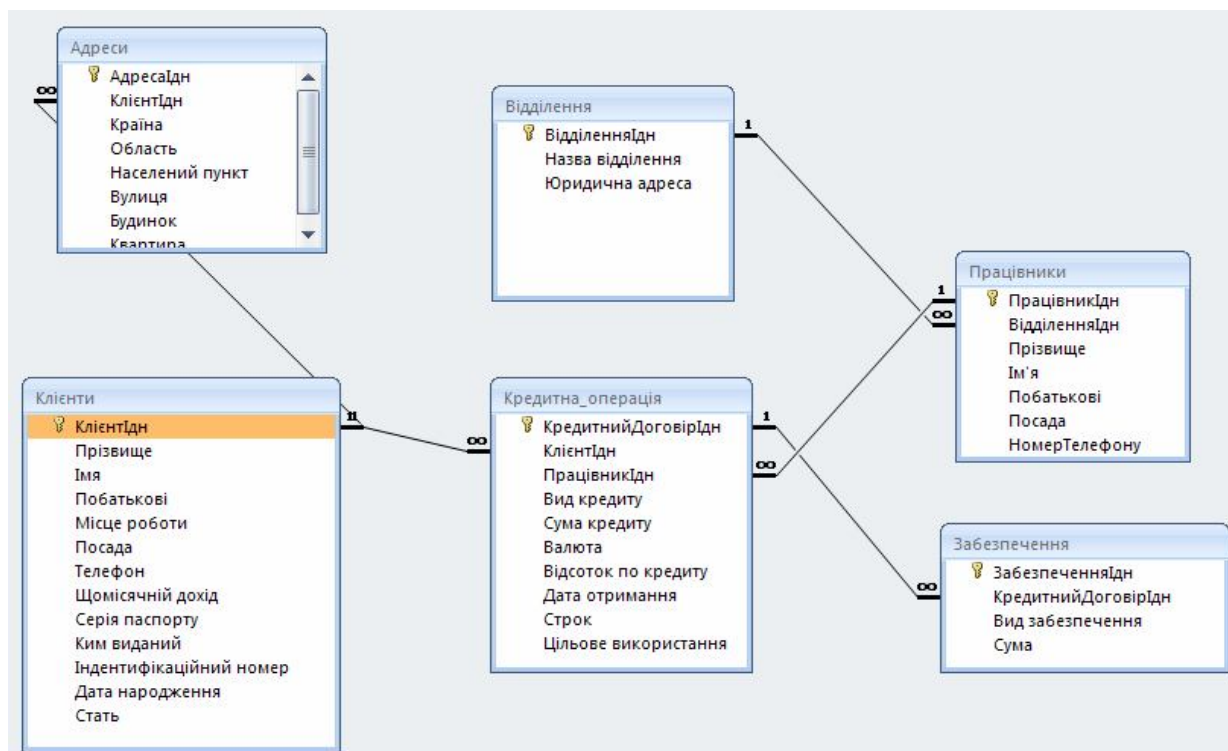


Рисунок 3.6 – Схема бази даних

Після застосування правил нормалізації логічні групи даних розміщуються не більше, як в одній таблиці. Це надає такі переваги:

- дані легко поновлювати або знищувати;
- виключається можливість неузгодження копій даних;
- зменшується можливість введення некоректних даних.

Теорія нормалізації ґрунтується на тому, що певний набір таблиць має кращі властивості при включенні, модифікації і знищенні даних, ніж всі інші набори таблиць, за допомогою яких можуть бути представлені ті ж дані. Введення нормалізації відношень при розробці концептуальної інформаційної

моделі забезпечує мінімальний об'єм фізичної інформації і її максимальну швидкодію.

Для досягнення ефективного, стабільного проектування та розробки програмного забезпечення потрібно мати чітку організацію відповідних робіт, яка забезпечується в межах спеціального проекту. Для цього потрібно спочатку побудувати загальний алгоритм виконання поставленої задачі автоматизації у вигляді блок-схеми, який представлений на рисунку 3.7, а потім алгоритм виконання перевірки аудитора, який представлений на рисунку 3.8.

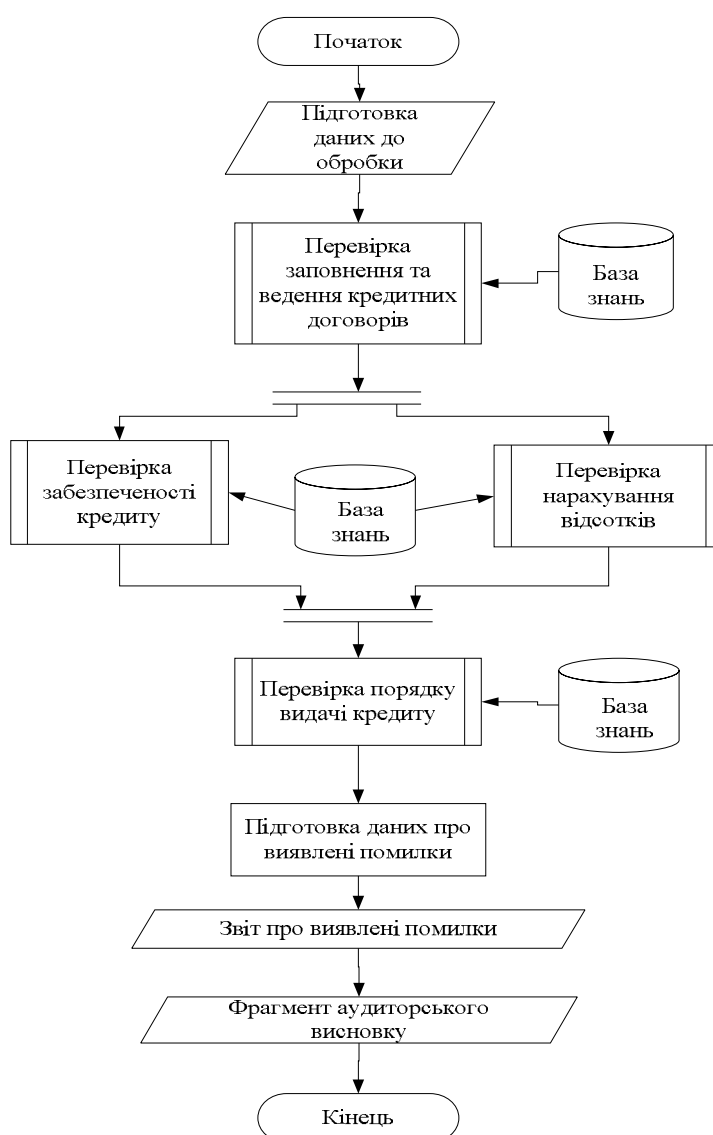


Рисунок 3.7 – Алгоритм автоматизації аудиту кредитних операцій

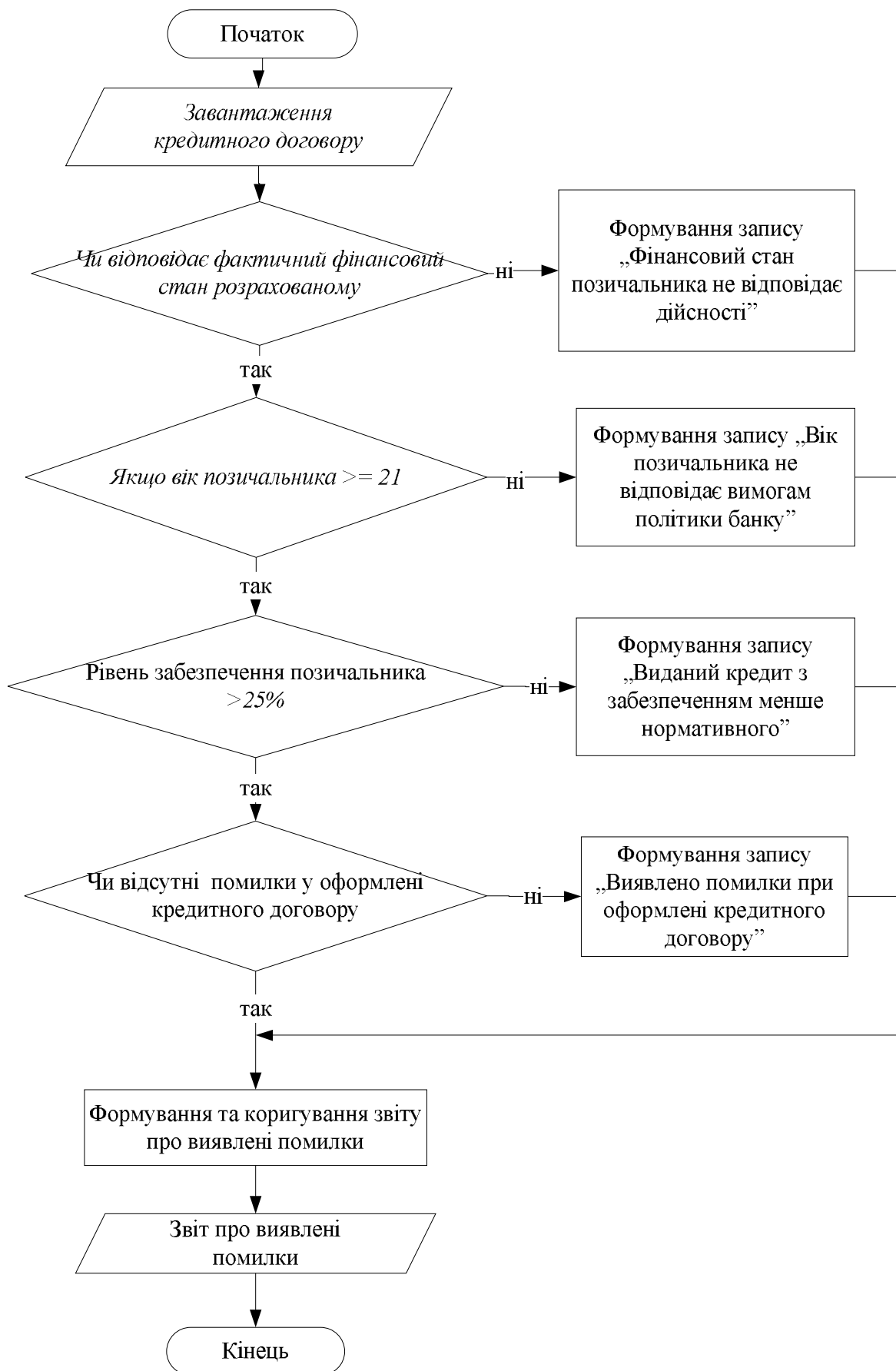


Рисунок 3.8 – Алгоритм перевірки кредитного договору

Алгоритм виконання поставлення задачі автоматизації необхідний для логічного та детального розподілу усіх робіт на окремі операції, які плануються відповідно до обраної задачі.

Детальніше розглянемо алгоритм розрахунку кредитоспроможності фізичної особи, який використовується у ПАТ «Райффайзен банк Аваль». Для цього використовується рейтингова, бальна оцінка. Вона передбачає розрахунок рейтингового балу по кожному з тринадцяти показників. Розглянемо більш детально, яким чином це робиться.

При оцінці по першому розділу максимальний рейтинг, який можна присвоїти позичальнику, становить 15 балів, враховуючи, що по окремим показникам встановлена окрема бальність.

По показнику «Вік позичальника»:

- від 18 років до 23 років або від 55 років = 0 балів;
- від 23 років до 45 = 2 бали;
- від 45 років до 55 років = 1 бал.

По показнику «Стаж позичальника на останньому місці роботи»:

- менше 6 місяців = 0 балів;
- від 6 до 12 місяців = 0,5 бала;
- від 12 до 24 місяців = 1 бал;
- від 24 місяців = 2 бали.

По показнику «Загальний стаж роботи позичальника»:

- до 1 року = 0 балів;
- від 1 року до 3 років = 1 бал;
- від 3 років = 2 бали.

По показнику «Ким працює позичальник»:

- студент – 0 балів;
- пенсіонер = 2 бали;
- найманий робітник або військовий, або військовий пенсіонер, або фізична особа - суб'єкт підприємницької діяльності = 3 бали;

– держслужбовець, керівник підприємства, власник підприємства = 4 бали.

По показнику «Освіта позичальника»:

– середня базова (9 класів і менше) або середня повна, незакінчена вища = 0 балів;

– середня спеціальна = 1 бал;

– вища = 2 бали.

По показнику «Сімейний стан позичальника»:

– одружений/заміжня = 2 бали;

– неодружений/Незаміжня або розведений(а) – 0 балів;

– вдівець(вдова) = 1 бал.

По показнику «Наявність дітей у позичальника»:

– немає або більше 3 дітей = 0 балів;

– одна або дві дитини = 1 бал;

– трое дітей = 0,5 бала.

Максимальна кількість балів, яку Позичальник може набрати в процесі оцінки по другому розділу складає 12 балів.

По показнику «Вага щомісячного погашення по запитуваному кредиту в щомісячному доході позичальника»:

– до 30% = 5 балів;

– від 30% до 50% = 4 бали;

– від 50% до 70% = 3 бали;

– від 70% = 0 балів.

По показнику «Наявність майна у власності позичальника»:

– майно відсутнє або документально не підтверджене = 0 балів;

– один або більше автомобіль(ів) = 2 бали;

– земельна ділянка = 2 бали;

– садовий дім (в т.ч. враховується садовий наділ, на якому побудовано дім) = 4 бали;

- квартира або жилий дім (в т.ч. враховується земельна ділянка, на якій побудовано дім) = 5 балів;
- комерційна нерухомість (один або декілька об'єктів) = 5 балів;
- комбінація з двох або більше вище перерахованих елементів = 6 балів (тільки для декількох автомобілів існує свій показник (див. вище), тільки для декількох земельних ділянок = 3 бали).

По показнику «Наявність депозиту»:

- є = 1 бал;
- немає = 0 балів.

При оцінці кредитної угоди за третім розділом максимально можливий рейтинг позичальника становить 8 балів.

По показнику «Наявність у позичальника кредитної історії»:

- позитивна = 8 балів;
- раніше не кредитувався або позитивна в інших банках = 5 балів;
- допускались прострочення платежів = 0 балів.

За показниками розділу 4 позичальник максимально може отримати рейтинг у 10 балів.

По показнику «Наявність забезпечення по запитуваному кредиту»:

- забезпечений = 7 балів;
- частково забезпечений = 3 бали;
- бланковий = 1 бал.

По показнику «Строк кредитування» встановлення балів дещо складніше, адже воно передбачає ще й орієнтацію на показник «Наявність забезпечення по запитуваному кредиту».

Якщо «Наявність забезпечення по запитуваному кредиту» = «забезпечений», тоді бал по строку кредитування:

- до 1 року = 3 балів;
- від 1 року до 3 років = 3 бали;
- від 3 років = 3 бали.

Якщо «наявність забезпечення по запитуваному кредиту» = «частково забезпечений», тоді бал по строку кредитування:

- до 1 року = 3 бали;
- від 1 року до 3 років = 3 бали;
- від 3 років = 1 бал.

Якщо «наявність забезпечення по запитуваному кредиту» = «бланковий», тоді бал по строку кредитування:

- до 1 року = 3 бали;
- від 1 року до 3 років = 2 бали;
- від 3 років = 0 балів.

Загальна рейтингова оцінка (загальний бал) знаходиться шляхом сумування балів по кожному з показників. Максимальний загальний бал складає 45 балів.

Наступним кроком є визначення фінансового класу позичальника. Фінансовий клас може бути: «А», «Б», «В», «Г», «Д». Шкала для визначення класу позичальника приведена у таблиці 3.7.

Таблиця 3.7 – Шкала для визначення класу позичальника – фізичної особи

Рейтингова оцінка	Клас фінансового стану позичальника
[28; 45]	А
[16; 28)	Б
[13; 15)	В
[5; 13)	Г
[0; 5)	Д

3.3 Інтерфейс користувача та інструкція по використанню

Одним з найбільш поширених підходів, які використовують при розробці клієнтських прикладень для взаємодії з реляційними базами даних під ОС Windows, є використання інструментарію Visual Studio корпорації Microsoft. MS

Visual Studio – це інтегроване середовище розробки (Integrated Development Environment (IDE)) для створення, документування, запуску та коректування програм, написаних на сучасних мовах створення програмних систем. Це потужний інструмент професійної розробки складних додатків, один з найкращих у світі.

Перед нами стоїть задача протестувати можливості застосування об'єктів технології ADO (Active Data Objects) для відображення даних, які зберігаються в персональній реляційній базі даних, а також можливості використання технології ODBC (Open Database Connectivity) для підключення клієнтського додатку до різних джерел даних без необхідності корегувань програмного коду та перекомпіляції.

Для рішення цієї задачі створимо багатодокументний інтерфейс користувача – MDI (Multiple Document Interface). Windows-додатки типу MDI можуть підтримувати роботу водночас з декількома документами на відміну від другого типу – SDI (Single Document Interface), які можуть функціонувати тільки з одним документом. Таким чином, для розробки програми будемо використовувати багатодокументний інтерфейс MDI.

В MDI-додатках одна з форм об'являється як головна, яка вміщує елементи управління для виклику дочірніх форм. Функціональність додатку реалізується в дочірніх формах. Більшість Windows-додатків мають головне меню, яке представляє собою ієрархічну структуру виконуючих функцій та команд. Для створення головного меню використовується елемент управління MenuStrip, розміщений на Панелі елементів.

Для обмеження доступу до автоматизованої системи був реалізований механізм авторизації та аутентифікації, вікно для введення прізвища та пароль користувача, а також шляху до бази даних представлено на рисунку 3.9. Інтерфейс користувача має вигляд, який представлено на рисунку 3.10.

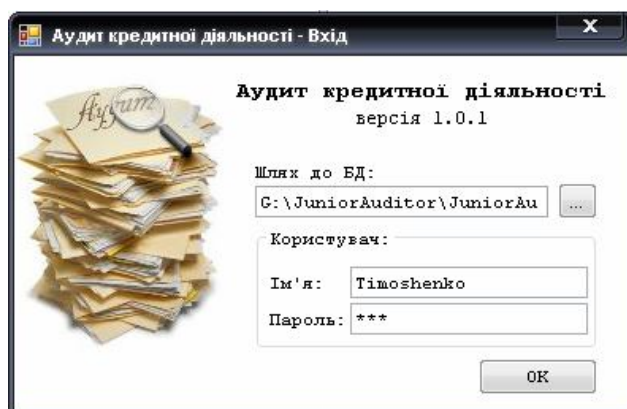


Рисунок 3.9 – Вікно входу у систему

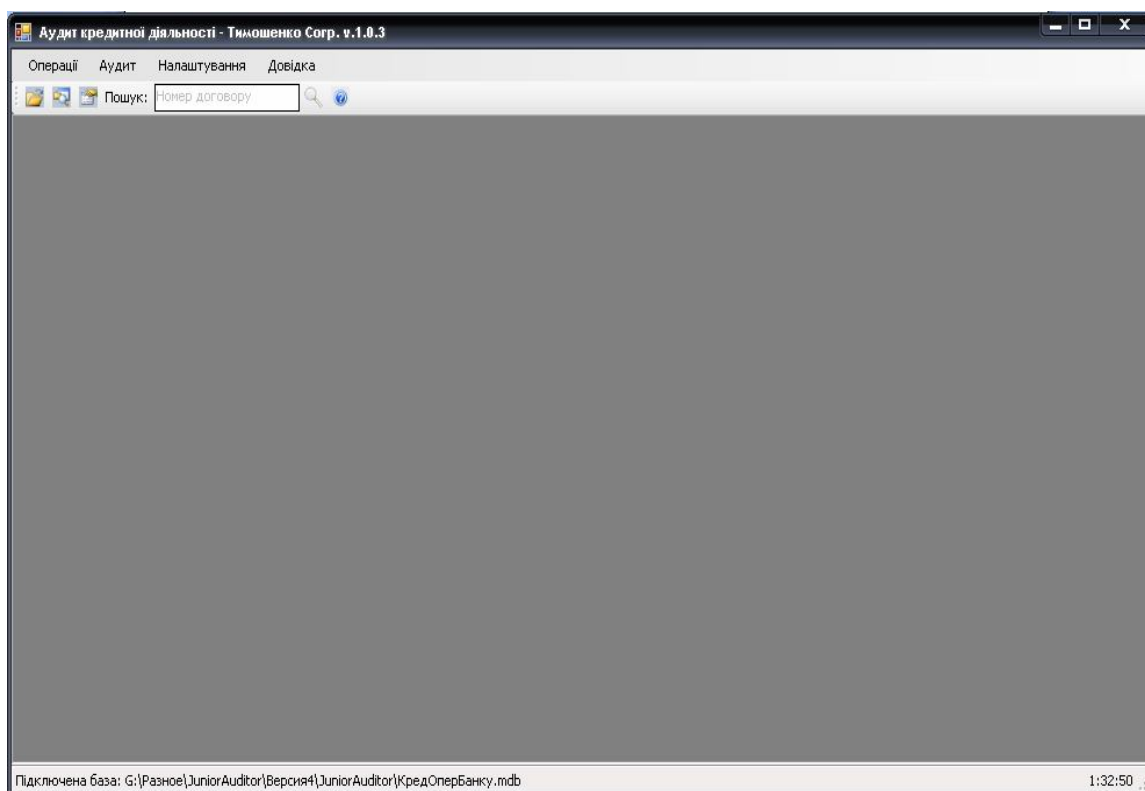


Рисунок 3.10 – Вікно інтерфейсу користувача

Тепер необхідно створити віконні форми, які будуть відкриватись при виборі відповідних пунктів меню. Розглянемо детальніше кожен пункт головного меню.

Пункт меню «Операції» (рис. 3.11) складається з наступних підпунктів:

- аналіз кредитів;
- шлях до бази даних;
- вихід.

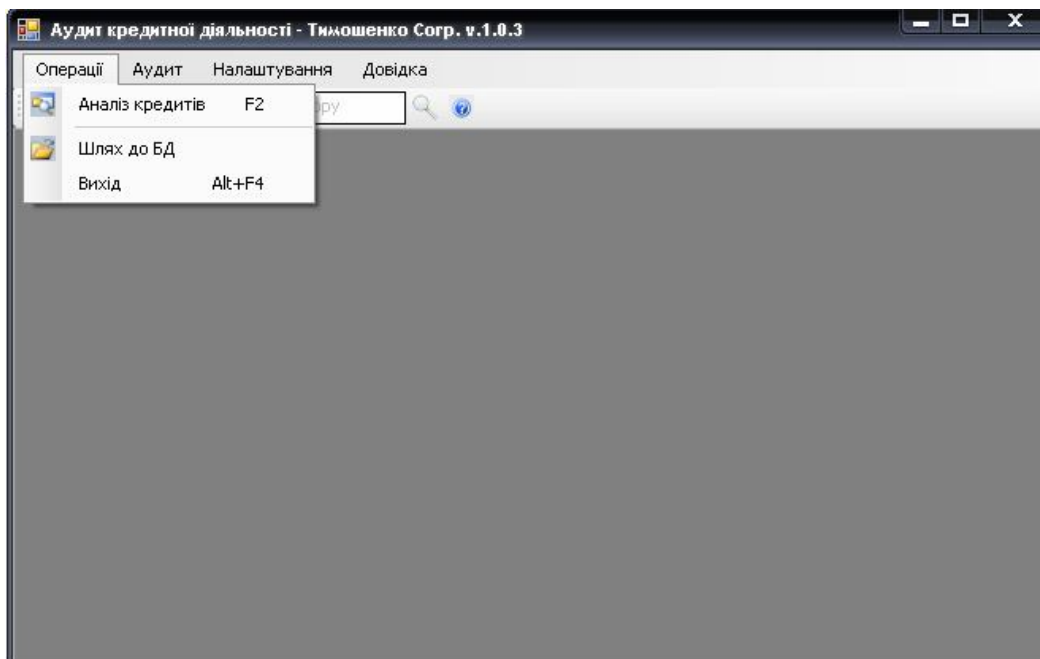


Рисунок 3.11 – Пункт меню «Операції»

Пункт меню «Аудит» (рис. 3.12) складається з наступних підпунктів:

- журнал перевірок;
- статистика;
- константи.

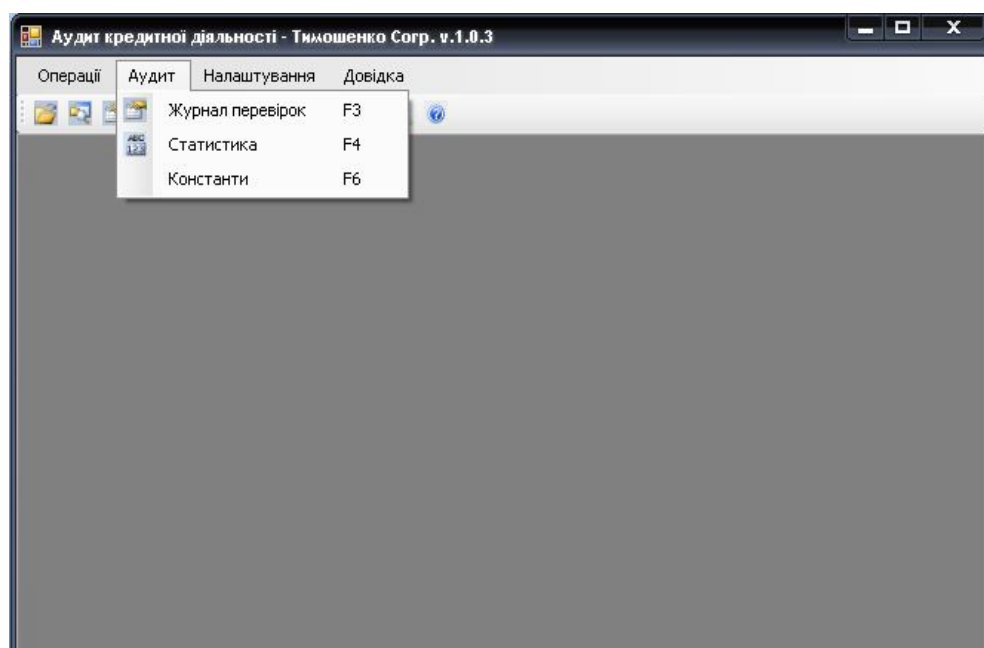


Рисунок 3.12 – Пункт меню «Аудит»

Пункт меню «Налаштування» (рис. 3.13) складається з наступних підпунктів:

- розташування вікон (горизонтально, вертикально, каскад);
- строка стану.

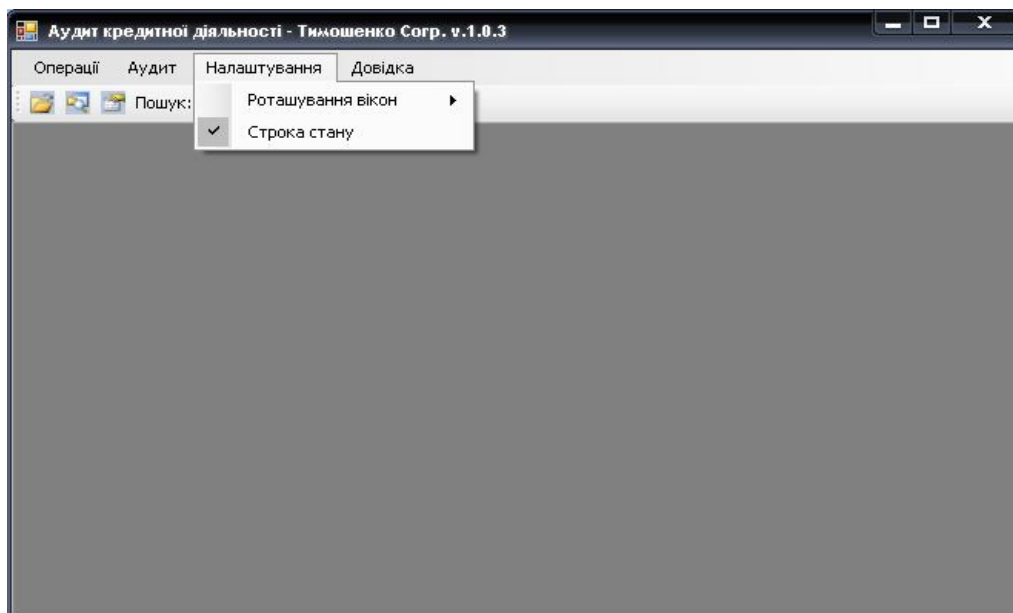


Рисунок 3.13 – Пункт меню «Налаштування»

Пункт меню «Довідка» (рис. 3.14) складається з наступних підпунктів:

- про програму.

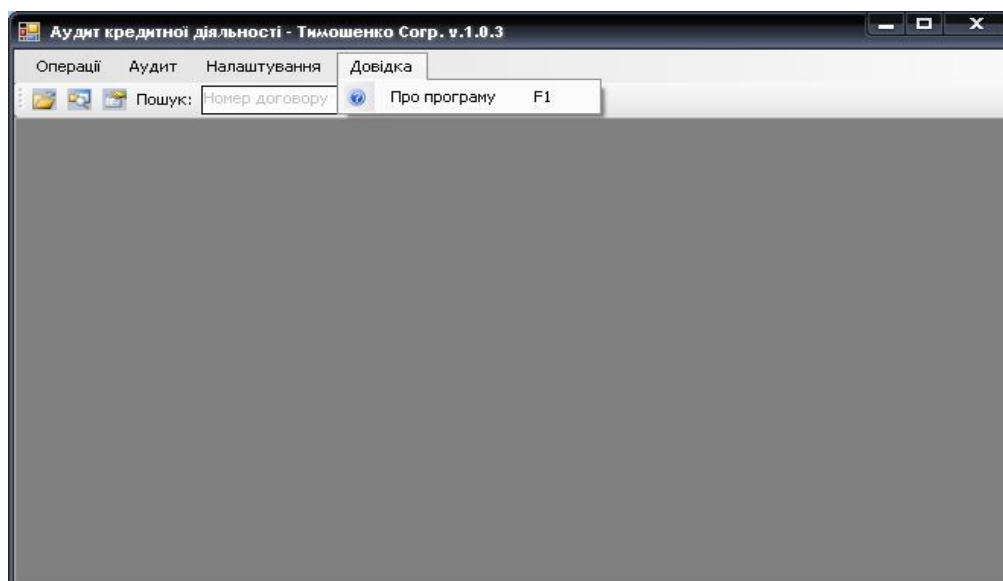


Рисунок 3.14 – Пункт меню «Довідка»

При виконанні операції Головне меню → Операції → Аналіз кредитів відкриється віконна форма, яка представлена на рисунку 3.15.

Номер	Вид кредиту	Сума кредиту	Валюта	Відсоток	Строк	Дата отримання
1000000001	Готівкою	200000	UAH	0,17	10	21.02.2002 0:00:...
1000000002	Споживчий	2500	UAH	0,15	14	20.05.2010 0:00:...
1000000003	На авто	150000	UAH	0,18	36	02.05.2010 0:00:...
1000000004	Готівкою	5000	UAH	0,1	12	29.12.2010 0:00:...
1000000005	Споживчий	12000	UAH	0,17	24	13.07.2010 0:00:...
1000000006	На авто	89000	UAH	0,22	48	06.05.2009 0:00:...
1000000007	Споживчий	5500	UAH	0,22	18	01.01.2011 0:00:...

Рисунок 3.15 – Вікно переліку кредитних договорів

Наведена вища форма відображає коротку інформацію про кредитні договори, а саме: номер кредитного договору, вид кредиту, сума кредиту, валюта, процентна ставка, строк надання кредиту, дата отримання кредиту клієнтом.

Так як діяльність аудитора основана на статистичних вибірках, то на формі, що представлена на рисунку 3.15 реалізована можливість формування вибірок на основі фільтрування даних за номером договору, валютою кредиту, видом кредиту, строком кредиту, сумою кредиту та за датою надання кредиту.

Формування вибірки за строком (середньострокові) представлено на рисунку 3.16.

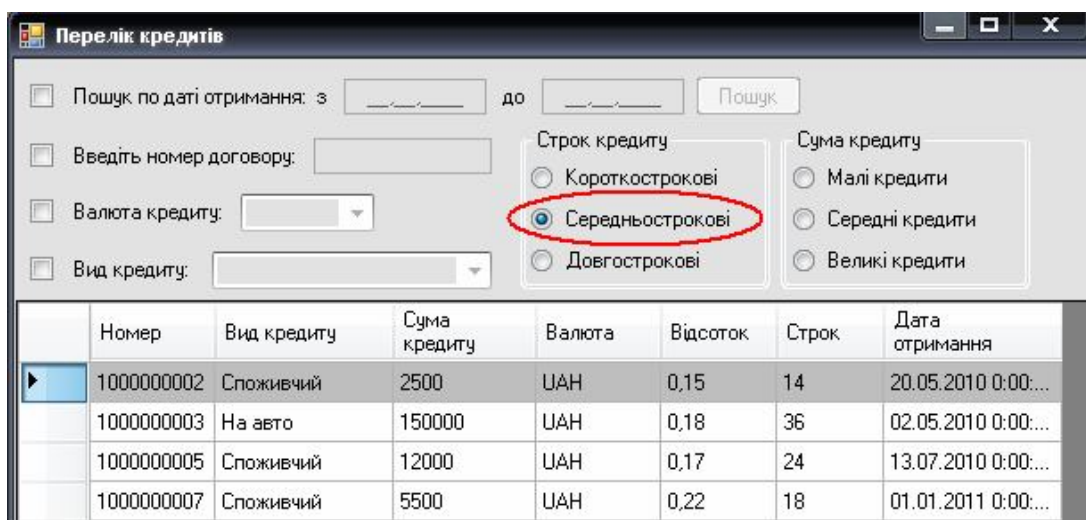


Рисунок 3.16 – Формування статистичної вибірки

Як видно з наведеного вище рисунку до середньострокових кредитів відноситься чотири кредитних договорів.

Для відображення детальної інформації по відповідному кредитному договору реалізовано 2 способи:

- за допомогою подвійного натискання лівої кнопки миші на відповідний кредитний договір;
- за допомогою контекстного меню, після виділення відповідного кредитного договору.

Наведенні вище способи представлені на рисунку 3.17.

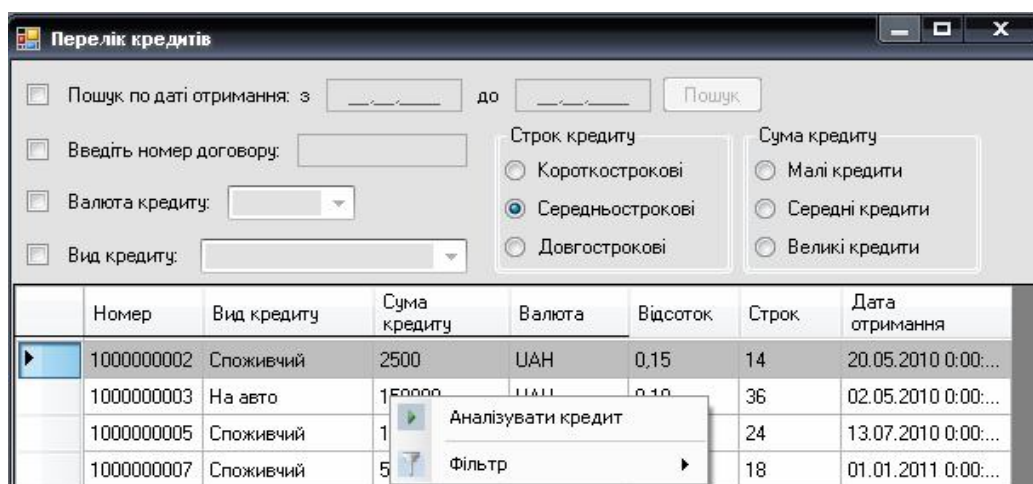


Рисунок 3.17 – Спосіб відображення детальної інформації по кредитному договору

Після натискання «Аналізувати кредит» відкривається віконна форма «Кредитна інформація», яка складається з вкладок, що наведені нижче.

Основні дані – складається з таких полів як: прізвище, ім'я, по батькові позичальника, дата народження, вік, стать, ідентифікаційний номер та паспортні дані. Дана вкладка представлена на рисунку 3.18.

Кредитна інформація	
Основні дані	Адреса
Соціальний статус	Місце роботи
Параметри кредиту	Забезпечення к
Прізвище:	Маркевич
Ім'я:	Валентина
По батькові:	Іванівна
Дата народження:	05.02.1950 0:00:00
Вік:	61 рр
Стать:	Жін
Ідентифікаційний номер:	87878988986
Паспортні дані	
Номер паспорту:	EE 112443
Ким виданий:	Конотопським РВ МВС у Сумській області
Аналізувати	
Фінансовий стан	
Закрити	

Рисунок 3.18 – Вкладка «Основні дані»

Поле вік на відміну від інших є розрахунковим полем і змінюється з часом.

Адреса – вміщує інформацію про фактичну адресу проживання позичальника та контактні телефони. Дана вкладка представлена на рисунку 3.19.

Кредитна інформація

Основні дані | **Адреса** | Соціальний статус | Місце роботи | Параметри кредиту | Забезпечення к

Поштова адреса

Країна: Україна

Область: Сумська

Нас. пункт: Суми

Вулиця: Петропавлівська

Дом: 30 Квартира: 15

Зв'язок

Телефон/факс: 0502125656

Мобільний телефон: 0502125656

Аналізувати Фінансовий стан Закрити

Рисунок 3.19 – Вкладка «Адреса»

Соціальний статус – дана вкладка складається з таких полів як: сімейний статус, наявність депозиту, наявність дітей, наявність кредитної історії, освіта. Дана вкладка, що представлена на рисунку 3.20, виступає основою для оцінки фінансового стану позичальника.

Кредитна інформація

Соціальний статус | Місце роботи | Параметри кредиту | Забезпечення кредиту | Сума кредиту

Соціальний статус

Сімейний стан: Одружений/заміжня

Наявність дітей: Немає

Освіта: Середня спеціальна

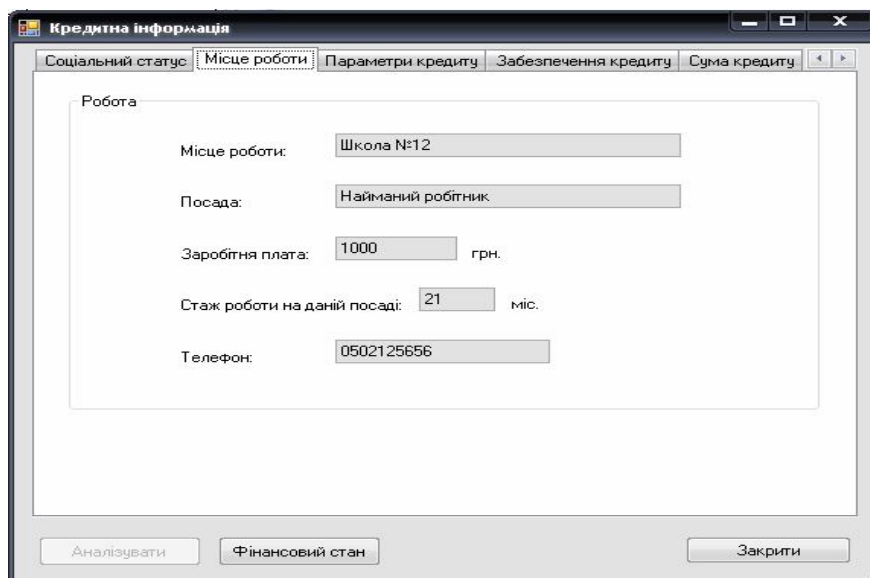
Наявність депозиту: Так

Наявність кредитної історії: Раніше не кредитувався

Аналізувати Фінансовий стан Закрити

Рисунок 3.20 – Вкладка «Соціальний статус»

Місце роботи – дана вкладка складається з таких полів як: місце роботи, посада, заробітна плата, стаж роботи та робочий телефон. Дана вкладка представлена на рисунку 3.21.



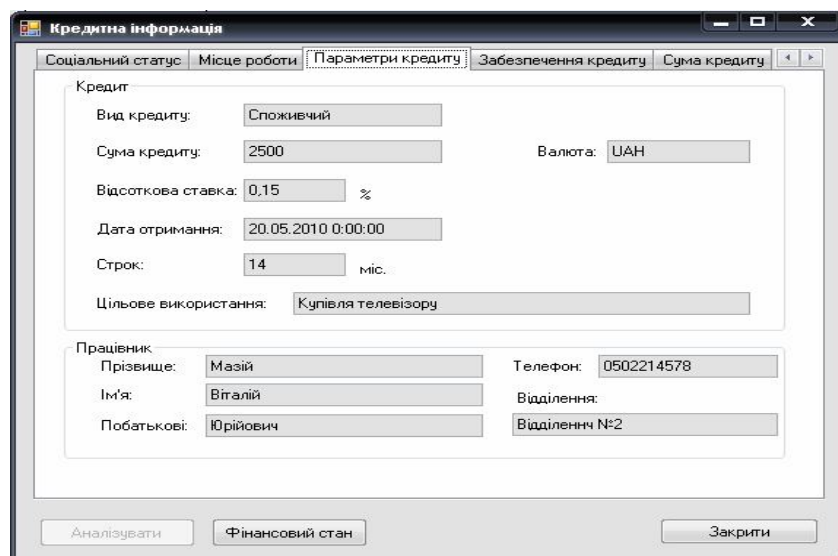
The screenshot shows a window titled 'Кредитна інформація' with several tabs: 'Соціальний статус', 'Місце роботи', 'Параметри кредиту', 'Забезпечення кредиту', and 'Сума кредиту'. The 'Місце роботи' tab is active. The form contains the following fields:

Місце роботи:	Школа №12
Посада:	Найманий робітник
Заробітна плата:	1000 грн.
Стаж роботи на даній посаді:	21 міс.
Телефон:	0502125656

At the bottom of the window, there are three buttons: 'Аналізувати', 'Фінансовий стан', and 'Закрити'.

Рисунок 3.21 – Вкладка «Місце роботи»

Параметри кредиту – дана вкладка вміщує основні характеристики кредиту, такі як: сума, строк, відсоткова ставка, цільове використання, валюта кредиту, а також інформацію про працівника, що надавав кредит. Дана вкладка представлена на рисунку 3.22.



The screenshot shows the same window with the 'Параметри кредиту' tab active. The form is divided into two sections: 'Кредит' and 'Працівник'.

Кредит:

Вид кредиту:	Споживчий		
Сума кредиту:	2500	Валюта:	UAH
Відсоткова ставка:	0,15 %		
Дата отримання:	20.05.2010 0:00:00		
Строк:	14 міс.		
Цільове використання:	Купівля телевізору		

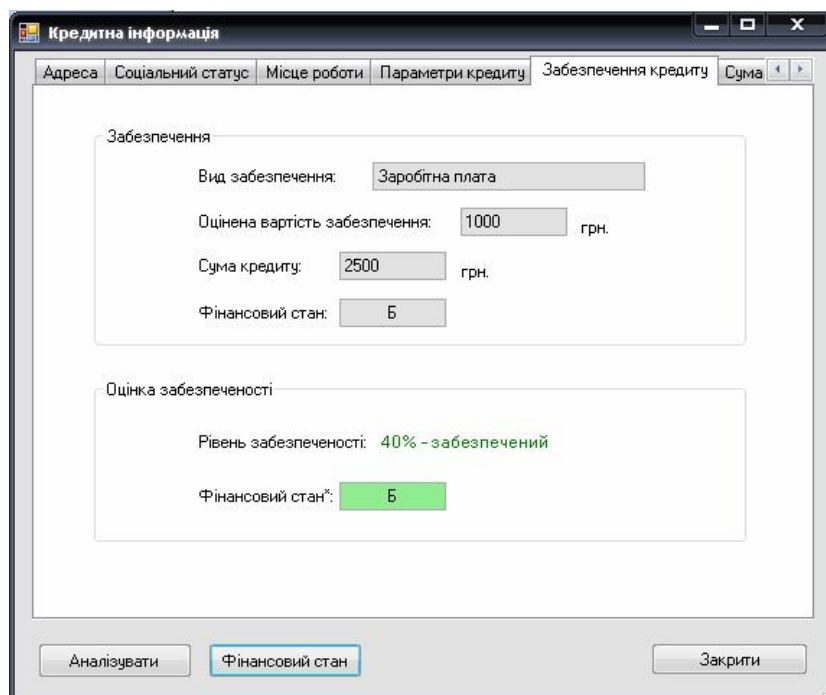
Працівник:

Прізвище:	Мазій	Телефон:	0502214578
Ім'я:	В'гальй	Відділення:	
Побатькові:	Юрійович	Відділенч №2	

At the bottom of the window, there are three buttons: 'Аналізувати', 'Фінансовий стан', and 'Закрити'.

Рисунок 3.22 – Вкладка «Параметри кредиту»

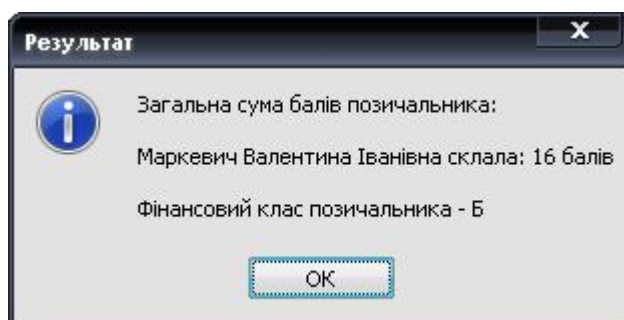
Забезпечення кредиту – дана вкладка складається з таких полів як: вид забезпечення, оцінена вартість забезпечення, фінансовий стан (визначений), а також розрахункові поля: рівень забезпечення та фактичний фінансовий стан позичальника. Дана вкладка представлена на рисунку 3.23.



The screenshot shows a window titled "Кредитна інформація" (Credit Information) with several tabs: "Адреса", "Соціальний статус", "Місце роботи", "Параметри кредиту", "Забезпечення кредиту", and "Сума". The "Забезпечення кредиту" (Credit Security) tab is active. It contains two main sections: "Забезпечення" (Security) and "Оцінка забезпеченості" (Security Assessment). In the "Забезпечення" section, the "Вид забезпечення" (Type of security) is "Заробітна плата" (Salary), "Оцінена вартість забезпечення" (Estimated value of security) is "1000" грн., "Сума кредиту" (Credit amount) is "2500" грн., and "Фінансовий стан" (Financial status) is "Б". In the "Оцінка забезпеченості" section, "Рівень забезпеченості" (Security level) is "40% - забезпечений" (40% - secured) and "Фінансовий стан*" (Financial status*) is "Б". At the bottom of the window, there are three buttons: "Аналізувати" (Analyze), "Фінансовий стан" (Financial status), and "Закрити" (Close).

Рисунок 3.23 – Вкладка «забезпечення кредиту»

Для розрахунку фінансового стану необхідно натиснути кнопку «Фінансовий стан», після чого з'явиться інформаційне вікно, що представлено на рисунку 3.24, про кількість балів та відповідний фінансовий стан у літеральному виразі від А до Д.



The screenshot shows a small window titled "Результат" (Result) with an information icon on the left. It displays the following text: "Загальна сума балів позичальника:" (Total score of the borrower:), "Маркевич Валентина Іванівна склала: 16 балів" (Markevich Valentina Ivanivna scored: 16 points), and "Фінансовий клас позичальника - Б" (Financial class of the borrower - B). There is an "ОК" button at the bottom.

Рисунок 3.24 – Інформаційне вікно оцінки фінансового стану позичальника

Сума кредиту – дана вкладка складається з полів параметрів кредиту та сукупних витрат, які необхідні для розрахунку суми щомісячної виплати та складання графіку платежів для моніторингу виплат тіла кредиту та відсотків. Дана вкладка з відповідним графіком платежів представлена на рисунку 3.25.

The screenshot shows a software window titled "Кредитна інформація" (Credit Information). It has several tabs: "Соціальний статус", "Місце роботи", "Параметри кредиту", "Забезпечення кредиту", and "Сума кредиту". The "Сума кредиту" tab is active. The window is divided into two main sections: "Дані по кредиту" (Credit Data) and "Сукупні витрати" (Total Expenses).

Дані по кредиту:

- Тип кредиту: Споживчий
- Дата надання: 20.05.2010
- Сума кредиту до сплати: 3095 грн.
- Термін користування: 14 місяців
- Погашення: щомісяця, рівними частинами
- Відсоткова ставка: 15 %

Сукупні витрати:

- Комісія за надання: 4 %
- Страхування: 120 грн.

Графік платежів:

Номер платежу	Дата платежу	Сума платежу	Сплачена сума
1	20.06.2010	221,07	221,07
2	20.07.2010	221,07	442,14
3	20.08.2010	221,07	663,21
4	20.09.2010	221,07	884,29

At the bottom of the window, there are three buttons: "Аналізувати", "Фінансовий стан", and "Закрити".

Рисунок 3.25 – Вкладка «Сума кредиту»

Вікно «Кредитна інформація» є основним для аудитора, в якому він може проводити перевірки. Перш за все аудитор повинен перевірити фінансовий стан, після чого проводити перевірку відповідно до вимоги та алгоритму.

Кожен кредитний договір перевіряється на:

- наявність записів у полях;
- вік позичальника (≥ 21);
- забезпеченість кредиту (сума кредиту повинна складати не менше 25% забезпечення);
- фінансовий стан позичальника;
- наявність прострочених кредитів.

Для проведення перевірки необхідно натиснути кнопку «Аналізувати», в результаті чого відобразиться інформаційне вікно з результатами перевірки, яке представлено на рисунку 3.26.

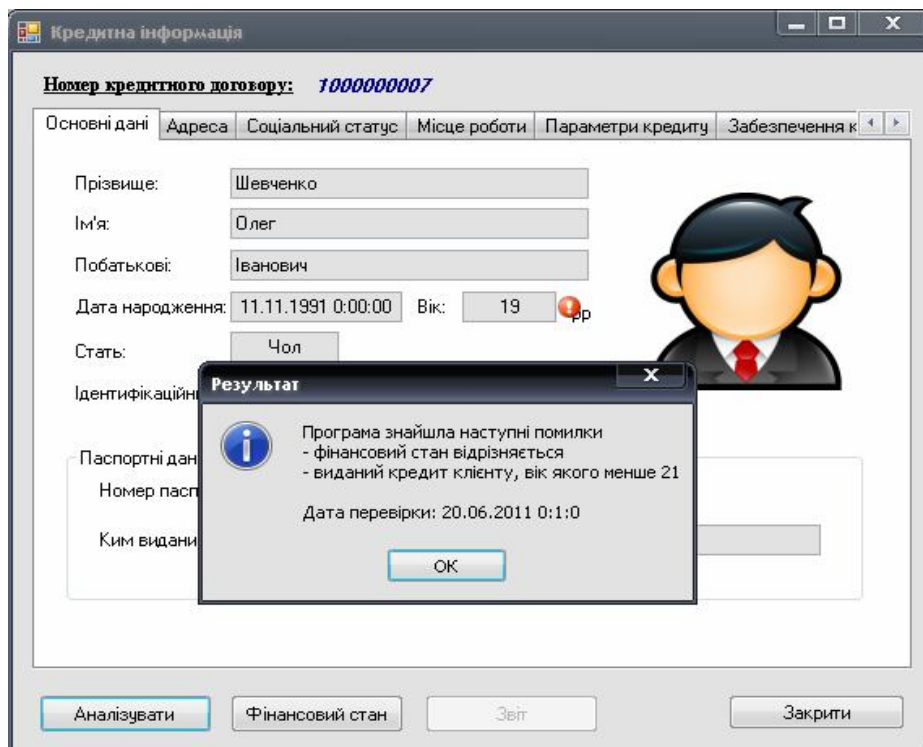


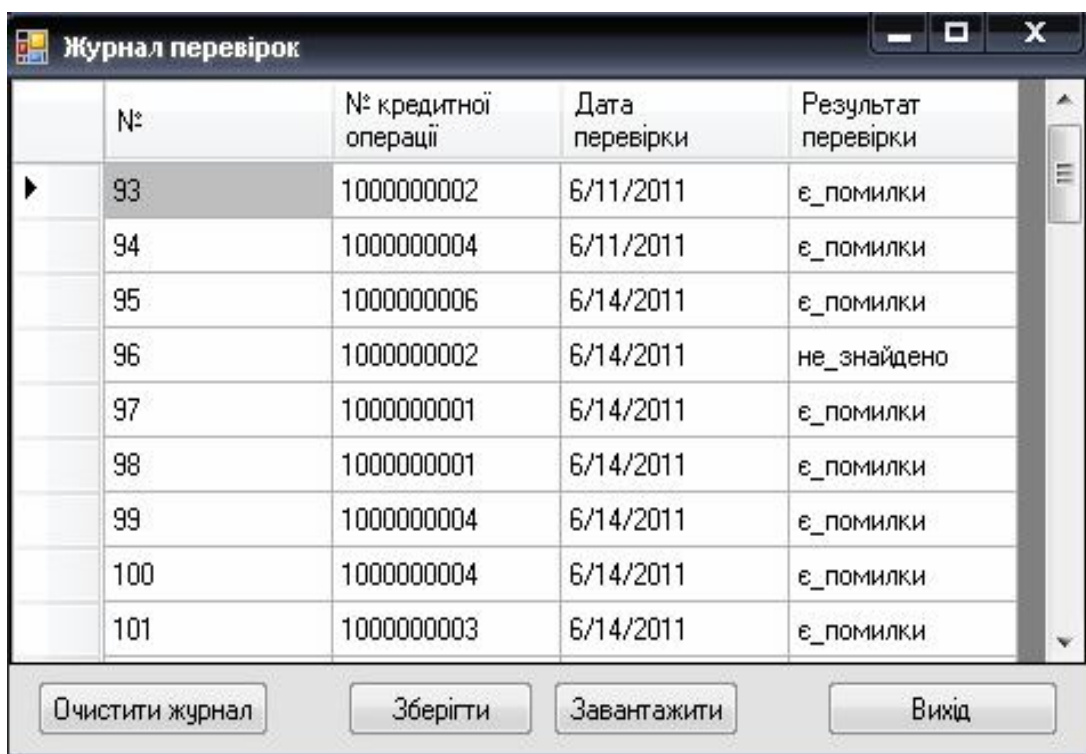
Рисунок 3.26 – Вікно відображення результатів перевірки кредитного договору

Як видно з рисунку 3.26 в результаті перевірки кредитного договору №1000000007 було виявлено наступні помилки:

- розрахований фінансовий стан відрізняється від фактичного;
- був виданий кредит клієнту, вік якого менше 21.

Також програма позначає знаком оклику в якому саме полі кредитного договору була виявлена помилка, так як це показано на рисунку 3.22. Після натискання кнопки «ОК» результати перевірки записуються у «Журнал перевірок».

Для відображення віконної форми «Журнал перевірок» необхідно виконати наступну команду: Головне меню → Аудит → Журнал перевірок. Після виконання команди відобразиться форма, яка представлена на рисунку 3.27.



№	№ кредитної операції	Дата перевірки	Результат перевірки
93	1000000002	6/11/2011	є_помилки
94	1000000004	6/11/2011	є_помилки
95	1000000006	6/14/2011	є_помилки
96	1000000002	6/14/2011	не_знайдено
97	1000000001	6/14/2011	є_помилки
98	1000000001	6/14/2011	є_помилки
99	1000000004	6/14/2011	є_помилки
100	1000000004	6/14/2011	є_помилки
101	1000000003	6/14/2011	є_помилки

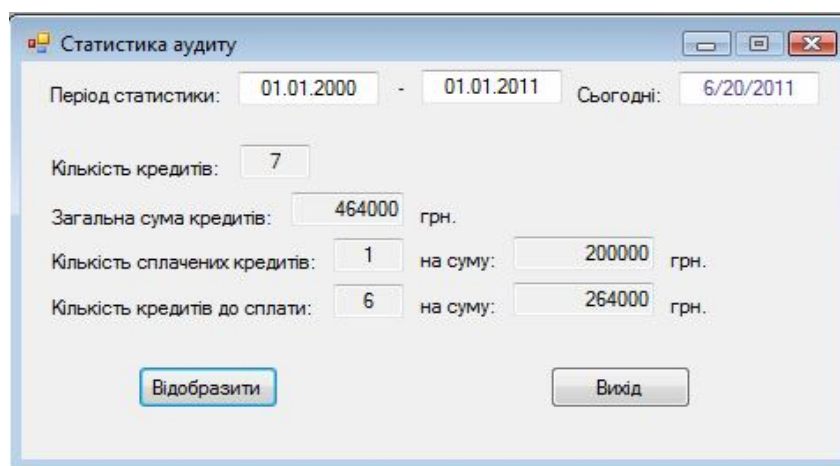
Рисунок 3.27 – Форма «Журнал перевірок»

Дана форма вміщує елемент управління DataGrid, який складається з чотирьох стовбців: номер запису у журналі, номер кредитної операції (договір), дата перевірка, результат перевірки. Також на даній формі реалізовані наступні можливості по роботі з журналом:

- видалення записів з журналу;
- зберігання журналу у файл з розширенням .txt;
- завантаження журналу з файлу з розширенням .txt.

Також реалізована можливість, за допомогою контекстного меню, виконати оновлення журналу у разі запису (видалення) перевірок.

У пункті головного меню «Аудит» можна також продивитись статистику по кредитам, що були видані банком, для цього необхідно виконати наступну команду: Головне меню → Аудит → Статистика. В результаті виконання команди з'явиться форма, що представлена на рисунку 3.28.



Статистика аудиту

Період статистики: 01.01.2000 - 01.01.2011 Сьогодні: 6/20/2011

Кількість кредитів: 7

Загальна сума кредитів: 464000 грн.

Кількість сплачених кредитів: 1 на суму: 200000 грн.

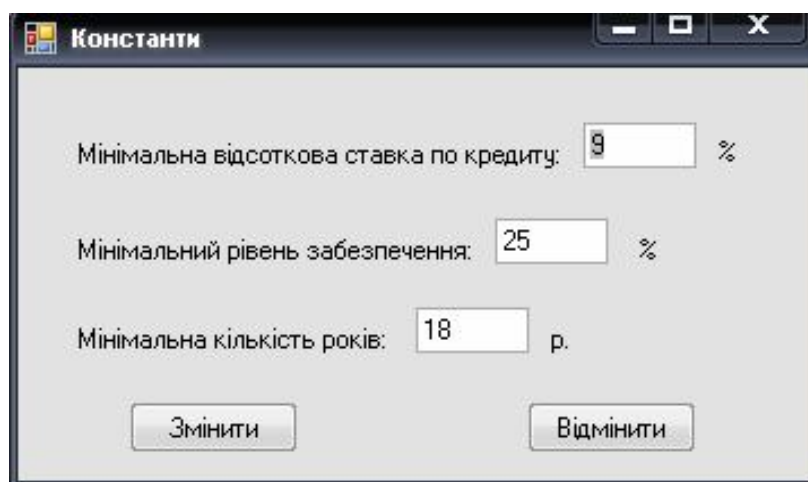
Кількість кредитів до сплати: 6 на суму: 264000 грн.

Відобразити Вихід

Рисунок 3.28 – Форма «Статистика аудиту»

В даній формі реалізована можливість відображення кількості та суми кредитів за певний проміжок часу.

У пункті головного меню «Аудит» можна встановити відповідні константи для перевірки, для цього необхідно виконати наступну команду: Головне меню → Аудит → Константи. В результаті виконання команди з'явиться форма, що представлена на рисунку 3.29.



Константи

Мінімальна відсоткова ставка по кредиту: 9 %

Мінімальний рівень забезпечення: 25 %

Мінімальна кількість років: 18 р.

Змінити Відмінити

Рисунок 3.29 – Форма «Константи»

При роботі з декількома документами в MDI-додатках зручно впорядковувати їх на екрані. Можна розподіляти форми вручну, але при роботі з великою кількістю документів це представляє певні труднощі. Для впорядкування віконних форм (горизонтально, вертикально, каскад) необхідно

виконати наступну команду: Головне меню → Налаштування → Розташування вікон. Дана можливість представлена на рисунку 3.30.

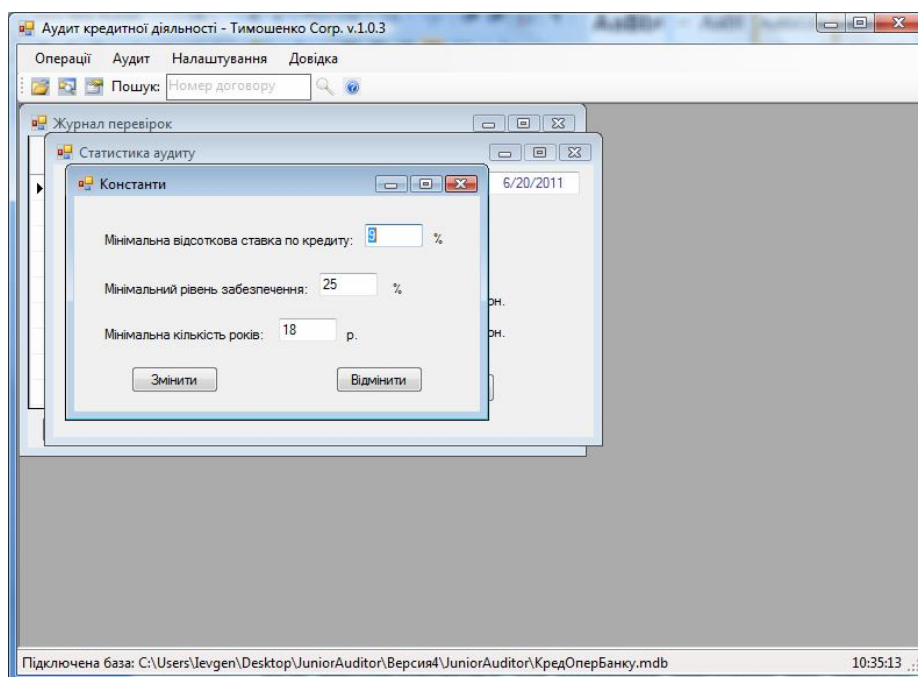


Рисунок 3.30 – Упорядкування форм «каскадом»

Поширеним компонентом Windows-додатків, який забезпечує користувачів додатковою інформацією, є строка стану. В даному проєкті вона буде використовуватись для отримання інформації про джерело даних та реалізації відображення поточного часу. Для цього розмістимо на головній формі елемент управління StatusStrip. Виконавши команду Головне меню → Налаштування → Строка стану, можна відобразити або скрити строку стану, якщо це необхідно користувачу.

На головній формі програмного додатку, що представлена на рисунку 3.10 реалізована панель інструментів, за допомогою якої можна швидко викликати наступні віконні форми:

- завантаження бази даних;
- перелік кредитів;
- журнал перевірок;
- про програму.

Також на панелі інструментів реалізована можливість пошуку кредитного договору по його номеру. Дана можливість представлена на рисунку 3.31.

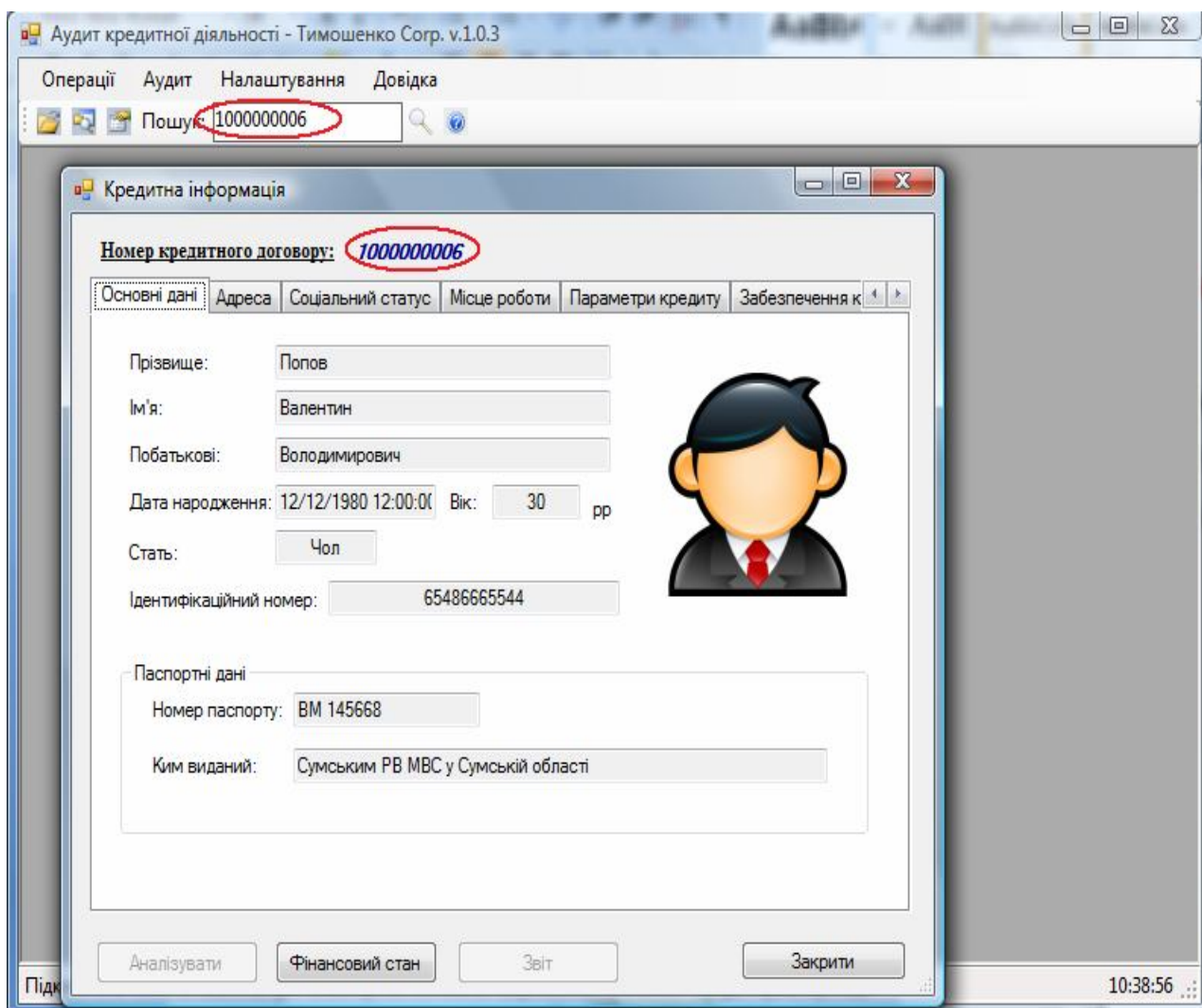


Рисунок 3.31 – Пошук кредитного договору з панелі інструментів

Також в автоматизованій системі внутрішнього аудиту банку реалізована функція перегляду інформаційного вікна про програму, виконавши команду Головне меню → Довідка → Про програму. Дане вікно представлено на рисунку 3.32.

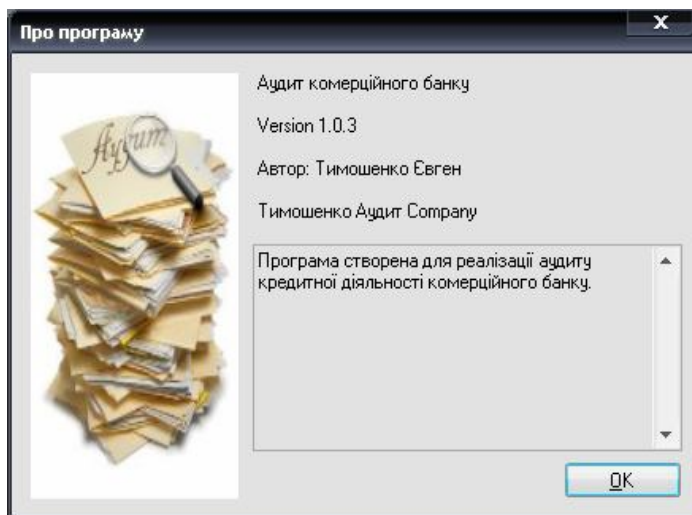


Рисунок 3.32 – Інформаційне вікно «Про програму»

Програмний код, який був використаний для реалізації графічного інтерфейсу представлений у додатку А.

3.4 Оцінка ефективності від впровадження запропонованого програмного рішення

Ефект від впровадження автоматизованої системи аудиту кредитних операцій можна поділити на обчислювальний і не обчислювальний. Обчислювальний полягає в тому, що система дозволить попередити та виявити кількість неповернених кредитів, що збільшить прибуток банку і скоротить збиток.

Можна виділити наступні джерела ефективності:

- а) економія прямих витрат праці (зменшення обслуговуючого персоналу);
- б) ріст продуктивності праці за рахунок збільшення обсягу перевірок;
- в) покращення якості перевірки;
- г) зменшення витрат на обслуговування;
- д) зменшення трудомісткості процесів обробки і використання даних;
- е) переорієнтація персоналу, вивільненого від рутинних задач обробки даних, на більш інтелектуальні види діяльності.

В середньостатистичному територіальному відділенні банку працює 1 аудитор. Заробітна плата аудитора в місяць складає 5000 грн.

В середньому на перевірку одного кредитного договору до впровадження автоматизованої системи аудитор витрачав 2 години. Після впровадження автоматизованої системи час на перевірку склав 5 хвилин (1/12 години).

Розрахуємо кількість днів, необхідних для проведення запланованої річної перевірки 1000 кредитних договорів:

$$Перевірка_{до} = 1000_{договорів} \cdot 2_{год} = 2000_{год} / 8_{год} = 250 \text{ днів};$$

$$Перевірка_{після} = 1000_{договорів} \cdot \frac{1}{12_{год}} \approx 84_{год} / 8_{год} \approx 11 \text{ днів}.$$

Розрахуємо заробітну плату аудитора за один робочий день:

$$З_{день} = 5000_{грн} / 22_{днів} \approx 228 \text{ грн}.$$

Розрахуємо витрати на заробітну плату аудитора на перевірку 1000 кредитних договорів.

$З_{до} = 250_{днів} \cdot 228_{грн} = 57000$ грн. (витрати на зарплату до впровадження за рік);

$З_{після} = 11_{днів} \cdot 228_{грн} = 2508$ грн. (витрати на зарплату після впровадження за рік).

Розрахуємо суму річної економії від впровадження системи (3.1):

$$E_p = E_{до} - E_{після}, \quad (3.1)$$

де E_p – сума річної економії, грн.;

$E_{до}$ – витрати на обробку інформації до впровадження системи, грн.;

$E_{після}$ – витрати на обробку інформації після впровадження системи, грн.

$$E_p = 57000_{грн} - 2508_{грн} = 54492 \text{ грн}.$$

Основні витрати на впровадження складають витрати на навчання працівника роботі з автоматизованою системою, на яке аудитору знадобиться 2 робочих дня. Таким чином, витрати на навчання одного аудитора становлять:

$$B_k = 5000_{\text{грн}} / 22_{\text{дня}} \cdot 2_{\text{дня}} \approx 455_{\text{грн}}.$$

Розрахуємо річний економічний ефект від впровадження системи (3.2):

$$E = E_p - B_k \cdot K_n, \quad (3.2)$$

де E – річний економічний ефект, грн.;

B_k – капітальні витрати, грн.;

K_n – нормативний коефіцієнт окупності капітальних витрат (0,33).

$$E = 54492_{\text{грн}} - (455_{\text{грн}} \cdot 0,33) = 54341,85_{\text{грн}}.$$

Розрахуємо період окупності T_0 даної підсистеми (3.3):

$$T_0 = B_k / E_p. \quad (3.3)$$

$$T_0 = 455_{\text{грн}} / 54492_{\text{грн}} = 0,0083 \text{ роки}$$

Аналізуючи дані показники можна сказати, що модуль окупиться достатньо швидко, а потім він буде приносити достатній ефект. Впровадження програми є економічно доцільним.

ВИСНОВКИ

У ході виконання магістерської дипломної роботи було визначено теоретичні основи внутрішнього аудиту в комерційних банках в цілому та кредитних операцій зокрема.

Внутрішній аудит кредитних операцій пов'язаний з обробленням великої кількості однотипної первинної інформації, а отже необхідність його автоматизації є незаперечною.

Хоча на сьогоднішній день використання автоматизованих систем в аудиті є вимогою часу, і вже склалися всі передумови для автоматизації аудиторської діяльності, аналіз ринку спеціалізованих програмних продуктів показує його обмеженість. Науковці та розробники програмного забезпечення висувають різноманітні припущення щодо причин незадовільного стану автоматизації аудиту. До них належать, наприклад, специфічний характер аудиторської діяльності, що полягає в неможливості повної формалізації процесу аудиту, та велика нормативно-правова база, яка постійно змінюється.

Дослідження показали, що на сьогоднішній день основну нішу на ринку аудиторських послуг займають російські розробки, більшість з яких не адаптована до використання в Україні.

У ході виконання магістерської роботи було проаналізовано потреби аудиторів та сформовано перелік основних вимог, яким повинен відповідати автоматизована аудиторська система. У відповідності з ними за допомогою системи прийняття рішення PRIME Decisions було оцінено основні системи на ринку програмних засобів аудиторської діяльності з метою вибору найбільш привабливої для аудиторських фірм розробки. Результати аналізу показали, що найкращою системою є IT Audit: Аудитор 2.0, та це російська розробка, яка не адаптована до використання на території України. А система AuditXP «Комплекс Аудит» використовується на українському ринку аудиторських послуг, але не відповідає всім вимогам, поставленим системам.

Таким чином, для покращення стану автоматизації аудиторської діяльності в комерційних банках необхідно створювати нові рішення, або покращувати функціональні можливості вже існуючих програмних рішень.

Виходячи з потреб аудиторської практики було визначено склад функціональних задач аудиторської фірми. Враховуючи їх, було виявлено, що автоматизована система аудиту має: проводити розрахунок фінансового стану позичальника, оцінювати рівень забезпечення, виявляти помилки у кредитних договорах та формувати звіт про проведену перевірку. За результатами дослідження було розроблено функціональну структуру автоматизованої системи аудиту.

Було виділено наступні підзадачі автоматизації аудиту кредитних операцій: аудит заповнення та ведення документів, аудит нарахування відсотків; аудит рівня забезпечення; аудит фінансового стану позичальника. Враховуючи дані задачі, було розроблено узагальнену схему інформаційного забезпечення автоматизованої системи аудиту кредитних операцій та схему інформаційних потоків задачі.

Під час виконання магістерської роботи було розроблено інформаційне забезпечення, на основі якого було створено програму «Автоматизація внутрішнього аудиту кредитних операцій».

Програма «Автоматизація внутрішнього аудиту кредитних операцій» реалізована за допомогою програмного забезпечення MS Visual Studio 2010.

Програма передбачає завантаження даних про кредитні договори з бази даних та проведення безпосередньої перевірки.

Було розроблено інструкцію користувача для супроводу даної програми, та проведено тестування програми для оцінки якості її роботи. Аналіз тестових даних показав, що програма вірно робить розрахунки, і тому може вважатися якісно виконаною, а отже придатна до використання аудиторськими фірмами.

При проведенні оцінки ефективності було виявлено, що впровадження даної програми зменшує час на перевірку кредитних договорів та збільшує якість проведення аудиту за рахунок збільшення обсягу аудиторської вибірки.

ПЕРЕЛІК ПОСИЛАНЬ

1. Аудит [Текст] : навч.-метод. посібник для самост. вивч. дисц. / В.Я. Савченко, В.О. Зотов, С.А. Кириленко та ін. – К. : КНЕУ, 2003. – 268 с. – ISBN 966-574-498-4.
2. Аудит у банках [Текст] : навч. посібник / за заг. ред. канд. екон. наук О.М. Сарахман. – К. : УАБ НБУ, 2007. – 334 с. - ISBN 978-966-368-039-2.
3. Богатова, Т. Грядет автоматизация аудита [Электронный ресурс] / Т. Богатова // PCWeek. – 2006. – №6 – Режим доступа : http://www.audit-soft.ru/about/articles/news_detail.php?ID=395PCWeek/RE. – Загл. с экрана.
4. Бутинець, Ф.Ф. Аудит [Текст] : підручник / Ф.Ф. Бутинець. – Житомир : Рута, 2002. – 672 с. – ISBN 1-5472-8722-3.
5. Внутрішній аудит [Текст] : навч. посібник / за ред. д.е.н., проф. В.О. Шевчука. – К. : ДП «Інформаційно-аналітичне агентство», 2006. – 247 с. – ISBN 966-8998-08-1.
6. Галузинський, Г.П. Перспективні технологічні засоби оброблення інформації [Текст] : навчально-методичний посібник / Г.П. Галузинський, І.В. Гордієнко. – К. : КНЕУ, 2002. – 280 с. – ISBN 966-574-359-7.
7. Гончарук, Я.А. Аудит [Текст] : навч. посіб. / Я.А. Гончарук, В.С. Рудницький. – К. : Знання, 2007. – 443 с. – ISBN 1-2321-9600-2.
8. Давидов, Г.М. Аудит [Текст] : підручник / Г.М. Давидов. – К. : Знання, 2004. – 511 с. – ISBN 966-8148-15-0.
9. Івахненко, С.В. Комп'ютерний аудит: контрольні методики і технології [Текст] : наукове видання / С.В. Івахненко. – К. : Знання, 2005. – 286 с. – ISBN 966-346-057-1.
10. Івахненко, С.В. Поняття комп'ютерного контролю та аудиту [Електронний ресурс] / С.В. Івахненко. – Режим доступу : http://www.nbuu.gov.ua/portal/Soc_Gum/Menedzhment/2009_11/ivachnenko.html. – Загл. с экрана.

11. Кіреєв, О.І. Внутрішній аудит [Текст] : навч. посібник. / О.І. Кіреєв, О.С. Любунь, В.М. Кравець, М.П. Денисенко, В.Г. Кабанов, В.М. Домрачев, В.С. Міщенко, О.К. Єременко. – К. : Центр навчальної літератури, 2006. – 220 с. – ISBN 966-364-204-1.
12. Кловез, К. Аудит процесів електронної обробки даних [Текст] / К. Кловез. – Канада : Новое знание, 1988. – 590 с. - ISBN 2-7006-1126-1.
13. Когаловский, М.Р. Энциклопедия технологий баз данных [Текст] / М.Р. Когаловский. – М. : Финансы и статистика, 2002. – 800 с. – ISBN 5-279-02276-4.
14. Комплекс Аудит: аудит по-новому! [Электронный ресурс] / Режим доступа: <http://auditxr.ru/kompleksxr>. – Загл. с экрана.
15. Косенко, Е. Реванш встроенных СУБД. Часть первая. Выбор подхода или подход к выбору? [Текст] / Е. Косенко // Компьютеры + программы (рус.). – 2002. – № 4. – с. 50 – 54.
16. Кравцова, Т.О. Теоретико-методологічні аспекти аудиту кредитних операцій банку [Текст] / Т.О. Кравцова. – К. : Аудитор України, 2006. – №12. – с.14-19 – ISBN 882-522-001-3.
17. Кузин, А.В. Базы данных. Учебное пособие для вузов [Текст] / А.В. Кузин, С.В. Левонисова. – Academia, 2008. – 315 с. – ISBN 978-5-7695-4833-8.
18. Кушнарєв, А. Проблемы автоматизации аудита [Электронный ресурс] / А. кушнарєв // Аудит сегодня. – 2007. – №4. – Режим доступа : http://www.audit-soft.ru/about/articles/news_detail.php?ID=396. – Загл. с экрана.
19. Романов, А.Н. Автоматизация аудита [Текст] / А.Н. Романов, Б.Е. Одинцов. – М. : Аудит, ЮНИТИ, 1999. – 366 с. – ISBN 5-238-0054-5.
20. Пилипенко, І.І. Стандарти аудиту та етики [Текст] : навч. посіб. / І.І. Пилипенко, О.Ю. Редько. – К. : ДП «Інформаційно-аналітичне агентство», 2007. – 277 с. – ISBN 966-8998-32-4.

21. Подольский, В.И. Компьютерный аудит [Текст] : практ. пособие / В.И. Подольский, Н.С. Щербакова, В.Л. Комисаров ; под. ред. проф. В.И. Подольского. – М. : ЮНИТИ-ДАНА, 2004. – 128 с. – ISBN 5-238-00638-1.

22. Положення про організацію внутрішнього аудиту в комерційних банках України, постанова Правління НБУ від 20.03.98 № 114. – Режим доступу : <http://zakon.rada.gov.ua/cgi-bin/laws/main.cgi?nreg=v0548500-98>. – Загл. с экрана.

23. Понікаров, В.Д. Аудит [Текст] : навчальний посібник / В.Д. Понікаров. – Х. : ВД «ІНЖЕК», 2006. – 256 с. – ISBN 263-002-001-6.

24. Скотт, Риггс и Флетчер (аудит) [Электронный ресурс] / Режим доступа : <http://srf.ru/service/it/Progs>. – Загл. с экрана.

25. Сухарева, Л. О. Організація і методика аудиту [Текст] : навч. посіб. / Л.О. Сухарева. – Донецьк : ДонНУЕТ, 2008. – 480 с. – ISBN 211-501-578-7.

26. Татарчук, М.І. Корпоративні інформаційні системи [Текст] : навч. посібник. / М.І. Татарчук. – К. : КНЕУ, 2005. – 291 с. – ISBN 966-574-717-7.

27. Хансен, Г. Базы данных: разработка и управление [Текст] / Г. Хансен, Дж. Хансен. – М. : БИНОМ, 2000. – 704 с. – ISBN 5-7989-0015-0.

28. Чемберс, Е. Компьютерный аудит [Текст] / Е. Чемберс. - Лондон : ЮНИТИ-ДАНА, 1981. – 238 с. – ISBN 4-7556-0076-5.

29. Шигун, М.М. Проблеми комп'ютеризації аудиторської діяльності [Текст] / М.М. Шигун // Збірник тез і текст. доп. на міжнар. наук.-практ. конф. з аудиту «Незалежний аудит в Україні та міжнародний досвід». – К., 2003. – с. 98-102.

30. «ІТ Аудит: Аудитор 2.0»: автоматизация внешнего и внутреннего аудита [Электронный ресурс] / Режим доступа : <http://www.audit-soft.ru>. – Загл. с экрана.

Додаток А

Лістинги програмного коду

Лістинг А.1 – Програмний код форми Main_2

```

public partial class Main_2 : Form
{
    public int balvik, balprof, balOsvita, balroduna, balDeposit, balHistory, balzabez;
    public double balOstStag, balditu, sumbal;
    public string text;

    //Тут храним обмеження
    public double MinStavka = 0.0;
    public double RivenZabez = 0.0;
    public byte MinRik = 0;

    public string is_validate() {
        string result=««;
        if (double.Parse(StavkeTextBox.Text) < MinStavka) {
            errorProvider.SetError(StavkeTextBox, «Відсоткова ставка не може бути менше « +
MinStavka*100 + « %»);
            result += «- відсоткова ставка не може бути менше « + MinStavka*100 + « %\r\n»;
        }

        string vivod = ««;
        double zabez = double.Parse(OcinenaVartistTextBox.Text) /
double.Parse(SumaTextBox.Text) * 100;
        if (zabez > RivenZabez)
        {
            vivod = «забезпечений»;
            ZabezLabel.ForeColor = Color.Green;
        }
        else
        {
            errorProvider.SetError(ZabezLabel, «Кредит не забезпечений»);
            result += «- рівень забезпечення менше « + RivenZabez + « %\r\n»;
            vivod = «не забезпечений»;
            ZabezLabel.ForeColor = Color.Red;
        }
        ZabezLabel.Text = Math.Round(zabez, 2).ToString() + «% -» + vivod;

        if (textBoxFinStanFact.Text == textBoxFinStanRozr.Text)
        {
            textBoxFinStanFact.BackColor = Color.Green;
        }

        Else
    }
}

```

Продовження лістингу А.1

```

    {
        errorProvider.SetError(textBoxFinStanFact, «Фінансовий стан відрізняється»);
        result += «- фінансовий стан відрізняється\r\n»;
        textBoxFinStanFact.BackColor = Color.LightPink;
    }

    if (textBoxFinStanRozr.Text == «Г» || textBoxFinStanRozr.Text == «Д»)
    {
        errorProvider.SetError(textBoxFinStanRozr, «Фінансовий стан позичальника
незадовільний»);
        result += «- фінансовий стан позичальника незадовільний\r\n»;
    }

    if (Convert.ToByte(textBoxVik.Text) < MinRik)
    {
        errorProvider.SetError(textBoxVik, «Виданий кредит клієнту, вік якого менше « +
MinRik);
        result += «- виданий кредит клієнту, вік якого менше « + MinRik + «\r\n»;
    }

    foreach (TabPage t in tabControl1.TabPages)
    {
        foreach (Control c in t.Controls)
        {
            if (c is TextBox)
            {
                if (c.Text == «»)
                {
                    errorProvider.SetError(c, «Пусте значення»);
                    result += «- пусте значення поля \»« + c.Tag + «\r\n»;
                }
            }
        }
    }

    return result;
}
public int creditIdd;
string IdDogovory, Vidilennya, Auditor, Inspector;
public Main_2(int creditId)
{
    creditIdd=creditId;
    InitializeComponent();

    labelCreditid.Text = creditId.ToString();
    labelCreditid.ForeColor = Color.DarkBlue;

    ValidateButton.Enabled = false;
    ReportButton.Enabled = false;
    try
    {

```

Продовження лістингу А.1

```

// create an open the connection
OleDbConnection conn = new
OleDbConnection(JuniorAuditor.Properties.Settings.Default.DBpath);
conn.Open();

// create the DataSet
DataSet ds = new DataSet();

// create the adapter and fill the DataSet
OleDbDataAdapter adapter =
new OleDbDataAdapter(«Select * from Кредитна_операція WHERE
КредитнийДоговірІдн=« + creditId.ToString(), conn);
adapter.Fill(ds);

DataTable dt = ds.Tables[0];
DataRow dr = dt.Rows[0];
CreditTypeTextBox.Text = dr[«Вид кредиту»].ToString();
CreditType2TextBox.Text = dr[«Вид кредиту»].ToString();
SumaTextBox.Text = dr[«Сума кредиту»].ToString();
ValutaTextBox.Text = dr[«Валюта»].ToString();
StavkeTextBox.Text = (double.Parse(dr[«Відсоток по
кредиту»].ToString())*100).ToString();
CreditVidsotokTextBox.Text = (double.Parse(dr[«Відсоток по кредиту»].ToString()) *
100).ToString();
DOTrimannyaTextBox.Text = dr[«Дата отримання»].ToString();
CreditDateNadanyaTextBox.Text = dr[«Дата отримання»].ToString().Remove(9, 8);
StrokTextBox.Text = dr[«Строк»].ToString();
CreditTerminTextBox.Text = dr[«Строк»].ToString();
VikoristannyaTextBox.Text = dr[«Цільове використання»].ToString();
SumaCreditTextBox.Text = dr[«Сума кредиту»].ToString();
CreditKomisiyaTextBox.Text = dr[«Комісія»].ToString();
CreditStrahuvannyaTextBox.Text = dr[«Страхування»].ToString();
CreditSumaDoSplatitTextBox.Text = (double.Parse(SumaCreditTextBox.Text) +
double.Parse(SumaCreditTextBox.Text) * double.Parse(StavkeTextBox.Text) +
double.Parse(SumaCreditTextBox.Text) * double.Parse(CreditKomisiyaTextBox.Text) / 100 +
double.Parse(CreditStrahuvannyaTextBox.Text)).ToString();

double permonth = double.Parse(CreditSumaDoSplatitTextBox.Text) /
double.Parse(CreditTerminTextBox.Text);
for (int i = 0; i < Int16.Parse(CreditTerminTextBox.Text); i++)
{
RozrahunokDataGridView.Rows.Add();
RozrahunokDataGridView.Rows[i].Cells[0].Value = (i + 1).ToString();
RozrahunokDataGridView.Rows[i].Cells[1].Value =
GetDate(DOTrimannyaTextBox.Text, i + 1);
RozrahunokDataGridView.Rows[i].Cells[2].Value = Math.Round(permonth, 2);
RozrahunokDataGridView.Rows[i].Cells[3].Value = Math.Round(GetSum(permonth, i
+ 1), 2);
}

```

Продовження лістингу А.1

```

        if (DateTime.Parse(RozrahunokDataGridView.Rows[i].Cells[1].Value.ToString()) <
DateTime.Now.Date)
        {
            ColoriseDatagridView(RozrahunokDataGridView, Color.LightGreen, i);
        }
        else
        {
            ColoriseDatagridView(RozrahunokDataGridView, Color.YellowGreen, i);
        }
    }

    DataSet ds1 = new DataSet();
    OleDbDataAdapter adapter1 =
        new OleDbDataAdapter(«Select * from Працівники WHERE ПрацівникІдн=« +
dr[«ПрацівникІдн»].ToString(), conn);
    adapter1.Fill(ds1);
    DataTable dt1 = ds1.Tables[0];
    DataRow dr1 = dt1.Rows[0];
    PrSurnameTextBox.Text = dr1[«Прізвище»].ToString();
    PrNameTextBox.Text = dr1[«Ім'я»].ToString();
    PrLastNameTextBox.Text = dr1[«Побатькові»].ToString();
    PrPhonetextBox.Text = dr1[«НомерТелефону»].ToString();

    DataSet ds11 = new DataSet();
    OleDbDataAdapter adapter11 =
        new OleDbDataAdapter(«Select * from Відділення WHERE ВідділенняІдн=« +
dr1[«ВідділенняІдн»].ToString(), conn);
    adapter11.Fill(ds11);
    DataTable dt11 = ds11.Tables[0];
    DataRow dr11 = dt11.Rows[0];
    PrViddilTextBox.Text = dr11[«Назва відділення»].ToString();

    ds11 = new DataSet();
    adapter11 =
        new OleDbDataAdapter(«Select * from Клієнти WHERE КлієнтІдн=« +
dr[«КлієнтІдн»].ToString(), conn);
    adapter11.Fill(ds11);
    dt11 = ds11.Tables[0];
    dr11 = dt11.Rows[0];
    SurnameTextBox.Text = dr11[«Прізвище»].ToString();
    NameTextBox.Text = dr11[«Імя»].ToString();
    LastNameTextBox.Text = dr11[«Побатькові»].ToString();
    DOBTextBox.Text = dr11[«Дата народження»].ToString();
    GenderTextBox.Text = dr11[«Стать»].ToString();
    IdnNumberTextBox.Text = dr11[«Ідентифікаційний номер»].ToString();
    PassportTextBox.Text = dr11[«Серія паспорту»].ToString();
    PassportVidaniyTextBox.Text = dr11[«Ким виданий»].ToString();
    textBoxStagRobotu.Text = dr11[«Стаж роботи»].ToString();
    textBoxFinStanFact.Text = dr11[«Фінансовий стан»].ToString();

```


Продовження лістингу А.1

```

DateTime dateValue;
string dateString = DOBTextBox.Text;
try
{
    dateValue = DateTime.Parse(dateString);
    textboxVik.Text = CalculateAge(dateValue).ToString();
    textboxVik.ForeColor = Color.Silver;
}
catch (FormatException)
{
    MessageBox.Show(«Помилка розрахунку віку!!!», «Результат»,
    MessageBoxButtons.OK);
}

ds1 = new DataSet();
adapter1 =
    new OleDbDataAdapter(«Select * from Адреси WHERE КлієнтІдн=« +
    dr[«КлієнтІдн»].ToString(), conn);
adapter1.Fill(ds1);
dt1 = ds1.Tables[0];
dr1 = dt1.Rows[0];
CountryTextBox.Text = dr1[«Країна»].ToString();
RegionTextBox.Text = dr1[«Область»].ToString();
TownTextBox.Text = dr1[«Населений пункт»].ToString();
StreetTextBox.Text = dr1[«Вулиця»].ToString();
HouseTextBox.Text = dr1[«Будинок»].ToString();
ApparmentTextBox.Text = dr1[«Квартира»].ToString();
TelephonTextBox.Text = dr1[«Телефон»].ToString();
MobileTextBox.Text = dr1[«Телефон»].ToString();
WorkplaceTextBox.Text = dr1[«Місце роботи»].ToString();
PosadatextBox.Text = dr1[«Посада»].ToString();
SalaryTextBox.Text = dr1[«Щомісячний дохід»].ToString();
WorkPhoneTextBox.Text = dr1[«Телефон»].ToString();

ds1 = new DataSet();
adapter1 =
    new OleDbDataAdapter(«Select * from Соціальний_статус WHERE КлієнтІдн=« +
    dr[«КлієнтІдн»].ToString(), conn);
adapter1.Fill(ds1);
dt1 = ds1.Tables[0];
dr1 = dt1.Rows[0];
textBoxSimeynyuStan.Text = dr1[«Сімейний стан»].ToString();
textBoxNayavnistDitey.Text = dr1[«Наявність дітей»].ToString();
textBoxNayavnistDepo.Text = dr1[«Наявність депозиту»].ToString();
textBoxKredytnaIstoriya.Text = dr1[«Наявність кредитної історії»].ToString();
textBoxOsvita.Text = dr1[«Освіта»].ToString();

ds1 = new DataSet();
adapter1 =

```

Продовження лістингу А.1

```

        new OleDbDataAdapter(«Select * from Забезпечення WHERE
КредитнийДоговірІдн=« + creditId.ToString(), conn);
        adapter1.Fill(ds1);
        dt1 = ds1.Tables[0];
        dr1 = dt1.Rows[0];
        OcinenavArtistTextBox.Text = dr1[«Сума»].ToString();
        VidZabezpechenyaTextBox.Text = dr1[«Вид забезпечення»].ToString();
        conn.Close();
    }
    catch {
        MessageBox.Show(«При з'єднанні з базою виникла помилка», «Помилка»,
        MessageBoxButtons.OK, MessageBoxIcon.Error);
        this.Close();
    }
}

private void ValidateButton_Click(object sender, EventArgs e)
{
    string myInsertQuery = «»;
    OleDbConnection conn = new
OleDbConnection(JuniorAuditor.Properties.Settings.Default.DBpath);
    if (is_validate() != «»)
    {
        MessageBox.Show(«Програма знайшла наступні помилки\r\n» + is_validate() +
«\r\nДата перевірки: «+DateTime.Now.Date.ToShortDateString()+»
«+DateTime.Now.TimeOfDay.Hours + «:» + DateTime.Now.TimeOfDay.Minutes + «:» +
DateTime.Now.TimeOfDay.Seconds, «Результат»,
        MessageBoxButtons.OK,
        MessageBoxIcon.Information);
        myInsertQuery = «INSERT INTO Результат([Кредитна операція],[Дата],[Результат])
Values(« + creditId.ToString() + «', '« + DateTime.Now.Date.ToShortDateString() +
«', 'є_помилки')»»;
    }
    else {
        MessageBox.Show(«Помилку в оформленні кредиту не знайдено» + «\r\nДата
перевірки: « + DateTime.Now.Date.ToShortDateString() + « « +
DateTime.Now.TimeOfDay.Hours + «:» + DateTime.Now.TimeOfDay.Minutes + «:» +
DateTime.Now.TimeOfDay.Seconds, «Результат»,
        MessageBoxButtons.OK,
        MessageBoxIcon.Information);
        myInsertQuery = «INSERT INTO Результат([Кредитна операція],[Дата],[Результат])
Values(« + creditId.ToString() + «', '« + DateTime.Now.Date.ToShortDateString() +
«', 'не_знайдено')»»;
    }
    OleDbCommand myCommand = new OleDbCommand(myInsertQuery);
    myCommand.Connection = conn;
    conn.Open();
    myCommand.ExecuteNonQuery();
    conn.Close();

    ReportButton.Enabled = true;
}

```

Продовження лістингу А.1

```

}

private void button1_Click(object sender, EventArgs e)
{
    this.Close();
}

private double GetSum(double summa, int month) {
    double result = 0;
    for (int i = 0; i < month; i++) {
        result += summa;
    }
    return result;
}

private string GetDate(string date, int i) {
    DateTime result;
    DateTime.TryParse(date, out result);
    result=result.AddMonths(i);
    return result.ToShortDateString();
}

private void ColoriseDatagridView(DataGridView dtgv, Color cl, int RowsCount)
{
    for (int i = 0; i < dtgv.Rows[0].Cells.Count; i++)
    {
        RozrahunokDataGridView.Rows[RowsCount].Cells[i].Style.BackColor = cl;
    }
}

public static int CalculateAge(DateTime Birthdate)
{
    int YearPassed = DateTime.Now.Year - Birthdate.Year;
    if (DateTime.Now.Month < Birthdate.Month || (DateTime.Now.Month == Birthdate.Month
    && DateTime.Now.Day < Birthdate.Day))
    {
        YearPassed--;
    }
    return YearPassed;
}

private void Main_2_FormClosing(object sender, FormClosingEventArgs e)
{
    if (MessageBox.Show(«Звершити аналіз?», «Кредитна операція»,
    MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.No) e.Cancel = true;
}

private void buttonFinStan_Click(object sender, EventArgs e)
{

```

Продовження лістингу А.1

```

    if (textBoxVik.Text == «» || textBoxStagRobotu.Text == «» || PosadatextBox.Text == «» ||
    textBoxOsvita.Text == «» || textBoxSimeynyuStan.Text == «» || textBoxNayavnistDitey.Text == «»
    || textBoxNayavnistDepo.Text == «» || textBoxKredytnaIstoriya.Text == «»)
    {
        MessageBox.Show(«Відсутні необхідні дані для оцінки фінансового стану
    позичальника!», «Помилка», MessageBoxButtons.OK, MessageBoxIcon.Error);
    }

    else
    {

        // VIK

    23) if (Convert.ToInt32(textBoxVik.Text) >= 18 && Convert.ToInt32(textBoxVik.Text) <=
        {
            balvik = 0;
        }
        if (Convert.ToInt32(textBoxVik.Text) > 23 && Convert.ToInt32(textBoxVik.Text) <= 45)
        {
            balvik = 2;
        }
        else
        {
            balvik = 1;
        }

        //Stag Robotu

        if (Convert.ToInt32(textBoxStagRobotu.Text) < 6)
        {
            balOstStag = 0;
        }
        if (Convert.ToInt32(textBoxStagRobotu.Text) >= 6 &&
    Convert.ToInt32(textBoxStagRobotu.Text) <= 12)
        {
            balOstStag = 0.5;
        }
        if (Convert.ToInt32(textBoxStagRobotu.Text) > 12 &&
    Convert.ToInt32(textBoxStagRobotu.Text) <= 24)
        {
            balOstStag = 1;
        }
        if (Convert.ToInt32(textBoxStagRobotu.Text) > 24)
        {
            balOstStag = 2;
        }

        //Professia

```

Продовження лістингу А.1

```

if (PosadatextBox.Text == «Студент»)
{
    balprof = 0;
}
if (PosadatextBox.Text == «Пенсіонер»)
{
    balprof = 2;
}
if (PosadatextBox.Text == «Держслужбовець» || PosadatextBox.Text == «Керівник
підприємства» || PosadatextBox.Text == «Власник підприємства»)
{
    balprof = 4;
}
else
{
    balprof = 3;
}

//Osvita

if (textBoxOsvita.Text == «Середня спеціальна»)
{
    balOsvita = 1;
}
if (textBoxOsvita.Text == «Вища»)
{
    balOsvita = 2;
}
else
{
    balOsvita = 0;
}

//Roduna

if (textBoxSimeynyuStan.Text == «Одружений/заміжня»)
{
    balroduna = 2;
}
if (textBoxSimeynyuStan.Text == «Вдівець(вдова)»)
{
    balroduna = 1;
}
else
{
    balroduna = 0;
}

//DITY

```

Продовження лістингу А.1

```

if (textBoxNayavnistDitey.Text == «3»)
{
    balditu = 0.5;
}
if (textBoxNayavnistDitey.Text == «1» || textBoxNayavnistDitey.Text == «2»)
{
    balditu = 1;
}
else
{
    balditu = 0;
}

//DEPOSIT

if (textBoxNayavnistDepo.Text == «Hi»)
{
    balDeposit = 0;
}
else
{
    balDeposit = 1;
}

// KREDUTNA ISTORIYA

if (textBoxKredytnaIstoriya.Text == «Позитивна»)
{
    balHistory = 8;
}
if (textBoxKredytnaIstoriya.Text == «Мав прострочки за платежами»)
{
    balHistory = 0;
}
else
{
    balHistory = 5;
}

// ZABEZPECHENNYA

double zabez = double.Parse(OcinenaVartistTextBox.Text) /
double.Parse(SumaTextBox.Text) * 100;
if (zabez > RivenZabez)
{
    balzabez = 5;
}
else
{
    balzabez = 0;
}

```

Продовження лістингу А.1

```

    }

    sumbal = balvik + balprof + balOsvita + balroduna + balDeposit + balHistory + balzabez
+ balOstStag + balditu;

    if (sumbal >= 0 && sumbal < 5)
    {
        text = «Д»;
    }
    if (sumbal >= 5 && sumbal < 10)
    {
        text = «Г»;
    }
    if (sumbal >= 10 && sumbal < 16)
    {
        text = «В»;
    }
    if (sumbal >= 16 && sumbal < 21)
    {
        text = «Б»;
    }
    if (sumbal >= 21 && sumbal <= 27)
    {
        text = «А»;
    }

    MessageBox.Show(«Загальна сума балів позичальника:\r\n « + «\r\n» +
SurnameTextBox.Text + « « + NameTextBox.Text + « « + LastNameTextBox.Text + « склала: « +
sumbal + « балів» + «\r\n» + «\r\n» + «Фінансовий клас позичальника - « + text, «Результат»,
MessageBoxButtons.OK, MessageBoxIcon.Information);
    textBoxFinStanRozr.Text = text;

    if (textBoxFinStanRozr.Text == «А»)
    {
        textBoxFinStanRozr.BackColor = Color.Green;
    }

    if (textBoxFinStanRozr.Text == «Б»)
    {
        textBoxFinStanRozr.BackColor = Color.LightGreen;
    }

    if (textBoxFinStanRozr.Text == «В»)
    {
        textBoxFinStanRozr.BackColor = Color.YellowGreen;
    }

    if (textBoxFinStanRozr.Text == «Г»)
    {
        textBoxFinStanRozr.BackColor = Color.Pink;
    }

```

Продовження лістингу А.1

```

    }

    if (textBoxFinStanRozr.Text == «Д»)
    {
        textBoxFinStanRozr.BackColor = Color.Red;
    }

    ValidateButton.Enabled = true;
}
}

private void button2_Click(object sender, EventArgs e)
{
    IdDogovory = this.labelCreditid.Text;
    Vidilennya = this.PrViddilTextBox.Text;
    Inspector = this.PrSurnameTextBox.Text;

    ReportForm reportform = new ReportForm(IdDogovory, Vidilennya, Inspector,
is_validate());
    reportform.Show();
}

//Метод, получающий из базы данных значения ограничений
public void PoluchitConstanti()
{
    // create an open the connection
    OleDbConnection conn = new
OleDbConnection(JuniorAuditor.Properties.Settings.Default.DBpath);
    conn.Open();

    // create the DataSet
    DataSet ds = new DataSet();

    // create the adapter and fill the DataSet
    OleDbDataAdapter adapter =
    new OleDbDataAdapter(«Select * from Константи«, conn);
    adapter.Fill(ds);

    DataTable dt = ds.Tables[0];
    DataRow dr = dt.Rows[0];
    MinStavka = (double.Parse(dr[«Кредитна ставка»].ToString())/100);
    RivenZabez = Convert.ToDouble(dr[«Рівень забезпечення»]);
    MinRik = Convert.ToByte(dr[«Мінімальний рік»]);
}
}

```


Лістинг А.2 – Програмний код форми LoginForm

```

public partial class LoginForm : Form
{
    public LoginForm()
    {
        InitializeComponent();
    }

    private void BrowseButton_Click(object sender, EventArgs e)
    {
        openFileDialog.Filter = «Access 2003 файл (*.mdb)|*.mdb|Усі файли (*.*)|*.»*;
        openFileDialog.Title = «Шлях до БД»;
        DialogResult result = openFileDialog.ShowDialog();
        if (result == DialogResult.OK && openFileDialog.FileName != «») {
            PathtextBox.Text = openFileDialog.FileName;
        }
    }

    private void EnterButton_Click(object sender, EventArgs e)
    {
        if (PathtextBox.Text == «»)
        {
            MessageBox.Show(«Ви не вказали шлях до БД.», «Повідомлення»,
            MessageBoxButtons.OK, MessageBoxIcon.Warning);
        }
        else if (NametextBox.Text != «Timoshenko») // Name and password can be changed :)
        {
            MessageBox.Show(«Невірно вказаний користувач.», «Повідомлення»,
            MessageBoxButtons.OK, MessageBoxIcon.Warning);
            NametextBox.Text = «»;
            PasswordtextBox.Text = «»;
        }

        else if (PasswordtextBox.Text != «111»)
        {
            MessageBox.Show(«Невірно вказаний пароль», «Повідомлення»,
            MessageBoxButtons.OK, MessageBoxIcon.Warning);
            NametextBox.Text = «»;
            PasswordtextBox.Text = «»;
        }
        else
        {
            JuniorAuditor.Properties.Settings.Default.DBpath =
            @»Provider=Microsoft.JET.OLEDB.4.0;»
            + @»data source=« + PathtextBox.Text;
            JuniorAuditor.Properties.Settings.Default.Save();
            this.Close();
        }
    }
}

```

Лістинг А.3 – Програмний код форми Form1

```

public partial class Form1 : Form
{
    public Form1()
    {
        if (JuniorAuditor.Properties.Settings.Default.DBpath == «»)
        {
            LoginForm loginf = new LoginForm();
            loginf.ShowDialog();
            if (JuniorAuditor.Properties.Settings.Default.DBpath == «»)
            {
                Environment.Exit(0);
            }
            else
            {
                InitializeComponent();
            }
        }
        else {
            InitializeComponent();
        }
        string[] temp;
        temp = JuniorAuditor.Properties.Settings.Default.DBpath.Split('=');
        ToolStripStatusLabel1.Checked = true;
        toolStripStatusLabel1.Text = «Підключена база: « + temp[2];
        toolStripStatusLabel2.Text = DateTime.Now.TimeOfDay.Hours + «:» +
        DateTime.Now.TimeOfDay.Minutes + «:» + DateTime.Now.TimeOfDay.Seconds;

        toolStripTextBox.Text = «Номер договору»;
        toolStripTextBox.ForeColor = Color.Silver;
    }

    private void Form1_FormClosing(object sender, FormClosingEventArgs e)
    {
        {
            if (MessageBox.Show(«Ви бажаєте завершити роботу?», «JuniorAuditor»,
            MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.No) e.Cancel = true;
        }
    }

    private void AnalyzeCredittoolStripMenuItem_Click(object sender, EventArgs e)
    {
        CreditsForm frm = new CreditsForm();
        frm.MdiParent = this;
        frm.Show();
    }
}

```

Продовження лістингу А.3

```

private void CascadeToolStripMenuItem_Click(object sender, EventArgs e)
{
    this.LayoutMdi(System.Windows.Forms.MdiLayout.Cascade);
}

private void HorisontToolStripMenuItem_Click(object sender, EventArgs e)
{
    this.LayoutMdi(System.Windows.Forms.MdiLayout.TileHorizontal);
}

private void VerticalToolStripMenuItem_Click(object sender, EventArgs e)
{
    this.LayoutMdi(System.Windows.Forms.MdiLayout.TileVertical);
}

private void PathToolStripMenuItem_Click_1(object sender, EventArgs e)
{
    LoginForm login = new LoginForm();
    login.ShowDialog();
}

private void ExitToolStripMenuItem_Click_1(object sender, EventArgs e)
{
    this.Close();
    Environment.Exit(0);
}

private void timer_Tick(object sender, EventArgs e)
{
    toolStripStatusLabel2.Text = DateTime.Now.TimeOfDay.Hours + «:» +
    DateTime.Now.TimeOfDay.Minutes + «:» + DateTime.Now.TimeOfDay.Seconds;
}

private void StatusStringToolStripMenuItem_Click(object sender, EventArgs e)
{
    if (statusStrip1.Visible == true)
    {
        statusStrip1.Visible = false;
        StatusStringToolStripMenuItem.Checked = false;
    }
    else
    {
        statusStrip1.Visible = true;
        StatusStringToolStripMenuItem.Checked = true;
    }
}

private void toolStripButton1_Click(object sender, EventArgs e)
{

```

Продовження лістингу А.3

```

if (toolStripTextBox.Text == «» || toolStripTextBox.Text == «Номер договору»)
{
    MessageBox.Show(«Введіть номер договору!!!», «Увага», MessageBoxButtons.OK,
    MessageBoxIcon.Information);
}

else
{
    try
    {
        Main_2 frm = new Main_2(int.Parse(toolStripTextBox.Text));
        frm.ShowDialog();
        toolStripTextBox.Text = «Номер договору»;
        toolStripTextBox.ForeColor = Color.Silver;
    }
    catch
    {
        MessageBox.Show(«Даний номер договору відсутній!!!», «Увага»,
        MessageBoxButtons.OK, MessageBoxIcon.Information);
        toolStripTextBox.Text = «Номер договору»;
        toolStripTextBox.ForeColor = Color.Silver;
    }
}

private void AboutToolStripMenuItem_Click(object sender, EventArgs e)
{
    AboutBox1 dd = new AboutBox1();
    dd.MdiParent = this;
    dd.Show();
}

private void ResultAuditToolStripMenuItem_Click(object sender, EventArgs e)
{
    ResultAu au = new ResultAu();
    au.MdiParent = this;
    au.Show();
}

private void toolStripButton2_Click(object sender, EventArgs e)
{
    LoginForm login = new LoginForm();
    login.ShowDialog();
}

private void toolStripButton3_Click(object sender, EventArgs e)
{
    CreditsForm frm = new CreditsForm();
    frm.MdiParent = this;
    frm.Show();
}

```

Продовження лістингу А.3

```

}

private void toolStripButton4_Click(object sender, EventArgs e)
{
    ResultAu au = new ResultAu();
    au.MdiParent = this;
    au.Show();
}

private void toolStripButton5_Click(object sender, EventArgs e)
{
    AboutBox1 dd = new AboutBox1();
    dd.MdiParent = this;
    dd.Show();
}

private void Form1_FormClosed(object sender, FormClosedEventArgs e)
{
    JuniorAuditor.Properties.Settings.Default.DBpath = «»;
    JuniorAuditor.Properties.Settings.Default.Save();
}

private void toolStripTextBox_Click(object sender, EventArgs e)
{
    toolStripTextBox.Text = «»;
    toolStripTextBox.ForeColor = Color.Black;
}

private void StatisticsToolStripMenuItem_Click(object sender, EventArgs e)
{
    StatisticaAU statAU = new StatisticaAU();
    statAU.MdiParent = this;
    statAU.Show();
}

private void константиToolStripMenuItem_Click(object sender, EventArgs e)
{
    ConstForm constantFrm = new ConstForm();
    constantFrm.MdiParent = this;
    constantFrm.Show();
}

```

Лістинг А.4 – Програмний код форми ReportForm

```

public partial class ReportForm : Form
{
    private string filename = «»;
    public string Filename

```

Продовження лістингу А.4

```

get {return filename;}
set
{
    filename = value;
    Text = «Редагування « + filename;
}
}
string IdDogovory1, Vidilennya1, Inspector1, Error1;
public string is_result()
{
    string result = «»;
    if (labelError.Text.Contains(«- відсоткова ставка не може буде менше»))
    {
        result += «- визначити реальну відсоткову ставку та перезаклучити договір\r\n»;
    }

    if (labelError.Text.Contains(«- рівень забезпечення менше»))
    {
        result += «- провести огляд та оцінку забезпечення та перезаклучити договір\r\n»;
    }

    if (labelError.Text.Contains(«- фінансовий стан відрізняється»))
    {
        result += «- виявити причину неспівпадання фінансового стану позичальника\r\n»;
    }

    if (labelError.Text.Contains(«- фінансовий стан позичальника незадовільний»))
    {
        result += «- переглянути фінансовий стан позичальника та перезаклучити
договір\r\n»;
    }

    if (labelError.Text.Contains(«- виданий кредит клієнту, вік якого менше»))
    {
        result += «- виявити причину видачі кредити особі, вік якої менше за
нормативний\r\n»;
    }

    return result;
}
public ReportForm(string textIdDogovory, string textVidilennya, string textInspector, string
textError)
{
    InitializeComponent();
    IdDogovory1 = textIdDogovory;
    Vidilennya1 = textVidilennya;
    Inspector1 = textInspector;
    Error1 = textError;
}

```

Продовження лістингу А.4

```

}

private void вихідToolStripMenuItem_Click(object sender, EventArgs e)
{
    this.Close();
}

private void ReportForm_Load(object sender, EventArgs e)
{
    labelNoAudit.Text = (Convert.ToInt32(IdDogovory1) - 1000000000).ToString();
    labelKredDogov.Text = IdDogovory1;
    labelViddil.Text = Vidilennya1;
    labelAuditor.Text = «»;
    labelDate.Text = DateTime.Now.Date.ToShortDateString();
    labelKredInspec.Text = Inspector1;
    if (Error1 == «»)
    {
        labelError.Text = «Помилки відсутні»;
        labelVusnovok.Text = «Зауваження відсутні»;
    }
    else
    {
        labelError.Text = Error1;
        labelVusnovok.Text = is_rezult();
    }
}

private void зберегтиToolStripMenuItem_Click(object sender, EventArgs e)
{
    if (Filename == «»)
    {
        saveFileDialogReportsave.Filter = «txt files (*.txt)|*.txt|All files (*.*)|*.*»;
        if (saveFileDialogReportsave.ShowDialog() == DialogResult.OK)
        {
            Filename = saveFileDialogReportsave.FileName;
            openFileDialog1.FileName = Filename;
            richTextBoxReport.SaveFile(Filename, RichTextBoxStreamType.RichText);
        }
    }
    else
    {
        richTextBoxReport.SaveFile(Filename, RichTextBoxStreamType.RichText);
    }
}

private void завантажитиToolStripMenuItem_Click(object sender, EventArgs e)
{

```

Продовження лістингу А.4

```

saveFileDialogReportsave.Filter = «txt files (*.txt)|*.txt|All files (*.*)|*.*»;
if (saveFileDialogReportsave.ShowDialog() == DialogResult.OK)
{
    Filename = saveFileDialogReportsave.FileName;
    openFileDialog1.FileName = Filename;
    richTextBoxReport.SaveFile(Filename, RichTextBoxStreamType.RichText);
}
}

private void відмінаДіїToolStripMenuItem_Click(object sender, EventArgs e)
{
    richTextBoxReport.Undo();
}

private void вибратиВсеToolStripMenuItem_Click(object sender, EventArgs e)
{
    richTextBoxReport.SelectAll();
}

private void копіюватиToolStripMenuItem_Click(object sender, EventArgs e)
{
    richTextBoxReport.Copy();
}

private void шрифтToolStripMenuItem_Click(object sender, EventArgs e)
{
    fontDialog1.ShowDialog();
    richTextBoxReport.SelectionFont = fontDialog1.Font;
    richTextBoxReport.SelectionColor = fontDialog1.Color;
}
}

```