

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ХАРЧОВИХ ТЕХНОЛОГІЙ**

Інститут (факультет) Автоматизації і комп'ютерних систем
Кафедра Інформаційних технологій, штучного інтелекту і кібербезпеки

«До захисту в ЕК»
Директор інституту(декан факультету)
_____ Андрій ФОРСЮК
(підпис) (ім'я та прізвище)

« ____ » _____ 2024р.

«До захисту допущено»
Завідувач кафедри
_____ Сергій ГРИБКОВ
(підпис) (ім'я та прізвище)

« ____ » _____ 2024р.

**КВАЛІФІКАЦІЙНА РОБОТА
НА ЗДОБУТТЯ ОСВІТНЬОГО СТУПЕНЯ БАКАЛАВРА**

зі спеціальності 122 Комп'ютерні науки
(код та назва спеціальності)
освітньо-професійної програми Комп'ютерні науки

на тему: «Розроблення та впровадження програмного забезпечення для обчислення, аналізу платежів та графіка погашення кредитів»

Виконав: здобувач 3 курсу, групи 5-ІТ

Жолобняк Антон Анатолійович
(прізвище, ім'я, по батькові повністю) (підпис)

Керівник Шевчук Ольга Аркадіївна
(прізвище, ім'я та по батькові повністю) (підпис)

Консультанти _____
(ім'я та прізвище) (підпис)

_____ (ім'я та прізвище) (підпис)

_____ (ім'я та прізвище) (підпис)

Рецензент _____
(ім'я та прізвище) (підпис)

Я як здобувач(ка) Національного університету харчових технологій розумію і підтримую політику університету з академічної доброчесності. Я не надавав(-ла) і не одержував(-ла) незарядженої допомоги під час підготовки цієї роботи. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

Здобувач _____
(підпис)

Київ – 2024р.

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ХАРЧОВИХ ТЕХНОЛОГІЙ

Інститут (факультет) Автоматизації і комп'ютерних систем

Кафедра Інформаційних технологій, штучного інтелекту і кібербезпеки

Освітній ступінь бакалавр

Спеціальність 122 Комп'ютерні науки

Освітньо-професійна програма Комп'ютерні науки

ЗАТВЕРДЖУЮ

Завідувач кафедри

Інформаційних технологій, штучного інтелекту і кібербезпеки

Сергій ГРИБКОВ

“ 15 ” листопада 2023 року

З А В Д А Н Н Я

НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧА

Жолобняку Антону Анатолійовичу

(прізвище, ім'я, по батькові)

1. Тема роботи Розроблення та впровадження програмного забезпечення для обчислення, аналізу платежів та графіка погашення кредитів

керівник роботи Шевчук Ольга Аркадіївна

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом закладу вищої освіти від 14 листопада 2023 року № 934-кв

2. Строк подання здобувачем роботи 26.01.2024 р.

3. Вихідні дані до роботи

1. Види кредитів

2. Види кредитних платежів

3. Алгоритм обчислення за видами платежів

4. Розрахунок процентної ставки кредиту

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити) основи організації кредитного ринку, аналіз кредитних послуг, види платежів за кредитами, огляд існуючих програмних рішень, розробка UML діаграми класів в Class Diagrams in C#, розроблення інтерфейсу програмного забезпечення, розроблення інструкції користувача, розрахунок економічного ефекту від впровадження, опис заходів з охорони праці та техніки безпеки.

5. Перелік графічного матеріалу

1. Концептуальна діаграма

2. UML діаграма класів

3. Аналіз видів кредитних платежів та графік погашення кредиту

4. Приклади роботи програмного забезпечення (інтерфейс користувача)

5. Фрагменти коду

АНОТАЦІЯ

Жолобняк А. А. «Розроблення та впровадження програмного забезпечення для обчислення, аналізу платежів та графіка погашення кредитів».

Кваліфікаційна робота складається з 108 сторінки, 35 рисунків, 10 таблиць, 5 додатків та 18 джерел.

В даній роботі розроблено програмного забезпечення для обчислення, аналізу платежів та графіка погашення кредитів. Пояснювальна записка включає в себе системний аналіз предметної області та об'єкту дослідження, розроблено на основі аналізу концептуальну діаграму програмного забезпечення. На основі аналізу було виявлено, що впровадження програмного забезпечення для обчислення, аналізу платежів та графіка погашення кредитів буде мати позитивний ефект, тому дана робота містить опис всіх стадій розробки програмного засобу. Також пояснювальна записка містить розрахунок економічного ефекту від впровадження системи, технічне завдання на розроблення програмного додатку, інструкцію користувача та опис охорони праці та техніки безпеки при роботі з ПК.

Концептуальна діаграма створена в середовищі Draw.io.

Для проєктування UML діаграми класів використано інструмент Class Diagrams in C#.

Розроблення програмного забезпечення для обчислення, аналізу платежів та графіка погашення кредитів реалізовано в середовищі Visual Studio 2022. Мова програмування C#.

КЛЮЧОВІ СЛОВА: КОНЦЕПТУАЛЬНА ДІАГРАМА, UML ДІАГРАМА КЛАСІВ, CLASS DIAGRAMS, VISUAL STUDIO ,C#.

ANNOTATION

Zholobnyak A. "Development and implementation of software for calculation, analysis of payments and loan repayment schedule."

The qualification paper consists of 108 pages, 35 figures, 10 tables, 5 appendices and 18 sources.

In this work, software was developed for calculation, analysis of payments and loan repayment schedule. The explanatory note includes a systematic analysis of the subject area and the research object, a conceptual diagram of the software is developed on the basis of the analysis. Based on the analysis, it was found that the introduction of software for calculation, analysis of payments and loan repayment schedule will have a positive effect, therefore this work contains a description of all stages of the development of the software tool.

Also, the explanatory note contains a calculation of the economic effect of the system implementation, a technical task for the development of a software application, a user manual and a description of labor protection and safety techniques when working with a PC.

The conceptual diagram was created in the Draw.io environment.

The Class Diagrams in C# tool was used to design the UML class diagram.

The development of software for calculation, analysis of payments and loan repayment schedule is implemented in the Visual Studio 2022 environment. The programming language is C#.

KEYWORDS: CONCEPTUAL DIAGRAM, UML CLASS DIAGRAM, VISUAL STUDIO, C#.

ЗМІСТ

ВСТУП.....	7
РОЗДІЛ 1. СИСТЕМНИЙ АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	9
1.1 Основи організації кредитного ринку.....	9
1.2 Огляд кредитних послуг.....	11
1.3 Види платежів за кредитом	14
1.4 Огляд існуючих програмних рішень.....	17
1.5 Постановка задачі.....	27
1.6 Розрахунок економічного ефекту від впровадження програмного забезпечення	29
РОЗДІЛ 2. ТЕХНІЧНЕ ЗАВДАННЯ НА ПРОЄКТУВАННЯ.....	37
РОЗДІЛ 3. ОПИС КОМПЛЕКСУ ЗАДАЧ ДЛЯ АВТОМАТИЗАЦІЇ.....	40
3.1 Інформаційне забезпечення розробки.....	40
3.2 Алгоритмізація та реалізація комплексу задач автоматизації.....	47
3.3 Розробка програмного забезпечення.....	55
3.4 Інструкція користувача.....	69
3.5 Технічне та системне забезпечення системи	79
3.5.1 Обґрунтування вибору технічних засобів	79
3.5.2 Обґрунтування вибору операційної системи та протоколу обміну даними	80
РОЗДІЛ 4. ОХОРОНА ПРАЦІ	82
4.1 Вимоги безпеки під час виконання робіт за ПК	82
4.2 Дії персоналу в надзвичайних ситуаціях.....	87
ВИСНОВКИ.....	92
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	94
ДОДАТКИ.....	96
Додаток А. UML діаграма класів.....	96
Додаток Б. Код Form	97
Додаток В. Код для обчислення, аналізу платежів та погашення кредитів.....	99
Додаток Г. Аналіз диференційованого платежу та графіка погашення кредиту	105
Додаток Д. Аналіз ануїтетного платежу та графіка погашення кредиту	107

ВСТУП

Програмне забезпечення для обчислення, аналізу платежів та графіка погашення кредитів є актуальним і важливим інструментом для банківської сфери, фінансових установ та клієнтів, які користуються кредитними послугами. Таке програмне забезпечення дозволяє ефективно розраховувати різні параметри кредитних операцій, такі як вид кредиту, сума кредиту, термін погашення, процентна ставка, щомісячний платіж, загальна сума переплати, а також візуалізувати графік погашення кредиту з урахуванням можливих змін умов кредитування. Така візуалізація допомагає клієнтам краще розуміти свої фінансові зобов'язання та планувати свій бюджет. Крім того, програмне забезпечення для обчислення, аналізу платежів та графіка погашення кредитів може використовуватися для порівняння різних пропозицій від різних кредиторів та вибору найвигіднішої для клієнта.

Однак, не всі існуючі програмні продукти для обчислення, аналізу платежів та графіка погашення кредитів мають достатній рівень якості, надійності, зручності та функціональності. Деякі з них мають обмежену кількість опцій, не підтримують різні види кредитування, не враховують можливість дострокового погашення кредиту, не мають гнучкого інтерфейсу, не інтегруються з іншими системами тощо. Тому, розробка та впровадження програмного забезпечення для обчислення, аналізу платежів та графіка погашення кредитів, яке б відповідало сучасним вимогам та стандартам, є важливою та цікавою задачею.

Метою даної кваліфікаційної роботи є розроблення та впровадження програмного забезпечення для обчислення, аналізу платежів та графіка погашення кредитів мові програмування C#, яке б забезпечувало швидке, точне та зручне виконання розрахунків та візуалізацію результатів. Для досягнення цієї мети було поставлено наступні завдання:

- провести аналіз предметної області та вимог до програмного забезпечення;
- розробити архітектуру та дизайн програмного забезпечення;
- реалізувати програмний код за допомогою мови програмування C# та фреймворку .NET;
- провести тестування та налагодження програмного забезпечення.

Об'єктом дослідження є процес обчислення, аналізу платежів та графіка погашення кредитів. Предметом дослідження є програмне забезпечення для обчислення, аналізу платежів та графіка погашення кредитів.

Наукова новизна роботи полягає в розробленні програмного забезпечення, яке використовує сучасні технології та методики розробки, а також має високий рівень якості, надійності, зручності та функціональності. Практична значущість роботи полягає в тому, що розроблене програмне забезпечення може застосовуватися в реальних умовах для вирішення задач обчислення, аналізу платежів та графіка погашення кредитів.

РОЗДІЛ 1. СИСТЕМНИЙ АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Основи організації кредитного ринку

Однією з основних рушійних сил глобалізації та цифровізації економіки є науково-технічний прогрес, який у повній мірі впливає на розвиток фінансової сфери через упровадження інноваційних технологій, ідей та продуктів. Постійний прогрес у царині обробки й передачі інформації, що став можливим завдяки широкому застосуванню цифрових технологій, створює передумови трансформації платіжних систем у сфері міжбанківських розрахунків та надання кредитних послуг.

У процесі динамічного розвитку економіки виникають гранично високі вимоги до доступності, безпечності, швидкості, зручності й вартості розрахунків і переказів коштів. Основи організації кредитного ринку України – це процес формування, функціонування та регулювання ринку, на якому здійснюються операції з надання та отримання кредитів. Кредитний ринок є важливою складовою фінансової системи країни, яка впливає на економічний розвиток, інвестиційну активність, соціальне благополуччя та інші аспекти життя суспільства. Кредитний ринок України має свої особливості, тенденції та проблеми, які потребують аналізу та вирішення.

В Україні кредитний ринок розвивається досить активно. У 2023 році обсяг кредитування економіки становив близько 4 трлн гривень. Найбільшу частку в кредитуванні економіки займають комерційні банки.

Кредитний ринок – це особлива сфера економічних відносин, де об'єктом операцій є капітал, наданий на певних умовах. Об'єктом кредитного ринку є тимчасово вільний капітал у натуральній чи грошовій формі [5]. Кредитний ринок є важливим інструментом економічного розвитку. Він забезпечує перерозподіл фінансових ресурсів від кредиторів до позичальників, сприяє розвитку підприємництва, інвестицій та споживання.

Основними учасниками цього ринку є кредитори, що позичають вільні кошти на різних умовах, та позичальники (рисунок 1).

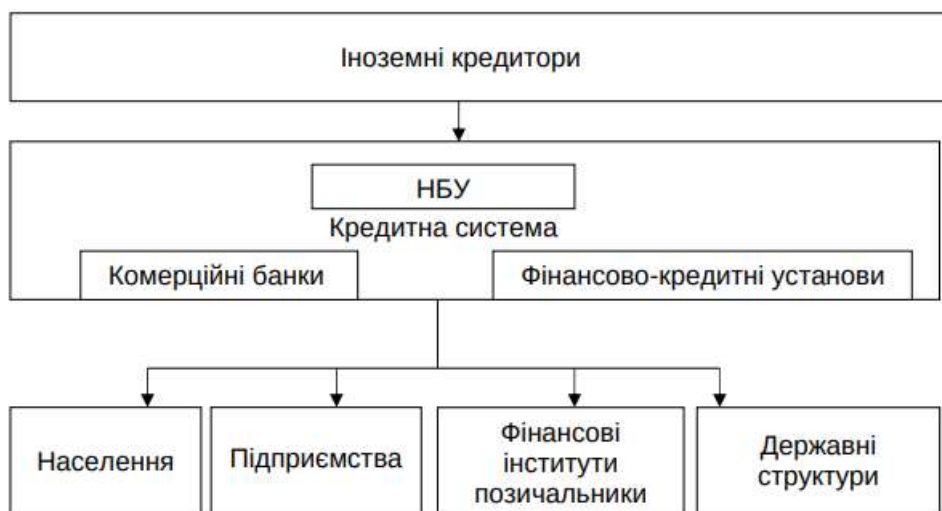


Рисунок 1.1 – Структура кредитного ринку України

Сьогодні в Україні діяльність із надання кредитів, позик регулюється такими законами:

- 1) Конституцією України;
- 2) Законом України «Про банки та банківську діяльність»;
- 3) Законом України «Про Національний банк України».

Функції кредитного ринку визначаються його суттю і роллю, яку він виконує в системі капіталістичного господарства, а також завданнями з відтворення капіталістичних виробничих відносин.

Слід виділити п'ять основних функцій кредитного ринку:

- обслуговування товарного обороту через кредит;
- акумуляція або збирання грошових заощаджень (накопичень) підприємств, населення, держави, а також іноземних клієнтів;
- трансформація грошових фондів безпосередньо в позичковий капітал і використання його у вигляді капіталовкладень для обслуговування процесу виробництва;
- обслуговування держави і населення як джерел капіталу для покриття державних і споживчих витрат;

- прискорення концентрації і централізації капіталу, сприяння утворенню потужних фінансово-промислових груп.

Ефективність кредитної політики залежить від правильного вибору параметрів кредитних відносин, найважливішими з яких є:

- розмір відсотків по кредиту;
- застереження про доступність кредиту для вітчизняних компаній;
- рівень кредитного ризику;
- термін позики.

Кредитний ринок сприяє зростанню виробництва і товарообігу, руху капіталів усередині країни, трансформації грошових заощаджень в капіталовкладення, реалізації науково-технічної революції, оновленню основного капіталу. У цьому сенсі ринок опосередковує різні фази відтворення, є своєрідною опорою матеріальної сфери виробництва, звідки вона черпає додаткові грошові ресурси.

Організація кредитного ринку в Україні є складним процесом, який враховує важливі аспекти фінансового законодавства, учасників ринку та регулювання з боку Національного банку. Це створює стабільний фундамент для функціонування кредитної системи та підтримки економічного розвитку країни.

1.2 Огляд кредитних послуг

Кредитні послуги - це вид фінансових послуг, які надаються банками та іншими фінансовими установами клієнтам, які потребують позичити гроші на різні цілі. Кредитні послуги можуть бути різними за видами, умовами, термінами, відсотковими ставками, вимогами до позичальників та іншими характеристиками. Для того, щоб обрати найкращу кредитну пропозицію, потрібно порівняти різні варіанти та ознайомитися з їх перевагами та недоліками.

У Законі України «Про банки і банківську діяльність» [4] окремого визначення поняття «кредитна операція» немає, хоча зазначено що до кредитних операцій належать:

– залучення у вклади (депозити) коштів та банківських металів від необмеженого кола юридичних і фізичних осіб;

– відкриття та ведення поточних (кореспондентських) рахунків клієнтів, у тому числі у банківських металах; розміщення залучених у вклади (депозити), у тому числі на поточні рахунки, коштів та банківських металів від свого імені, на власних умовах та на власний ризик;

– здійснення операцій на ринку цінних паперів від свого імені; надання гарантій і поруки та інших зобов'язань від третіх осіб, які передбачають їх виконання у грошовій формі;

– придбання права вимоги на виконання зобов'язань у грошовій формі за поставлені товари чи надані послуги, приймаючи на себе ризик виконання таких вимог і прийом платежів (факторинг).

Кредити, які надаються банками, можуть бути розподілені за такими ознаками [3]

1. За строками користування:

а) короткострокові – до одного року;

б) середньострокові – до трьох років;

в) довгострокові – понад три роки.

2. За забезпеченням:

а) забезпечені заставою (майном, майновими правами, цінними паперами);

б) гарантовані (банками, фінансами чи майном третьої особи);

в) з іншим забезпеченням (поручительство, свідоцтво страхової організації);

г) незабезпечені (бланкові).

3. За методами надання:

- а) у разовому порядку;
- б) відповідно до відкритої кредитної лінії.

4. За строками погашення:

- а) водночас;
- б) у розстрочку;
- в) достроково (за вимогою кредитора або за заявою позичальника);
- г) із регресією платежів;
- д) після закінчення обумовленого періоду (місяця, кварталу).

На підставі зазначених критеріїв банк формує кредитні портфелі однорідних кредитів (які складаються із загальної суми заборгованості за наданими банком однорідними кредитами).

Вагоме місце на кредитному ринку в реаліях сьогодення займає споживчий кредит. Споживчий кредит - це вид кредиту, який надається фінансовою установою (банком, кредитною організацією тощо) фізичній особі для задоволення її особистих потреб або витрат. Цей вид кредиту часто використовується для придбання товарів і послуг, таких як автомобілі, електроніка, меблі, ремонт житла, подорожі, оплата медичних послуг чи освіти.

Основні риси споживчого кредиту включають фіксований чи змінний відсотковий платіж, визначений термін погашення та регулярні платежі, які клієнт здійснює протягом узгодженого періоду. Процентна ставка на споживчий кредит може бути фіксованою або змінюватися залежно від ринкових умов та політики кредитора.

Зазвичай для отримання споживчого кредиту потрібно мати сталий дохід, добру кредитну історію та відповідати іншим умовам, встановленим фінансовою установою.

Для отримання споживчого кредиту в Україні потрібно:

- Бути дієздатною фізичною особою віком від 18 до 75 років.
- Місце проживання на території України.

- Не мати заборгованості за кредитами перед іншими кредитодавцями.
- Мати регулярні доходи, достатні для погашення кредиту.
- Надати кредитодавцю документи, що підтверджують вашу платоспроможність та дієздатність.

Споживчий кредит може бути наданий у формі кредитної картки, особистого кредиту або іншого фінансового інструменту, призначеного для особистого використання.

1.3 Види платежів за кредитом

Для розробки програмного забезпечення для обчислення, аналізу платежів та графіка погашення кредитів доцільно проаналізувати види платежів за кредитом для обрання найкращого способу погашення заборгованості.

Види платежів за кредитом диференційований та ануїтетний – це два способи розрахунку суми, яку позичальник повинен сплачувати банку або іншому кредитору за користування позикою. Вони відрізняються тим, як розподіляється погашення основної суми кредиту та нарахованих процентів протягом усього терміну кредитування.

Диференційований та ануїтетний кредити відрізняються не лише структурою погашення основної суми кредиту, але й видами платежів, які клієнт повинен здійснювати протягом терміну кредиту. Давайте розглянемо основні види платежів за кожним типом кредиту.

Диференційований кредит.

1. Початковий платіж.

Опис. Вартість платежу включає як частину основного боргу, так і відсотки за залишок боргу.

Особливості. Початковий платіж вищий, оскільки включає велику частину основного боргу.

2. Зменшення основного боргу:

Опис. З часом частина платежу, призначена для погашення основного боргу, зменшується, а відсотки залишаються відносно сталими.

Особливості. Призначено для поетапного погашення основної суми кредиту.

3. Спадаючий платіж:

Опис. У платежах протягом терміну кредиту основний борг поступово зменшується, що призводить до спадання загальної суми платежів.

Особливості. Зменшення платежів робить кредит доступнішим у перші роки, але загальні витрати на відсотки можуть бути вищими.

Розрахунок ставки при диференційованих платежах [6].

В основному, банки і мікрофінансові організації самі визначають схему, за якою їх клієнти будуть гасити борги. За диференційованою схемою кредит поступово зменшується до кінця терміну. Основний борг в ньому ділиться на рівні частки, а відсоток з кожним місяцем зменшується виходячи зі зменшення з кожною виплатою залишку.

Щоб розрахувати розмір виплат по тілу, потрібно розділити вартість кредиту на 12 місяців, використовуючи формулу:

Сума відсотків = (залишок тіла × процентна ставка × кількість днів користування) / (100 × 365).

Ануїтетний кредит.

1. Рівномірний платіж.

Опис. Щомісячний платіж включає рівну частину основного боргу та відсотків на залишок боргу.

Особливості. Регулярний рівномірний платіж протягом усього терміну кредиту.

2. Збільшення частки основного боргу.

Опис. Якщо в розрахунку на початкових етапах платіжів відсотки перевищують частину основного боргу, то з часом ця ситуація змінюється, і більша частина платежу використовується для погашення основного боргу.

Особливості. Зменшення відсоткової частки платежу в часі.

3. Стабільність платежів.

Опис. Платежі залишаються відносно сталими протягом терміну кредиту.

Особливості. Клієнт знає фіксовану суму, яку він повинен сплачувати щомісяця, що полегшує планування бюджету.

Розрахунок відсотків при ануїтетній схемі погашення [6].

Ануїтетна схема розрахунку з платежів зустрічається найчастіше. Ануїтетні (рівні) платежі є однією з можливих схем погашення кредиту. Розмір платежу визначається за формулами для ануїтету за умови, що перший платіж відбудеться за один період після надання суми кредиту.

Частою альтернативою ануїтетним платежам є стандартна (класична, лінійна) схема, відома також як диференційовані платежі. У стандартній схемі основна сума боргу погашається рівними частинами разом з поточними відсотками. Оскільки відсотки нараховуються від кожного разу з меншої суми боргу, то платежі з часом зменшуються.

Якщо відсотки сплачуються щомісячно (за ануїтетною схемою) розрахунок буде за формулою:

$A = K * S$, де A - щомісячний ануїтетний платіж, K - коефіцієнт ануїтету, S - сума кредиту.

Ануїтет завжди обходиться дорожче, ніж виплати при розрахунку диференційованими платежами. Але в той же час виплачувати кредит за ануїтетною схемою набагато легше, тому як виплати розподілені рівномірно на весь термін кредиту.

Обираючи між диференційованим та ануїтетним кредитами, клієнти повинні враховувати свої фінансові можливості, планування бюджету та інші

фактори. Кожен з типів кредитів має свої переваги та обмеження, і вибір залежить від конкретних потреб та фінансового стану позичальника.

1.4 Огляд існуючих програмних рішень

Існує ряд програмних рішень для обчислення, аналізу платежів та графіка погашення кредитів. До них відносяться:

- інтернет-банкінг;
- вебсайти погашення кредитів;
- фінансові калькулятори;
- програмне забезпечення для фінансового планування;
- інтернет-банкінг.

Багато банків пропонують своїм клієнтам можливість погашення кредитів через інтернет-банкінг. Ця функція зазвичай безкоштовна і проста у використанні. Але не гарантує конфіденційність.

Щоб здійснити платіж за кредит через інтернет-банкінг, вам потрібно буде ввести такі дані:

- номер договору або номер рахунку;
- сума платежу;
- дата платежу.

Деякі банки також дозволяють вам встановлювати регулярні платежі за кредитами через інтернет-банкінг.

Вебсайти погашення кредитів.

Існує ряд вебсайтів, які пропонують послуги погашення кредитів. Ці вебсайти зазвичай пропонують такі функції:

- оплата кредитів;
- історія платежів;
- нагадування про платежі;
- порівняння процентних ставок;

- розрахунок щомісячних платежів.

Деякі вебсайти погашення кредитів також пропонують додаткові функції, такі як:

- автоматичні платежі;
- планування бюджету;
- консультації з кредитом.

Фінансові калькулятори можуть бути використані для розрахунку таких показників, як:

- сума щомісячного платежу;
- загальна сума відсотків;
- термін погашення кредиту.

Фінансові калькулятори для погашення кредитів доступні в Інтернеті та в деяких мобільних додатках.

Програмне забезпечення для фінансового планування.

Програмне забезпечення для фінансового планування може допомогти відстежувати доходи та витрати, а також створювати бюджети. Деякі програми для фінансового планування також пропонують функції погашення кредитів, такі як:

- оплата кредитів;
- історія платежів;
- нагадування про платежі;
- порівняння процентних ставок;
- розрахунок щомісячних платежів.

При виборі програмного рішення для погашення кредиту слід враховувати такі фактори:

Вартість. Вартість програмних рішень для погашення кредитів може відрізнятися. Деякі програми є безкоштовними, інші - платними.

Функціональність. Важливо вибрати програмне рішення, яке пропонує всі необхідні функції для погашення кредиту.

Зручність використання. Програмне рішення повинно бути простим у використанні.

Для розроблення повнофункціонального програмного продукту слід розглянути аналоги програмного забезпечення для обчислення, аналізу платежів та графіка погашення кредитів, які допомагають у керуванні та відстеженні погашення кредитів. Деякі з них призначені для користувачів, а інші використовуються фінансовими установами для управління кредитами та взаємодії з клієнтами. Ось огляд деяких існуючих програмних рішень у цій області.

YNAB (You Need a Budget).

YNAB – це програма для особистого бюджетування, яка також дозволяє вести облік кредитів та планувати їх погашення. Програмне забезпечення YNAB має простий та інтуїтивно зрозумілий формат. Інструмент запланованих транзакцій згортається, якщо ви не бажаєте його використати, а рахунки, що використовуються для складання бюджету, позабюджетних та закритих рахунків, згортаються на бічній панелі.

Програмне забезпечення відображає зведену інформацію про бюджет на початку кожного місяця, а YNAB має функцію автозбереження, яка часто зберігає дані. Ви також можете відновити свій бюджет до попередніх версій, якщо припустилися помилки. Інші функції включають:

- простий друк;
- переключення між кількома бюджетами без перезапуску;
- майстер звірки облікових записів;
- жодних дублюючих транзакцій при імпорті файлів QIF;
- поліпшено гнучкий експорт даних;
- нові звіти дозволяють легко деталізувати деталі.

YNAB автоматично зберігає зміни під час використання програмного забезпечення за допомогою Cloud Sync. Для цієї функції потрібен безкоштовний обліковий запис Dropbox, і можна обмінюватися даними на всіх своїх

комп'ютерах та використовувати YNAB на будь-якому комп'ютері під керуванням Windows або Mac.

Для використання даного програмного забезпечення потрібна реєстрація, а також надається пробний період використання (рисунок 1.2).



Рисунок 1.2 – YNAB

Loan Calculator.

Декілька додатків для мобільних пристроїв, таких як "Loan Calculator" для iOS або "Loan Calculator Pro" для Android, дозволяють швидко розрахувати різні аспекти кредитів (рисунок 1.3).

Використовуючи кредитний калькулятор можна визначити щомісячний платіж за будь-яким кредитом. Також можна побачити, як ваш кредит амортизується або скільки виплачується протягом періоду погашення.

Кредитний калькулятор використовує базову інформацію для оцінки платежів на виплат і дає уявлення про те, скільки відсотків потрібно платити протягом терміну дії кредиту.

Кредитний калькулятор зазвичай використовує такі вхідні дані:

- Сума кредиту.
- Процентна ставка.
- Термін погашення кредиту.
- Періодичність платежів.

На основі цих даних кредитний калькулятор розраховує такі показники:

- Сума щомісячних платежів.
- Загальна сума відсотків за кредитом.
- Загальна сума кредиту.
- Графік погашення кредиту.

Кредитний калькулятор може бути корисним для фізичних та юридичних осіб, які планують взяти кредит. Він дозволить користувачам порівняти різні кредитні пропозиції та оцінити фінансові наслідки кредитування.

The screenshot displays a loan calculator interface with two main sections: 'Loan details' and 'Your loan estimate'.

Loan details:

- Loan amount: \$10,000
- Interest rate: 10%
- Loan term (months): 60
- Start date: 01 / 2024
- Button: ADD ORIGINATION FEE

Your loan estimate:

Monthly payment	Total principal	\$10,000
\$212.47	Total interest payments	\$2,748.23
	Total loan payments ⓘ	\$12,748.23
	Payoff date	01 / 2029

Buttons: Show amortization schedule, See if you pre-qualify for a personal loan - without affecting your credit score.

Additional text: Just answer a few questions to get personalized rate estimates from multiple lenders. Learn more about pre-qualifying.

Bottom section: Loan amount \$, SEE IF YOU PRE-QUALIFY, on NeedWallet.

Рисунок 1.3 – Loan Calculator

Mint.

Mint – це програма для управління фінансами, яка допомагає відслідковувати кредити, аналізувати витрати та планувати бюджет (рисунок 1.4).

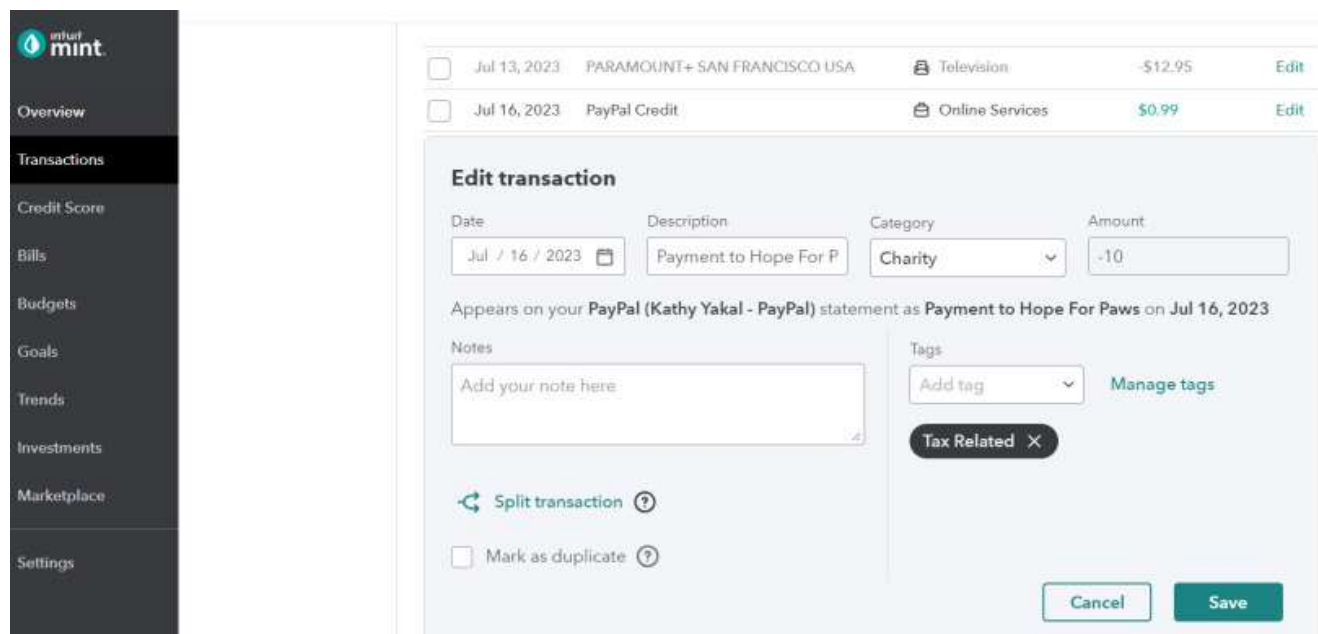


Рисунок 1.4 – Mint

Mint є популярною програмою для управління фінансами, розробленою Intuit. Вона дозволяє користувачам відслідковувати свої фінанси, аналізувати витрати та планувати бюджет. Основні функції Mint включають:

Відстеження фінансів: Mint автоматично збирає та категоризує транзакції з різних банків, кредитних карт та інших фінансових облікових записів.

Аналіз витрат: Програма надає користувачам графіки та звіти про їхні витрати, щоб вони могли краще розуміти, куди йде їхній бюджет.

Бюджетування: Mint дозволяє створювати бюджети для різних категорій витрат та надсилати сповіщення, якщо перевищений визначені ліміти.

Відстеження кредитів: Програма допомагає відслідковувати кредитний статус, включаючи кредитний рейтинг та інші фінансові параметри.

Планування фінансових цілей: Користувачі можуть встановлювати фінансові цілі, такі як заощадження на купівлю автомобіля чи оплату кредитів, і отримувати поради щодо досягнення цих цілей.

Попередження про події: Mint може надсилати користувачам сповіщення про надходження рахунків, намічені транзакції та інші важливі фінансові події.

Із Mint можна працювати як через вебверсію, так і через мобільні додатки для iOS та Android, що робить його зручним для використання в русі.

Quicken:

Quicken – це комплексне програмне рішення для фінансового управління, яке включає інструменти для керування кредитами (рисунок 1.5).



Рисунок 1.5 – Quicken

Quicken — це інструмент управління особистими фінансами, розроблений компанією Quicken Inc. Це популярне програмне забезпечення, яке допомагає окремим особам і малим підприємствам керувати своїми фінансовими операціями, відстежувати витрати, створювати бюджети та контролювати

інвестиції. Quicken існує вже багато років і розвинувся в різні версії, щоб задовольнити різні потреби користувачів.

Основні функції Quicken включають:

Відстеження витрат: користувачі можуть вводити та класифікувати свої витрати, що полегшує відстеження, куди йдуть їхні гроші.

Бюджетування: Quicken дозволяє користувачам створювати та керувати бюджетами, щоб допомогти їм планувати та контролювати свої витрати.

Управління рахунками: користувачі можуть налаштовувати та відстежувати свої рахунки, допомагаючи їм бути організованими та уникати прострочення платежів.

Відстеження інвестицій: Quicken надає інструменти для моніторингу інвестиційних портфелів, відстеження цін на акції та аналізу ефективності інвестицій.

Податкове планування: деякі версії Quicken пропонують функції податкового планування, допомагаючи користувачам оцінити свої податкові зобов'язання та оптимізувати свої фінансові стратегії.

Фінансова звітність: користувачі можуть створювати різні фінансові звіти, щоб отримати уявлення про свій фінансовий стан і приймати обґрунтовані рішення.

Quicken доступний у різних версіях, таких як Quicken Starter, Quicken Deluxe, Quicken Premier і Quicken Home & Business, кожна з яких пропонує різний набір функцій для задоволення конкретних вимог користувачів. Користувачі часто можуть підключати Quicken до своїх банківських рахунків та інших фінансових установ для автоматичного завантаження та класифікації транзакцій, що робить процес управління фінансами більш ефективним.

Важливо зауважити, що функції та можливості Quicken можуть змінюватися з новими версіями, тому бажано перевірити офіційний вебсайт Quicken або документацію користувача, щоб отримати найновішу інформацію. Крім того, на ринку доступні інші інструменти та програми для управління

особистими фінансами, тому користувачі можуть вивчити різні варіанти, щоб знайти той, який найкраще відповідає їхнім потребам.

Credit Karma.

Credit Karma – це сервіс, який надає не тільки інформацію про кредитний рейтинг, але й можливість відслідковувати та аналізувати кредити.

Credit Karma – це особиста фінансова компанія, яка надає безкоштовні кредитні рейтинги та кредитні звіти споживачам (рисунок 1.6).

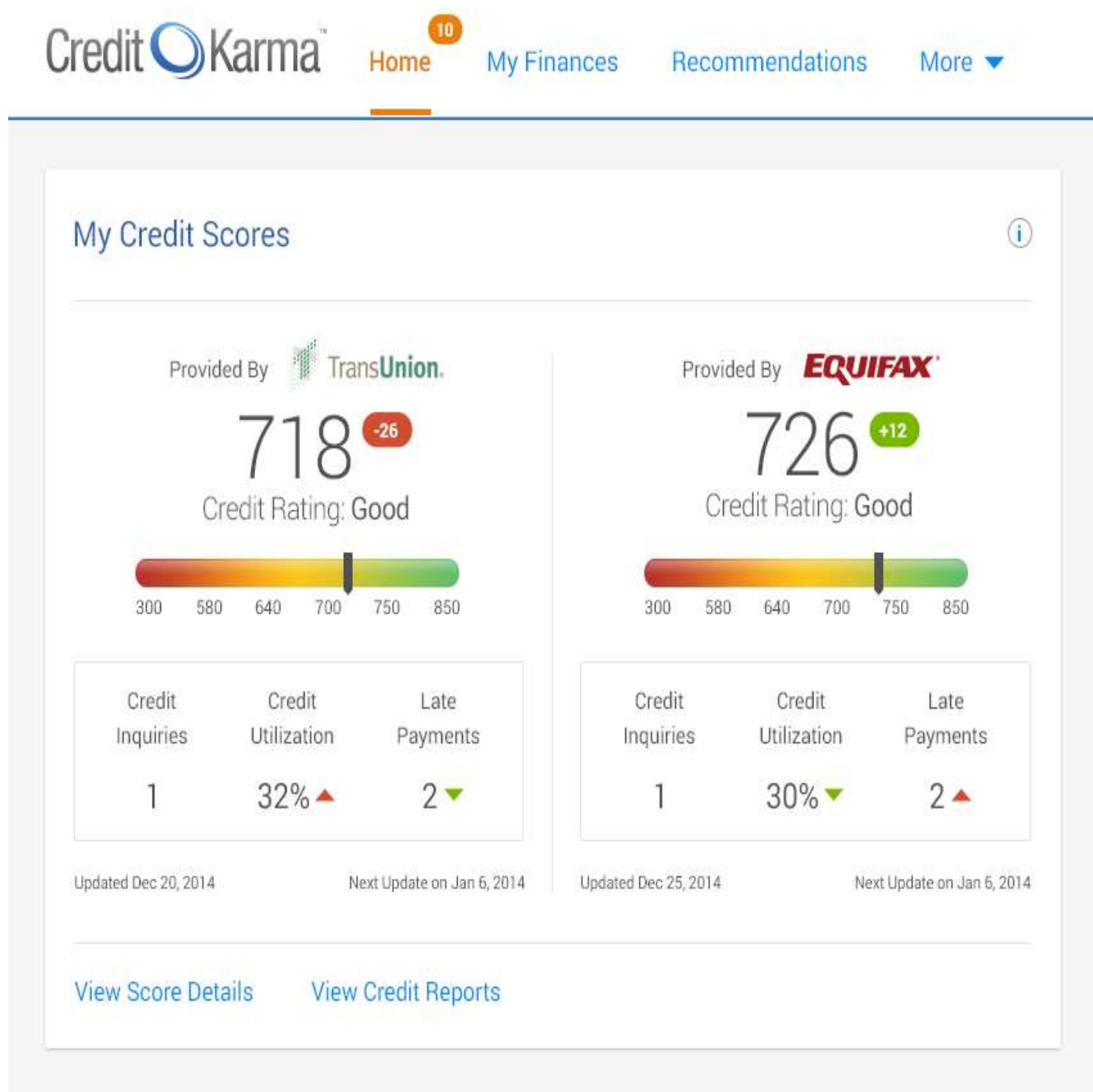


Рисунок 1.6 – Credit Karma

До основних функцій та послуг, які пропонує інформаційний сервіс Credit Karma належить:

Безкоштовні кредитні рейтинги. Credit Karma надає користувачам безкоштовний доступ до їхніх кредитних рейтингів, які розраховуються на основі інформації двох найбільших кредитних бюро: Equifax і TransUnion.

Кредитні звіти. Користувачі також можуть переглядати свої кредитні звіти на Credit Karma, які містять детальну інформацію про їх кредитну історію, рахунки та інші відповідні фінансові дані.

Кредитний моніторинг. Credit Karma пропонує послуги кредитного моніторингу, сповіщаючи користувачів про будь-які зміни в їхніх кредитних звітах, наприклад про нові рахунки, запити чи негативну інформацію.

Фінансові інструменти. Платформа надає різні фінансові інструменти та калькулятори, які допомагають користувачам краще розуміти свої фінанси та керувати ними. Ці інструменти можуть включати калькулятори бюджету, кредитні калькулятори та інші ресурси.

Симулятор кредитного рейтингу. Credit Karma пропонує симулятор кредитного рейтингу, який дозволяє користувачам побачити, як певні фінансові дії, такі як погашення боргу чи відкриття нової кредитної картки, можуть вплинути на їхній кредитний рейтинг.

Кредитна освіта. Credit Karma надає освітні ресурси, щоб допомогти користувачам зрозуміти кредитні оцінки, звіти та інші фінансові теми. Це включає статті, поради та посібники.

Важливо відзначити, що Credit Karma заробляє гроші за допомогою цільової реклами та рекомендацій фінансових продуктів користувачам на основі їхніх кредитних профілів. У 2020 році Credit Karma була придбана компанією Intuit, яка розробляє такі популярні фінансові програми, як QuickBooks і TurboTax. У результаті придбання Credit Karma тепер є частиною сімейства фінансових інструментів і послуг Intuit.

1.5 Постановка задачі

У сучасному світі все більше людей беруть кредити для придбання нерухомості, автомобілів, техніки тощо. Однак, не всі розуміють, як правильно розраховувати платежі за кредитами та як скласти графік погашення кредиту, який дозволить їм швидше позбутися кредитних зобов'язань.

Програмне забезпечення для обчислення, аналізу платежів та графіка погашення кредитів може допомогти користувачам у таких завданнях:

- Розрахунок щомісячних платежів за кредитом.
- Розрахунок загальної суми відсотків, які потрібно сплатити за весь період користування кредитом.
- Порівняння різних варіантів погашення кредиту.
- Складання графіка погашення кредиту.

Програмне забезпечення повинно бути розроблене на мові програмування, яка дозволяє легко розширювати та підтримувати систему.

Забезпечити сумісність з різними операційними системами (Windows, Linux).

Використання бази даних для збереження введених користувачем параметрів та історії розрахунків.

Етапи реалізації проекту.

1. Аналіз вимог:

- Визначення основних функцій та вимог до програмного продукту.
- Вивчення існуючих аналогів та їхніх переваг/недоліків.

2. Проєктування.

- Розробка структури бази даних для збереження інформації.
- Визначення архітектури програми та інтерфейсу.

3. Реалізація.

- Написання програмного коду відповідно до визначених специфікацій.
- Проведення тестування для виявлення та виправлення помилок.

4. Тестування.

- Перевірка правильності обчислень та відповідність результатів вимогам.
- Виявлення та виправлення можливих помилок та недоліків.

5. Впровадження.

- Запуск програмного забезпечення для загального використання.
- Забезпечення підтримки та можливості оновлень.

Програмне забезпечення повинно відповідати таким вимогам:

- Простота в користуванні.
- Доступність на різних платформах.
- Можливість налаштування під потреби користувачів
- Надійність та безпека

План впровадження програмного забезпечення повинен включати такі етапи:

- Аналіз потреб користувачів.
- Розробка прототипу.
- Тестування прототипу.
- Фіналізації програмного забезпечення.
- Впровадження програмного забезпечення.

Ефективність програмного забезпечення можна оцінити за такими показниками:

- Кількість користувачів.
- Рівень задоволеності користувачів.
- Економічний ефект.

Розробка та впровадження програмного забезпечення для обчислення, аналізу платежів та графіка погашення кредитів може бути корисним для користувачів, які беруть кредити. Таке програмне забезпечення може допомогти користувачам у таких завданнях:

- Розрахунок щомісячних платежів за кредитом.

- Розрахунок загальної суми відсотків, які потрібно сплатити за весь період користування кредитом.
- Порівняння різних варіантів погашення кредиту.
- Складання графіка погашення кредиту.
- Складання даних погашення кредитів у вигляді таблиці.

1.6 Розрахунок економічного ефекту від впровадження програмного забезпечення

Витрати на розробку програмного забезпечення для обчислення, аналізу платежів та графіка погашення кредитів залежать від таких факторів:

Комплексність програмного забезпечення. Чим більш складним є програмне забезпечення, тим вищі будуть витрати на його розробку.

Вибір технологій. Вибір технологій також впливає на витрати на розробку. Наприклад, використання сучасних технологій може підвищити вартість розробки, але також може підвищити ефективність програмного забезпечення.

Кваліфікація розробників. Кваліфікація розробників також впливає на витрати на розробку. Кваліфіковані розробники можуть розробити програмне забезпечення більш якісно, але їхня робота коштує дорожче.

Витрати на розробку програмного забезпечення для обчислення, аналізу платежів та графіка погашення кредитів наведені в наступних розрахунках (таблиця 1.1).

Таблиця 1.1 – Кошторис витрат на придбання програмного забезпечення

Назва програмного забезпечення	Кількість	Ціна за одиницю, грн	Загальна вартість, грн	Примітка
1	2	3	4	5
Visual Studio Code	1	3700,00	3700,00	Придбано раніше

Windows 10 Pro	1	4400,00	4400,00	Придбано раніше
Офісний пакет Microsoft Office 2021	1	2190,00	2190,00	Придбано раніше
Всього:			10290,00	

Також потрібно врахувати кошторис капітальних затрат на придбання необхідного програмного забезпечення який наведений в таблиці 1.2.

Для складання кошторису капітальних затрат на придбання програмного забезпечення важливо врахувати різні аспекти, такі як ліцензійні витрати, вартість розробки або придбання програми, витрати на інфраструктуру та впровадження, а також витрати на навчання персоналу.

Таблиця 1.2 – Зведений кошторис капітальних затрат на придбання необхідного програмного забезпечення

Найменування затрат	Сума затрат, грн.
1	2
Вартість програмного забезпечення	10290,00
Інші витрати:	
на заходи по науковій організації праці (5% від вартості нового обладнання).	330,00
На засоби автоматизації (2,5% від вартості нового обладнання)	170,00
Всього по кошторису:	10790,00

Розробка програмного забезпечення складається з етапів, кожний етап має витрати які повинні бути враховані. Для правильного планування процесу розробки усі можливі витрати повинні бути занотовані. В таблиці 1.3 наведений перелік етапів та робіт по розробці.

Таблиця 1.3 – Перелік етапів та робіт по розробці проєкту

Найменування	Вид роботи	Виконавець, посада
Стадії	зміст роботи	
1	2	3
1 Підготовча стадія	Дослідження аналогів	Розробник
	Вибір стандарту програмного забезпечення	
	Вивчення та аналіз аналогічних розробок	
	Економічне обґрунтування доцільності виконання проєкту	Проектменеджер
	Складання ТЗ	
Узгодження ТЗ із зацікавленими сторонами		
Складання плану та розрахунок розробки		
2 Технічна пропозиція	Аналіз ТЗ та визначення пріоритетних аспектів розробки	Розробник
	Доведення та уточнення загального обсягу робіт, строків виконання та витрат	Розробник Проектменеджер
	Дослідження технічних особливостей розробки програмного забезпечення	Розробник
3 Теоретична розробка	Визначення переліку технологій, які використовуватимуться при розробці програмного забезпечення	Розробник
4 Практична	Встановлення системи управління	Розробник

реалізація	Написання коду	Розробник
5 Доробка проекту	Тестування програмного забезпечення	Розробник
	Доробка програмного забезпечення з урахуванням результатів тестування	Розробник
	Підготовка звіту про розробку програмного забезпечення	Розробник
	Підготовка технічної документації по розробці програмного забезпечення	Розробник
6 Заключна стадія	Підготовка презентації	Розробник

На основі даного переліку визначається кількість виконавців, тривалість виконання робіт в днях, та рівень оплати праці виконавців по видам робіт, що виконуються. Такі дані приведені у таблиці 1.4

Таблиця 1.4 – Зведені результати тривалості та трудомісткості розробки та реалізації проекту.

Найменування роботи	Виконавець, посада, спеціальність	Кількість виконавців	Тривалість виконання роботи, год	Годинна тарифна ставка, грн	Оплата праці, грн
1	2	3	4	5	6
Аналіз аналогів	Розробник	1	1	110,00	110,00
Економічне обґрунтування доцільності виконання проекту	Розробник	1	3	110,00	330,00

Складання ТЗ Узгодження ТЗ із зацікавленими сторонами Складання плану та розрахунок розробки					
Доведення техніко- економічного обґрунтування Аналіз ТЗ та визначення пріоритетних аспектів розробки Доведення та уточнення загального обсягу робіт, строків виконання та витрат	Розробник	1	3	110	330,00
Теоретична розробка	Розробник	1	4	110,00	440,00
Практична реалізація	Розробник	1	9	110,00	990,00
Доробка проєкту	Розробник	1	3	110,00	330,00
Заключна стадія	Розробник	1	2	110,00	220,00
Всього заробітна плата					2640,00

Для розробки програмного забезпечення для обчислення, аналізу платежів та графіка погашення кредитів, витрати діляться на послідовність етапів:

а) витрати на основну заробітну плату виконавців, сума часткових заробітних плат за кожну роботу яка наведена в таблиці (1.4)

$$ЗП_{\text{основна}} = 2640,00$$

б) додаткова заробітна плата. Її можна обчислити за формулою:

$$ЗП_{\text{додаткова}} = 0,2 * ЗП_{\text{основна}} \quad (1.1)$$

$$ЗП_{\text{додаткова}} = 528$$

Таким чином загальний фонд заробітної плати, що обчислюється за формулою:

$$\Phi ЗП = ЗП_{\text{основна}} + ЗП_{\text{додаткова}} \quad (1.2)$$

$$\Phi ЗП = 3168$$

а) обов'язкові відрахування на заробітну плату (єдиний соціальний внесок – 22,0%). Таким чином обов'язкові відрахування складають:

$$\text{Відр} = \Phi ЗП * 0,22 = 696$$

б) накладні витрати розраховуємо за формулою:

$$\text{НВ} = 0,4 * ЗП_{\text{основна}} \quad (1.3)$$

$$\text{НВ} = 2640,00 * 0,4 = 1056 \text{ грн.}$$

Результати розрахунку інвестиційних витрат приведені в таблиці 1.5.

Таблиця 1.5 – Перелік інвестиційних витрат

Найменування елементу витрат	Сума, грн.
1	2
Капітальні затрати	-
Основна заробітна плата виконавців	2640,00
Додаткова заробітна плата виконавців	528,00
Нарахування на заробітну плату (єдиний соціальний внесок)	696,00
Накладні витрати	1056,00
Витрат всього	4920,00

На реалізацію даного проєкту потрібно витрати 4920,00 гривень.

Експлуатаційними витратами є такі витрати, які забезпечують нормальне функціонування певного технічного рішення в період його експлуатації. Величина експлуатаційних витрат в розрахунку за 1 рік може бути прогнозовано за формулою:

$$E = k \cdot Ц \cdot \beta = k \cdot A \cdot СП \cdot \beta \quad (1.4)$$

де Ц — ціна реалізації нової розробки, якщо вона була визначена раніше, грн./од.,

k — коефіцієнт, який ураховує витрати на амортизацію, електроенергію, обслуговування, ремонти тощо. Для обчислювальної техніки $k = 0,5 \dots 0,7$.

A — коефіцієнт, який ураховує прогнозований прибуток та податки, які повинен сплачувати виробник $A \approx 1,7 \dots 2,3$;

СП — собівартість нової розробки;

β — доля часу, який витрачає працівник на обслуговування технічної розробки в загальному часі своєї роботи. ($\beta \approx 0,07$)

$$E = 0,5 * 1,7 * 4920,00 * 0,07 = 292,74 \text{ грн.}$$

Створення цього програмного забезпечення потребує інвестицій у розмірі 4920,00 грн. Основні витрати пов'язані з оплатою розробника, оскільки необхідне програмне забезпечення вже було придбано заздалегідь. Ці невеликі витрати гарантують надійний доступ до широкого спектру можливостей, які надає програмне забезпечення. Зазначені розрахунки демонструють доцільність таких інвестицій.

Економічна доцільність розробки програмного забезпечення для обчислення, аналізу платежів та графіка погашення кредитів передбачається в отриманні прибутку за рахунок донатів за використання програмного забезпечення. Прогнозуємо отримання плати 10000,00 грн на рік.

$$T_{ок} = \frac{СП}{E\phi} \quad (1.5)$$

$$T_{ок} = 4920,00 / 10000,00 = 0,5 \text{ року}$$

Коефіцієнт економічної ефективності розраховується за формулою:

$$Кеф = \frac{1}{T_{ок}} \quad (1.6)$$

$$Кеф = 1 / 0,5 = 2$$

Таблиця 1.6 – Основні економічні показники

Показники	Значення
1	2
Собівартість проекту, грн.	4920,00
Сума умовно-річної економії, грн.	10 000
Строк окупності, років	0,5
Коефіцієнт економічної ефективності	2

Розробка програмного забезпечення для обчислення, аналізу платежів та графіка погашення кредитів потребує капіталовкладень в сумі 4920,00 грн. Капіталовкладення окупляться протягом півроку, а коефіцієнт економічної ефективності становить 2. Зроблені розрахунки дозволяють зробити висновок про доцільність реалізації програмного забезпечення для обчислення, аналізу платежів та графіка погашення кредитів.

РОЗДІЛ 2. ТЕХНІЧНЕ ЗАВДАННЯ НА ПРОЄКТУВАННЯ

2.1. Загальні положення.

2.1.1 Назва: Програмне забезпечення для обчислення, аналізу платежів та графіка погашення кредитів.

2.1.2 Замовник: Фізична особа, яка планує отримати кредит та бажає аналізувати його умови та графік погашення.

2.1.3 Розробник: Жолобняк Антон Анатолійович.

2.1.4 Термін реалізації: 15.11.2023 – 26.01.2024.

2.2. Призначення і цілі створення розробки.

2.2.1 Призначення: Розробка програмного забезпечення для обчислення, аналізу платежів та графіка погашення кредитів з метою надання користувачам інструменту для ефективного управління фінансами та планування погашення кредитів. Використання програмного забезпечення для розрахунку суми щомісячних платежів, загальної суми відсотків за кредитом та інших фінансових показників, пов'язаних з кредитуванням.

2.2.2 Цілі:

- Полегшення розрахунків для користувачів, які беруть кредити.
- Підвищення ефективності процесу погашення кредитів.
- Забезпечити можливість введення параметрів кредиту та отримання розрахунків платежів.
- Надати засіб для аналізу фінансових параметрів кредиту.
- Побудувати графік погашення кредиту для візуалізації та планування.

2.3. Характеристика об'єкта автоматизації.

2.3.1 Об'єкт автоматизації: Програмне забезпечення для обчислення, аналізу та візуалізації платежів за кредитами.

2.3.2 Характеристика об'єкта:

- Можливість введення основних параметрів кредиту.
- Аналіз фінансових показників кредиту.
- Генерація таблиці погашення кредиту.

2.4. Вимоги до розробки.

2.4.1 Мова програмування: C#.

2.4.2 Інтегроване середовище розробки: Visual Studio.

2.4.3 Кросплатформенність: Підтримка операційних систем Windows.

2.4.4 Технічні вимоги:

Використання .NET Framework.

Забезпечення зручного та інтуїтивно зрозумілого інтерфейсу користувача.

2.5. Склад і зміст робіт по створенню розробки

2.5.1 Аналіз вимог:

Визначення функціональних та технічних вимог до розробки.

2.5.2 Проєктування:

Розробка структури бази даних у вигляді таблиці погашення кредиту.

Визначення архітектури програми та користувацького інтерфейсу.

2.5.3 Реалізація:

Написання програмного коду відповідно до визначених специфікацій.

2.5.4 Тестування:

Виявлення та виправлення помилок та недоліків.

2.5.5 Впровадження:

Запуск системи для загального використання.

2.6. Порядок контролю і приймання розробки.

2.6.1 Контроль якості.

Проведення тестів для перевірки правильності роботи системи.

2.6.2 Приймання системи:

Передача замовнику завершеної та перевіреної системи для використання.

2.7. Вимоги до складу і змісту робіт із підготовки до введення програмного забезпечення в дію

2.7.1 Навчання персоналу:

Підготовка навчальних матеріалів та проведення тренінгів для персоналу.

2.7.2 Підтримка при впровадженні:

Забезпечення технічної підтримки та вирішення можливих проблем під час впровадження.

2.8. Вимоги до документації

Склад і зміст документів, що розроблюються, визначаються у відповідності з вимогами ДСТУ 3008-2015 [6] та узгоджуються Розробником і Замовником в процесі розробки. Всі документи розробляються державною мовою.

Перелік документів, що будуть створені:

- Технічне завдання;
- Інструкція користувача.

Вся документація повинна бути підготовлена і передана як в друкованому, так і в електронному вигляді (у форматі текстового файлу).

2.9. Джерела розробки

Дане Технічне завдання розроблено на основі наступних документів та інформаційних матеріалів:

- ДСТУ 3008-2015. Документація. Звіти у сфері науки і техніки. Структура та правила оформлювання. [6]
- ДСТУ Б А.2.4-4:2009 «Основні вимоги до проектної та робочої документації. [7]
- Довідники та нормативні документи, що регламентують процес обчислення, аналізу платежів та графіка погашення кредитів.
- Рекомендації фахівців у галузі інформаційних технологій.
- Досвід розробки аналогічних систем.

РОЗДІЛ 3. ОПИС КОМПЛЕКСУ ЗАДАЧ ДЛЯ АВТОМАТИЗАЦІЇ

3.1 Інформаційне забезпечення розробки

Для розробки програми обране середовище програмування Visual Studio 2022, також для створення програми знадобився об'єкт Windows Forms, який є в Visual Studio та мова програмування C#.

Visual Studio — це інтегроване середовище розробки (IDE), створене Microsoft для різних мов програмування (рисунок 3.1).



Рисунок 3.1 – Visual Studio 2022

Нижче наведено деякі загальні функції та вдосконалення, які очікувалися або були представлені у Visual Studio 2022.

64-розрядна версія: однією з значних змін став перехід до 64-розрядної архітектури, яка дозволяє IDE обробляти більші та складніші проекти з покращеною продуктивністю та стабільністю.

Підтримка .NET 6 Visual Studio 2022 забезпечує розширену підтримку .NET 6, останньої версії довгострокової підтримки (LTS) .NET Framework.

Ключові функції Visual Studio 2022 включають:

Підтримка мов програмування:

Visual Studio 2022 підтримує ряд мов програмування, включаючи C#, C++, Visual Basic, F#, Python, JavaScript та інші. Це робить його універсальним інструментом для розробки різноманітних додатків.

Інтегровані редагувальники коду:

Має потужні та інтуїтивно зрозумілі редагувальники коду з підтримкою автодоповнення, виправлення помилок, рефакторингу та інших корисних функцій.

Візуальний дизайнер інтерфейсу користувача:

Для розробки Windows-додатків мовою XAML та створення інтерфейсу за допомогою WPF або UWP технологій.

Управління версіями:

Вбудована підтримка систем контролю версій, таких як Git. Це дозволяє розробникам працювати над проектами в команді та відстежувати зміни в коді.

Відлагодження та профілювання:

Включає різноманітні інструменти для відлагодження коду, включаючи точкові зупинки, крокування, відстеження значень змінних та інші. Також є інструменти профілювання для аналізу продуктивності.

Підтримка Azure DevOps:

Інтегрованість з хмарними послугами Azure DevOps для управління життєвим циклом розробки (ALM), включаючи планування, відстеження задач, автоматизацію збірки і розгортання.

Розширення та додатки:

Можливість встановлення розширень та додатків для розширення функціональності Visual Studio згідно з конкретними потребами розробника.

Cross-Platform Development:

Підтримка розробки кросплатформених застосунків для операційних систем, таких як Windows, Linux та macOS.

Інструменти для розробки вебдодатків:

Підтримка розробки веб-додатків за допомогою технологій ASP.NET, HTML, CSS, JavaScript тощо.

Інструменти для розробки мобільних додатків:

Підтримка розробки мобільних додатків для платформ Android та iOS.

Загалом, Visual Studio 2022 — це потужна та універсальна IDE, яку можна використовувати для розробки різноманітних програм. Якщо для розробки програмного забезпечення необхідно потужну та багатofункціональну IDE, Visual Studio 2022 — чудовий варіант.

Додаткові відомості про Visual Studio 2022:

Він доступний у кількох версіях, включаючи Community, Professional і Enterprise. Версія Community є безкоштовною для використання, тоді як версії Professional і Enterprise мають додаткові функції та призначені для професійних розробників.

Windows Forms (WinForms) — це бібліотека класів графічного інтерфейсу користувача (GUI), що входить до складу Microsoft .NET Framework, .NET Framework або Mono, надає платформу для написання клієнтських програм для настільних, портативних і планшетних ПК.

Основні функції та поняття Windows Forms:

- Візуальний дизайн:
 - Інтерфейс перетягування для розміщення елементів керування на формах.
 - Visual Studio Designer для ефективного проектування форм.
 - Вікно властивостей для налаштування зовнішнього вигляду та поведінки елементів керування.
- Елементи керування:
 - Широкий набір вбудованих елементів керування, включаючи кнопки, мітки, текстові поля, списки, меню тощо.

- Можливість створювати власні елементи керування для конкретних потреб.

- Події:

- Модель програмування, керована подіями.

- Код реагує на взаємодію користувача з елементами керування та подіями програми.

- Прив'язка даних:

- Підключіть елементи керування до джерел даних для відображення та редагування даних.

- Візуальна спадковість:

- Створюйте нові форми на основі наявних форм, успадковуючи властивості та функціональність.

Використання Windows Forms:

- Бізнес програми:

- CRM системи.

- Програмне забезпечення для управління запасами.

- Системи торгових точок.

- Інструменти продуктивності:

- Текстові процесори.

- Електронні таблиці.

- Поштові клієнти.

- Службові програми:

- Конвертери файлів.

- Засоби моніторингу системи.

- Програмне забезпечення для резервного копіювання.

- Навчальне програмне забезпечення:

- Інтерактивні підручники.

- Додатки-вікторини.

Переваги Windows Forms.

- Простота використання:
- Інструменти візуального дизайну спрощують процес розробки.
- Багатий контрольний набір:
- Доступний широкий вибір стандартних елементів керування.
- Зріла технологія:
- Налаштована і широко використовувана структура.
- Хороша продуктивність:
- Ефективний для настільних програм.
- Інтеграція з іншими технологіями .NET:
- Бездоганно працює з іншими бібліотеками та фреймворками .NET.

Міркування:

- Орієнтований на робочий стіл:
- В основному розроблено для настільних програм Windows.
- Веброзробка:
- Не ідеально підходить для веб-додатків (краще підходять такі альтернативи, як ASP.NET або Blazor).
- Мобільна розробка:
- Не підходить безпосередньо для розробки програм для мобільних пристроїв (більш підходять Xamarin або нативні платформи).

У Windows Forms використовуються елементи керування (controls), які представляють собою різні графічні об'єкти, такі як кнопки, тексти, списки.

Основні елементи керування у Windows Forms включають:

Button (Кнопка): Використовується для створення кнопок, які виконують певні дії при натисканні.

```
Button myButton = new Button();
myButton.Text = "Натисніть мене";
myButton.Click += new EventHandler(myButton_Click);
```

TextBox (Текстове поле): Використовується для введення тексту користувачем.

```
TextBox myTextBox = new TextBox();
```

```
string userInput = myTextBox.Text;
```

Label (Мітка): Використовується для відображення тексту або іншої інформації.

```
Label myLabel = new Label();
```

```
myLabel.Text = "Це мітка";
```

ListBox (Список): Використовується для відображення списку елементів, які можуть бути вибрані користувачем.

```
ListBox myListBox = new ListBox();
```

```
myListBox.Items.Add("Елемент 1");
```

```
myListBox.Items.Add("Елемент 2");
```

ComboBox (Комбінований список): Поєднує можливості текстового поля та списку.

```
ComboBox myComboBox = new ComboBox();
```

```
myComboBox.Items.Add("Варіант 1");
```

```
myComboBox.Items.Add("Варіант 2");
```

DataGridView (Таблиця даних): Використовується для відображення та редагування табличних даних.

```
DataGridView myDataGridView = new DataGridView();
```

```
// Додавання стовпців та рядків до таблиці
```

PictureBox (Графічний об'єкт): Використовується для відображення графічних зображень.

```
PictureBox myPictureBox = new PictureBox();
```

```
myPictureBox.Image = Image.FromFile("image.jpg");
```

Це елемент керування у Windows Forms, які будуть використовувати при розробці програмного забезпечення обчислення, аналізу платежів та графіка погашення кредитів. Кожен елемент має свої властивості та події, які можна налаштовувати для взаємодії з користувачем та реалізації різних функцій.

Загалом Windows Forms є надійним вибором для розробки програмного забезпечення, пропонуючи поєднання простоти використання, потужного набору елементів керування та переваг у продуктивності.

Для розробки програмного забезпечення обчислення, аналізу платежів та графіка погашення кредитів обрано об'єктно-орієнтовану мову програмування C#.

C# (C Sharp) – це об'єктно-орієнтована мова програмування, розроблена компанією Microsoft. Вона є частиною платформи .NET і зазвичай використовується для розробки різноманітних типів програмних додатків, включаючи веб-додатки, десктоп-додатки, ігри, мобільні додатки та інші. Основні характеристики C# включають:

1. Об'єктно-орієнтований підхід.

C# підтримує об'єктно-орієнтоване програмування (ООП), що дозволяє створювати модульний, структурований та легко розширюваний код.

2. Інтеграція з .NET

Мова спеціально розроблена для платформи .NET, що дозволяє розробникам використовувати багато вбудованих бібліотек та сервісів .NET.

3. Безпека.

C# має вбудовані механізми безпеки, такі як система управління пам'яттю та перевірка типів, які сприяють запобіганню багатьом типам помилок.

4. Спрощена синтаксис.

Синтаксис C# подібний до синтаксису інших C-подібних мов, таких як C++ та Java, що полегшує вивчення для розробників, які вже володіють досвідом у цих мовах.

5. Мультипарадигмальність.

C# підтримує різні парадигми програмування, включаючи процедурне програмування, об'єктно-орієнтоване програмування та функціональне програмування.

6. Підтримка асинхронного програмування.

Мова має вбудовану підтримку асинхронного програмування за допомогою ключових слів `async` та `await`.

7. Інтерфейси та делегати.

Використання інтерфейсів та делегатів дозволяє розробникам використовувати принципи поліморфізму та забезпечує гнучкість коду.

8. Велика спільнота та документація.

C# має велику активну спільноту розробників та розгорнуту документацію, що полегшує вирішення проблем та вивчення нових концепцій.

9. Мова для розробки веб-додатків.

Спеціальні фреймворки, такі як ASP.NET, дозволяють використовувати C# для розробки веб-додатків та служб.

10. Інструменти розробки:

Використання розширених інтегрованих середовищ розробки, таких як Visual Studio, яке надає багато корисних інструментів для розробників C#.

Ці характеристики роблять C# потужною та високорівневою мовою програмування, що підходить для розробки програмного забезпечення.

3.2 Алгоритмізація та реалізація комплексу задач автоматизації

Для розробки програмного забезпечення обчислення, аналізу платежів та графіка погашення кредитів створено концептуальну модель у вигляді діаграми потоків даних (DFD). Для створення діаграми використовувався онлайн-інструмент Draw.io. (рисунок 3.2)

Draw.io – це онлайн-інструмент для створення схем, діаграм, малюнків та інших графічних елементів. Він надає можливість легко створювати, редагувати та спільно використовувати різні види діаграм. Цей інструмент може бути корисним для проєктування процесів, розробки програмного забезпечення, створення організаційних схем, мереж та багато іншого. До основних функцій Draw.io належать: підтримка широкого спектру типів моделей, простий у використанні інтерфейс, широкий вибір шаблонів і бібліотек, можливість редагування діаграм у співпраці, можливість експорту діаграм у різні формати.

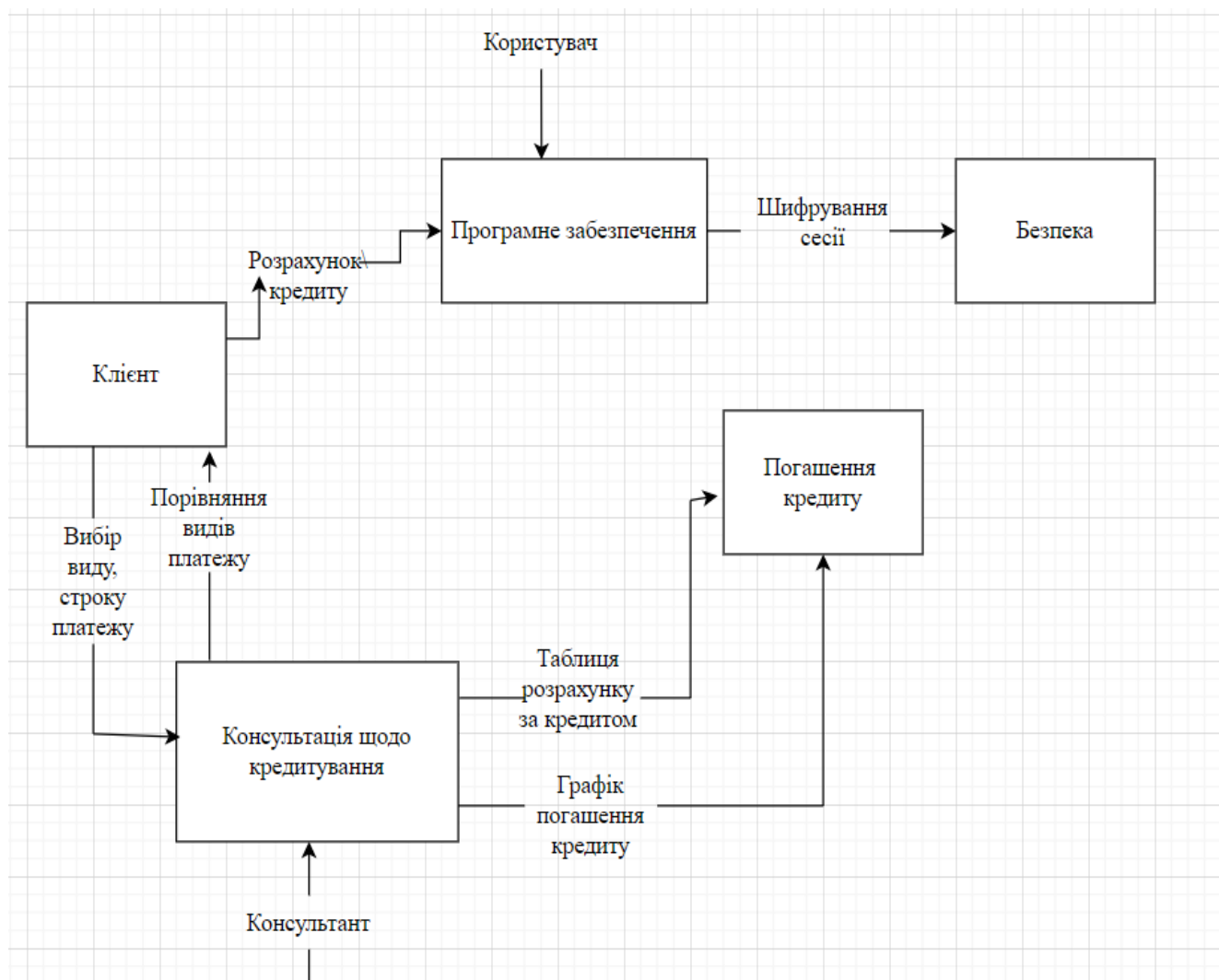


Рисунок 3.2 – Початкова концептуальна діаграма програмного забезпечення

Зовнішні сутності:

Клієнт – це юридична чи фізична особа, яка користується послугами банку. Він може розрахувати суму щомісячного платежу з допомогою програмного забезпечення, при цьому перебуваючи в будь-якому місці з можливістю доступу до ноутбука або комп'ютера.

Програмне забезпечення – це програмний додаток для обчислення, аналізу платежів та графіка погашення кредитів, допомагає підібрати види кредитування до розрахунку кредиту. Він доступний для користувачів пристроїв на базі операційної системи Windows.

Погашення кредиту дозволяє проаналізувати та переглянути графік погашення кредиту виходячи із різних видів платежу та строку.

Безпека. Шифрування файлів, що зберігаються на пристроях користувачів або в хмарних сховищах, для запобігання несанкціонованому доступу до важливої інформації..

Консультація щодо кредитування. Тут вирішується питання про те, який вигідний кредит обрати клієнту, а також розраховується щомісячні виплати за кредитом.

Для реалізації програмного забезпечення для обчислення, аналізу платежів та графіка погашення кредитів створений класичний додаток із назвою Kredit для пристроїв на базі операційної системи Windows. В якості користувацького інтерфейсу виступає власноруч розроблений за допомогою можливостей Windows Forms інтерфейс.

Програма складається з форми на якій розміщено можливість вибору:

- виду кредитного платежу;
- грошової одиниці;
- строку кредитування;
- початок першого платежу.

Передбачено задання:

- суми кредиту;
- процентної ставки;
- одноразової комісії.

Розрахунок проводиться в залежності від заданих користувачем-клієнтом даних. Для візуалізації, аналізу для погашення кредиту проведені розрахунки записуються в базу даних.

Одною із важливих додаткових вимог до програмного забезпечення для аналізу для обчислення, аналізу платежів та графіка погашення кредитів є швидкість функціонування, тому було вирішено не загроможувати базу даних великою кількістю таблиць. В разі потреби у майбутньому можна розширити базу даних додатковими таблицями.

База даних складається з таблиці в якій містять такі поля для записів (рисунок 3.3):

- №;
- Дата платежу;
- Сума платежу;
- Сума погашення основного боргу;
- Сума нарахованих основних процентів;
- Залишок боргу.

	№	Дата платежу	Сума платежу	Сума погашення основного боргу	Сума нарахованих процентів	Залишок боргу
*						

Рисунок 3.3 – Поля Таблиці

Програмне забезпечення використовує наступні бібліотеки (рисунок 3.4):

```

C# WindowsFormsApp1
1  using System;
2  using System.Collections.Generic;
3  using System.ComponentModel;
4  using System.Data;
5  using System.Drawing;
6  using System.Linq;
7  using System.Text;
8  using System.Threading.Tasks;
9  using System.Windows.Forms;
10 using System.IO;
11

```

Рисунок 3.4 – Бібліотеки

- System.

Містить фундаментальні і базові класи, що визначають часто використовувані типи значень, події і обробники подій, інтерфейси і атрибути.

- System.Collections.Generic.

Містить інтерфейси і класи, що визначають універсальні колекції, які дозволяють користувачам створювати строго типізовані колекції, що забезпечують підвищену продуктивність і безпеку типів у порівнянні з неуніверсальними строго типізованими колекціями.

- `System.ComponentModel`.

Надає класи, використовувані для реалізації поведінки компонентів і елементів управління під час розробки і під час виконання. Цей простір імен містить базові класи і інтерфейси для реалізації атрибутів і перетворювачів типів, прив'язки до джерел даних і ліцензування компонентів.

- `System.Data`.

Надає доступ до класів, які представляють архітектуру ADO.NET. ADO.NET дозволяє створювати компоненти, які ефективно керують даними з декількох джерел даних.

- `System.Drawing`.

Надає доступ до основних графічних функцій GDI +.

- `System.Linq`.

Надає класи та інтерфейси, що підтримують запити з використанням LINQ.

- `System.Text`.

Містить класи, які представляють кодування ASCII і Юнікода; абстрактні базові класи для перетворення блоків знаків в блоки байтів і назад; допоміжний клас, який обробляє і форматує об'єкти `String`, не створюючи проміжні екземпляри `String`.

- `System.Threading.Tasks`.

Надає типи, які спрощують роботу з написання паралельного і асинхронного коду

- `System.Windows.Forms`.

Містить класи для створення додатків Windows, які дозволяють найбільш ефективно використовувати розширені можливості призначеного для користувача інтерфейсу, які надаються операційною системою Microsoft Windows.

- System.Windows.Forms.DataVisualization.Charting.

Містить методи і властивості для елемента управління Windows Forms Chart.

- System.IO.

Ввід та вивід потоків даних.

Структура проекту Kredit зображена на рисунку 3.5.

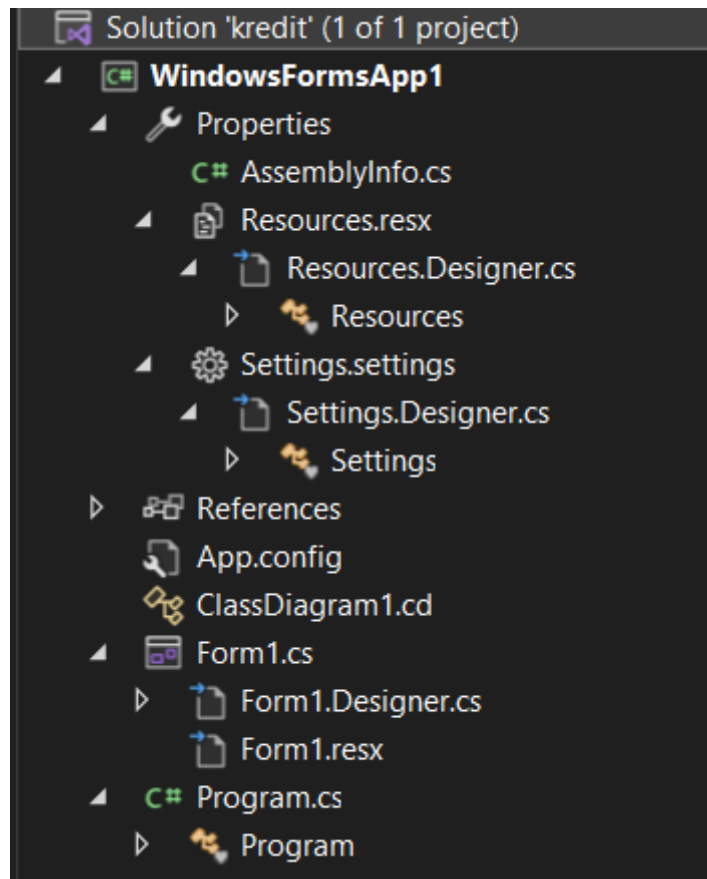


Рисунок 3.5 — Структура файлів проекту

Для забезпечення чіткої структури та взаємодії між компонентами програми, для полегшення розробки, тестування та управління проектом було спроектовано UML діаграму класів (Додаток А). UML (Unified Modeling Language) діаграма класів є важливим інструментом для моделювання структури програмного забезпечення та його компонентів. Вона надає графічний засіб відображення класів, їх взаємодії та відносин, що допомагає інтерпретувати та розуміти архітектуру програми.

На діаграмах класів буде показано різноманітні класи, які утворюють систему і їх взаємозв'язки. Діаграми класів називають «статичними діаграмами», оскільки на них показано класи разом з методами і атрибутами, а також статичний взаємозв'язок між ними: те, яким класам «відомо» про існування яких класів, і те, які класи «є частиною» інших класів.

Клас визначає атрибути і методи набору об'єктів. Всі об'єкти цього класу (екземпляри цього класу) мають спільну поведінку і однаковий набір атрибутів (кожен з об'єктів має свій власний набір значень). Іноді замість назви «клас» використовують назву «тип», але, слід зауважити, що ці назви описують різні речі: тип є загальнішим визначенням. У UML класи позначаються прямокутниками з назвою класу, у цих прямокутниках у вигляді двох «відсіків» може бути показано атрибути і операції класу.

Атрибути.

У UML атрибути показуються щонайменше назвою, також може бути показано їх тип, початкове значення і інші властивості. Крім того, атрибути може бути показано з областю видимості атрибута:

- + відповідає публічним (public) атрибутам;
- # відповідає захищеним (protected) атрибутам;
- відповідає приватним (private) атрибутам.

Операції.

Операції (методи) також показуються принаймні назвою, крім того, може бути показано їх параметри і типи значень, які буде повернуто. Операції, як і атрибути, може бути показано з областю видимості:

- + відповідає публічним (public) операціям;
- # відповідає захищеним (protected) операціям;
- відповідає приватним (private) операціям.

Для створення діаграми класів у Visual Studio 2022 виконано наступні кроки:

1. Запустити Visual Studio 2022.

2. Створити проєкт.
3. Додати клас.
4. Відкрити діаграму класу (рисунок 3.6).
5. Перетягнути класи з панелі «Клас» на поверхню діаграми класів.
6. Додати учасників і зв'язки:
7. Зберегти діаграму класу:

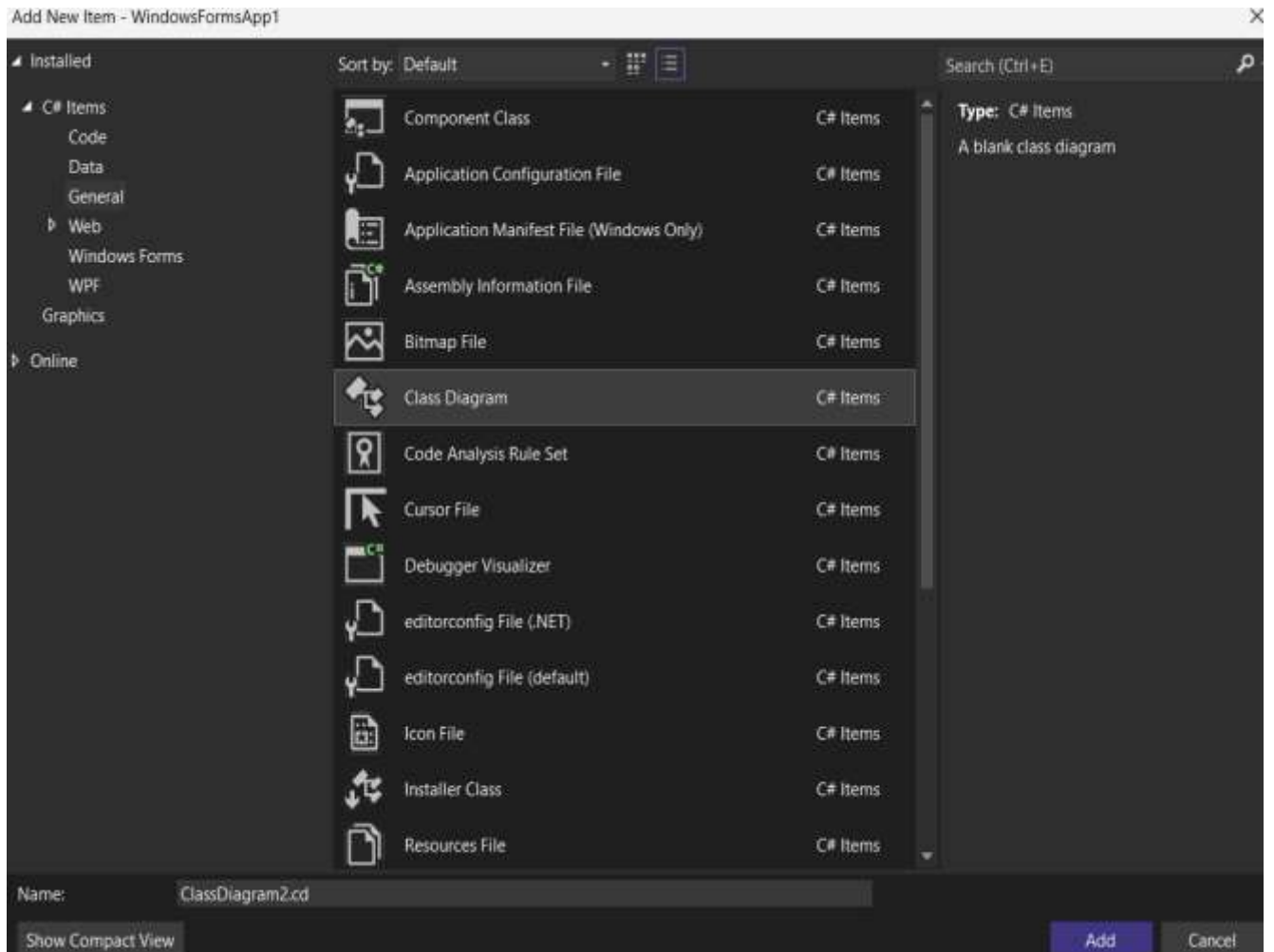


Рисунок 3.6 – Додавання до проєкта Діаграми класів

Програма не має розмежування прав доступу до даних, кожний користувач має можливість як переглядати дані, так і редагувати їх. Оскільки всі користувачі програми мають абсолютно рівні права, було вирішено не реалізовувати систему авторизації.

3.3 Розробка програмного забезпечення

Розробка програмного забезпечення для обчислення, аналізу платежів та графіка погашення кредитів розпочалася із проектування головної Form додатка Kredit. Код представлено Додатку Б.

Головна форма містить в собі такі компоненти надписи Label (рисунок 3.7):

- Label1 – Параметри кредиту.

Код властивостей Label1:

```
// label1
//
this.label1.AutoSize = true;
this.label1.Font = new System.Drawing.Font("Microsoft Sans Serif", 20.25F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte) (204)));
this.label1.ForeColor = System.Drawing.SystemColors.ControlDarkDark;
this.label1.Location = new System.Drawing.Point(155, 11);
this.label1.Margin = new System.Windows.Forms.Padding(4, 0, 4, 0);
this.label1.Name = "label1";
this.label1.Size = new System.Drawing.Size(339, 39);
this.label1.TabIndex = 7;
this.label1.Text = "Параметри кредиту";
this.label1.Click += new System.EventHandler(this.label1_Click);
```

- Label2 – Сума кредиту.

Код властивостей Label2:

```
// label2
//
this.label2.Font = new System.Drawing.Font("Microsoft Sans Serif", 12F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte) (204)));
this.label2.ForeColor = System.Drawing.SystemColors.ControlDarkDark;
this.label2.Location = new System.Drawing.Point(16, 96);
this.label2.Margin = new System.Windows.Forms.Padding(4, 0, 4, 0);
this.label2.Name = "label2";
this.label2.Size = new System.Drawing.Size(177, 32);
this.label2.TabIndex = 8;
this.label2.Text = "Сума кредиту:";
```

- Label3 – Строк кредиту.

Код властивостей Label3:

```
// label3
//
this.label3.Font = new System.Drawing.Font("Microsoft Sans Serif", 12F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte) (204)));
this.label3.ForeColor = System.Drawing.SystemColors.ControlDarkDark;
this.label3.Location = new System.Drawing.Point(16, 140);
this.label3.Margin = new System.Windows.Forms.Padding(4, 0, 4, 0);
this.label3.Name = "label3";
this.label3.Size = new System.Drawing.Size(164, 32);
```

```
this.label3.TabIndex = 9;
this.label3.Text = "Строк кредиту:";
```

- Label4 – Процентна ставка.

Код властивостей Label4:

```
// label4
//
this.label4.Font = new System.Drawing.Font("Microsoft Sans Serif", 12F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte) (204)));
this.label4.ForeColor = System.Drawing.SystemColors.ControlDarkDark;
this.label4.Location = new System.Drawing.Point(465, 96);
this.label4.Margin = new System.Windows.Forms.Padding(4, 0, 4, 0);
this.label4.Name = "label4";
this.label4.Size = new System.Drawing.Size(223, 32);
this.label4.TabIndex = 10;
this.label4.Text = "Процентна ставка: ";
```

- Label5 – Одноразова комісія.

Код властивостей Label5:

```
// label5
//
this.label5.Font = new System.Drawing.Font("Microsoft Sans Serif", 12F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte) (204)));
this.label5.ForeColor = System.Drawing.SystemColors.ControlDarkDark;
this.label5.Location = new System.Drawing.Point(465, 138);
this.label5.Margin = new System.Windows.Forms.Padding(4, 0, 4, 0);
this.label5.Name = "label5";
this.label5.Size = new System.Drawing.Size(300, 32);
this.label5.TabIndex = 11;
this.label5.Text = "Одноразова комісія:";
this.label5.Click += new System.EventHandler(this.label5_Click);
```

- Label6 – Вид платежу.

Код властивостей Label6:

```
// label6
//
this.label6.Font = new System.Drawing.Font("Microsoft Sans Serif", 12F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte) (204)));
this.label6.ForeColor = System.Drawing.SystemColors.ControlDarkDark;
this.label6.Location = new System.Drawing.Point(16, 54);
this.label6.Margin = new System.Windows.Forms.Padding(4, 0, 4, 0);
this.label6.Name = "label6";
this.label6.Size = new System.Drawing.Size(153, 32);
this.label6.TabIndex = 12;
this.label6.Text = "Вид платежу:";
this.label6.Click += new System.EventHandler(this.label6_Click);
```

- Label7 – Початок виплат.

Код властивостей Label7:

```

// label7
//
this.label7.Font = new System.Drawing.Font("Microsoft Sans Serif", 12F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte) (204)));
this.label7.ForeColor = System.Drawing.SystemColors.ControlDarkDark;
this.label7.Location = new System.Drawing.Point(465, 57);
this.label7.Margin = new System.Windows.Forms.Padding(4, 0, 4, 0);
this.label7.Name = "label7";
this.label7.Size = new System.Drawing.Size(180, 32);
this.label7.TabIndex = 13;
this.label7.Text = "Початок виплат:";
this.label7.Click += new System.EventHandler(this.label7_Click);

```

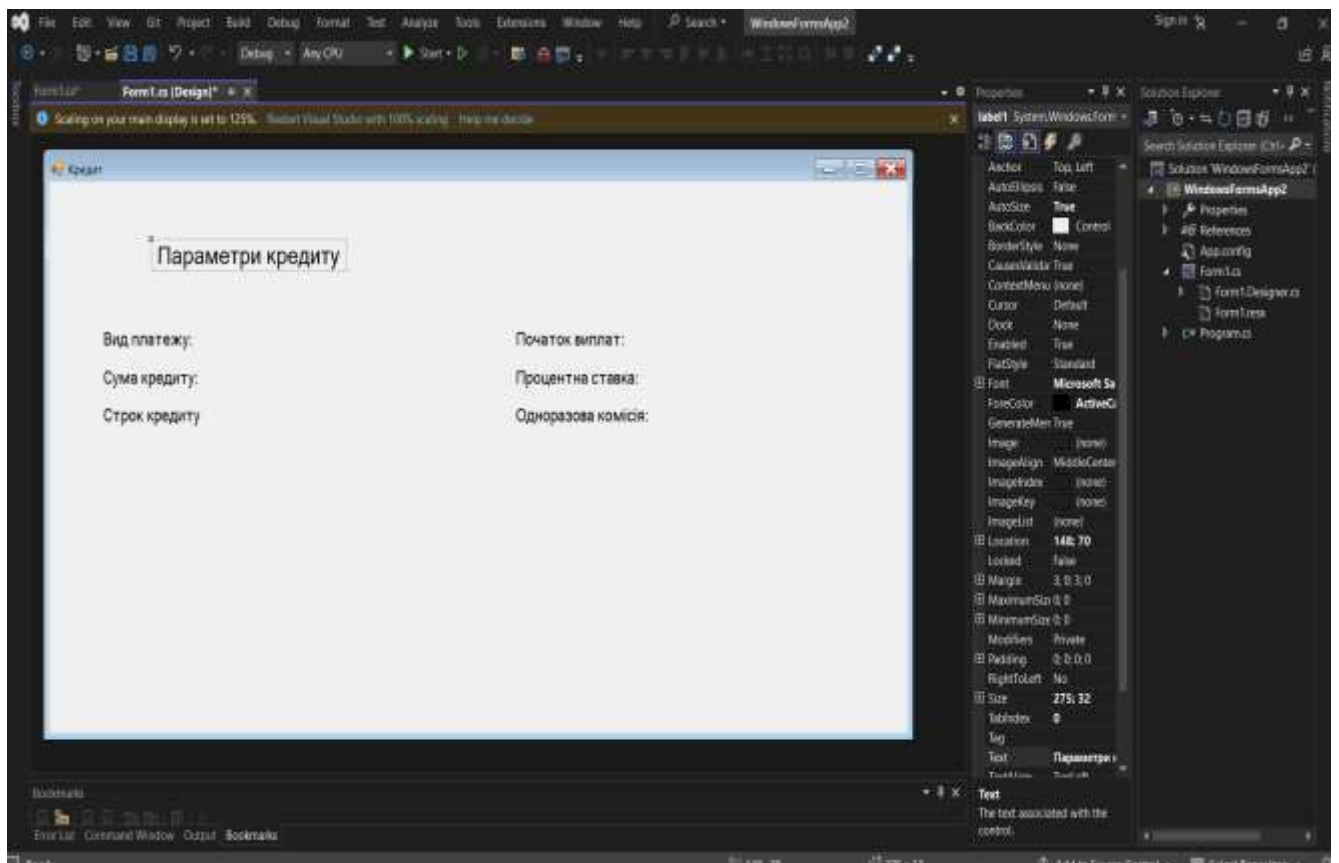


Рисунок 3.7 – Розміщення компонентів Label

Для списків, обрано компонент комбінованого списку `comboBox` (рисунок 3.8):

- `comboBox1` – Вибір грошової одиниці: ₴, \$, €, ¥.

Код `comboBox1`:

```

// comboBox1
//
this.comboBox1.DisplayMember = "грн.";
this.comboBox1.Font = new System.Drawing.Font("Microsoft Sans Serif", 11.25F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte) (204)));
this.comboBox1.ForeColor = System.Drawing.SystemColors.ControlDarkDark;
this.comboBox1.FormattingEnabled = true;

```

```

this.comboBox1.IntegralHeight = false;
this.comboBox1.Items.AddRange(new object[] {
    "e",
    "$",
    "€",
    "¥"});
this.comboBox1.Location = new System.Drawing.Point(345, 96);
this.comboBox1.Margin = new System.Windows.Forms.Padding(4);
this.comboBox1.Name = "comboBox1";
this.comboBox1.Size = new System.Drawing.Size(111, 32);
this.comboBox1.TabIndex = 21;
this.comboBox1.Tag = "";
this.comboBox1.Text = "e";
this.comboBox1.SelectedIndexChanged += new
System.EventHandler(this.comboBox1_SelectedIndexChanged);

```

- **comboBox2** – Вибір одноразової комісії: % від суми кредиту, не враховується.

Код **comboBox2**:

```

// comboBox2
//
this.comboBox2.Font = new System.Drawing.Font("Microsoft Sans Serif", 11.25F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)204));
this.comboBox2.ForeColor = System.Drawing.SystemColors.ControlDarkDark;
this.comboBox2.FormattingEnabled = true;
this.comboBox2.IntegralHeight = false;
this.comboBox2.Items.AddRange(new object[] {
    "% від суми кредиту",
    "не враховується"});
this.comboBox2.Location = new System.Drawing.Point(863, 138);
this.comboBox2.Margin = new System.Windows.Forms.Padding(4);
this.comboBox2.Name = "comboBox2";
this.comboBox2.Size = new System.Drawing.Size(224, 32);
this.comboBox2.TabIndex = 22;
this.comboBox2.Text = "% від суми кредиту";

```

- **comboBox3** – Вибір процентної ставки: % на рік.

Код **comboBox3**:

```

// comboBox3
//
this.comboBox3.Font = new System.Drawing.Font("Microsoft Sans Serif", 11.25F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)204));
this.comboBox3.ForeColor = System.Drawing.SystemColors.ControlDarkDark;
this.comboBox3.FormattingEnabled = true;
this.comboBox3.IntegralHeight = false;
this.comboBox3.Items.AddRange(new object[] {
    "% на рік"});
this.comboBox3.Location = new System.Drawing.Point(801, 94);
this.comboBox3.Margin = new System.Windows.Forms.Padding(4);
this.comboBox3.Name = "comboBox3";
this.comboBox3.Size = new System.Drawing.Size(111, 32);
this.comboBox3.TabIndex = 23;
this.comboBox3.Text = "% на рік";

```

- `comboBox4` – Обрання строку кредиту: місяць, рік.

Код `comboBox4`:

```
// comboBox4
//
this.comboBox4.Font = new System.Drawing.Font("Microsoft Sans Serif", 11.25F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)204));
this.comboBox4.ForeColor = System.Drawing.SystemColors.ControlDarkDark;
this.comboBox4.FormattingEnabled = true;
this.comboBox4.IntegralHeight = false;
this.comboBox4.Items.AddRange(new object[] {
"місяць",
"рік"});
this.comboBox4.Location = new System.Drawing.Point(331, 138);
this.comboBox4.Margin = new System.Windows.Forms.Padding(4);
this.comboBox4.Name = "comboBox4";
this.comboBox4.Size = new System.Drawing.Size(111, 32);
this.comboBox4.TabIndex = 24;
this.comboBox4.Text = "місяць";
```

- `comboBox5` – Вибір виду платежу: диференційований, ануїтетний;

Код `comboBox5`:

```
// comboBox5
//
this.comboBox5.Font = new System.Drawing.Font("Microsoft Sans Serif", 11.25F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)204));
this.comboBox5.ForeColor = System.Drawing.SystemColors.ControlDarkDark;
this.comboBox5.FormattingEnabled = true;
this.comboBox5.IntegralHeight = false;
this.comboBox5.Items.AddRange(new object[] {
"диференційований",
"ануїтетний"});
this.comboBox5.Location = new System.Drawing.Point(201, 54);
this.comboBox5.Margin = new System.Windows.Forms.Padding(4);
this.comboBox5.Name = "comboBox5";
this.comboBox5.Size = new System.Drawing.Size(255, 32);
this.comboBox5.TabIndex = 25;
this.comboBox5.Text = "диференційований";
this.comboBox5.SelectedIndexChanged += new
System.EventHandler(this.comboBox5_SelectedIndexChanged);
```

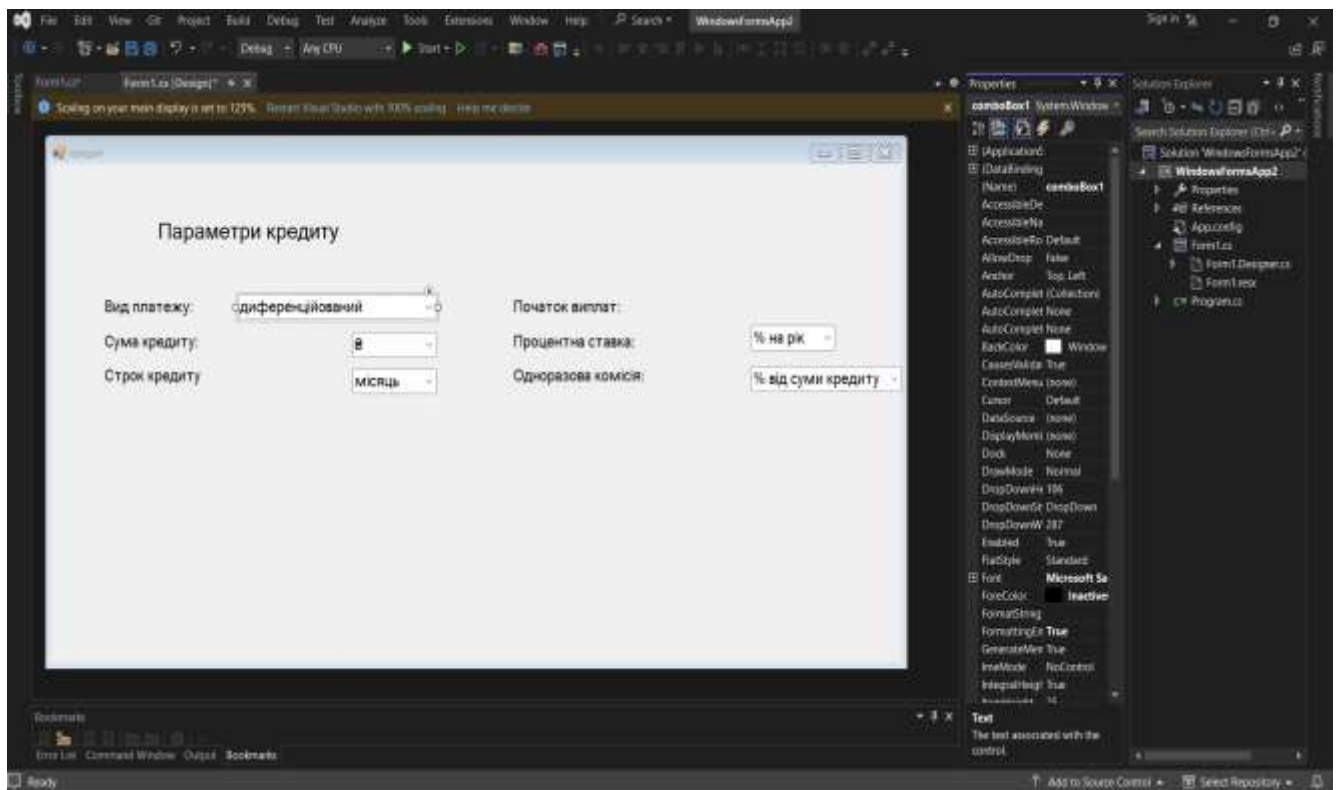


Рисунок 3.8 – Розміщення компонентів comboBox

Для задання користувачем параметрів введення обрано компоненти `numericUpDown` (рисунок 3.9). Для коректного введення даних задано властивість `DecimalPlaces` – 2. Елемент `NumericUpDown` представляє користувачеві вибір числа з певного діапазону:

- `numericUpDown1` – Вибір суми кредиту.

Код `numericUpDown1`:

```
// numericUpDown1
//
this.numericUpDown1.DecimalPlaces = 2;
this.numericUpDown1.Font = new System.Drawing.Font("Microsoft Sans Serif", 12F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte) 204));
this.numericUpDown1.ForeColor = System.Drawing.SystemColors.ControlDarkDark;
this.numericUpDown1.Location = new System.Drawing.Point(201, 94);
this.numericUpDown1.Margin = new System.Windows.Forms.Padding(4);
this.numericUpDown1.Maximum = new decimal(new int[] {
276447231,
23283,
0,
0});
this.numericUpDown1.Name = "numericUpDown1";
this.numericUpDown1.Size = new System.Drawing.Size(136, 30);
this.numericUpDown1.TabIndex = 31;
this.numericUpDown1.ValueChanged += new
System.EventHandler(this.numericUpDown1_ValueChanged);
```

- numericUpDown2 – Вибір одноразової комісії.

Код numericUpDown2:

```
// numericUpDown2
//
this.numericUpDown2.DecimalPlaces = 2;
this.numericUpDown2.Font = new System.Drawing.Font("Microsoft Sans Serif", 12F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte) 204));
this.numericUpDown2.ForeColor = System.Drawing.SystemColors.ControlDarkDark;
this.numericUpDown2.Location = new System.Drawing.Point(759, 138);
this.numericUpDown2.Margin = new System.Windows.Forms.Padding(4);
this.numericUpDown2.Name = "numericUpDown2";
this.numericUpDown2.Size = new System.Drawing.Size(96, 30);
this.numericUpDown2.TabIndex = 32;
```

- numericUpDown3 – Вибір процентної ставки.

Код numericUpDown3:

```
// numericUpDown3
//
this.numericUpDown3.DecimalPlaces = 2;
this.numericUpDown3.Font = new System.Drawing.Font("Microsoft Sans Serif", 12F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte) 204));
this.numericUpDown3.ForeColor = System.Drawing.SystemColors.ControlDarkDark;
this.numericUpDown3.Location = new System.Drawing.Point(691, 96);
this.numericUpDown3.Margin = new System.Windows.Forms.Padding(4);
this.numericUpDown3.Name = "numericUpDown3";
this.numericUpDown3.Size = new System.Drawing.Size(96, 30);
this.numericUpDown3.TabIndex = 33;
this.numericUpDown3.ValueChanged += new
System.EventHandler(this.numericUpDown3_ValueChanged);
//
```

- numericUpDown4 – Вибір строку кредиту.

Код numericUpDown4:

```
// numericUpDown4
//
this.numericUpDown4.DecimalPlaces = 2;
this.numericUpDown4.Font = new System.Drawing.Font("Microsoft Sans Serif", 12F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte) 204));
this.numericUpDown4.ForeColor = System.Drawing.SystemColors.ControlDarkDark;
this.numericUpDown4.Location = new System.Drawing.Point(201, 138);
this.numericUpDown4.Margin = new System.Windows.Forms.Padding(4);
this.numericUpDown4.Maximum = new decimal(new int[] {
276447231,
23283,
0,
0});
this.numericUpDown4.Name = "numericUpDown4";
this.numericUpDown4.Size = new System.Drawing.Size(121, 30);
this.numericUpDown4.TabIndex = 34;
```

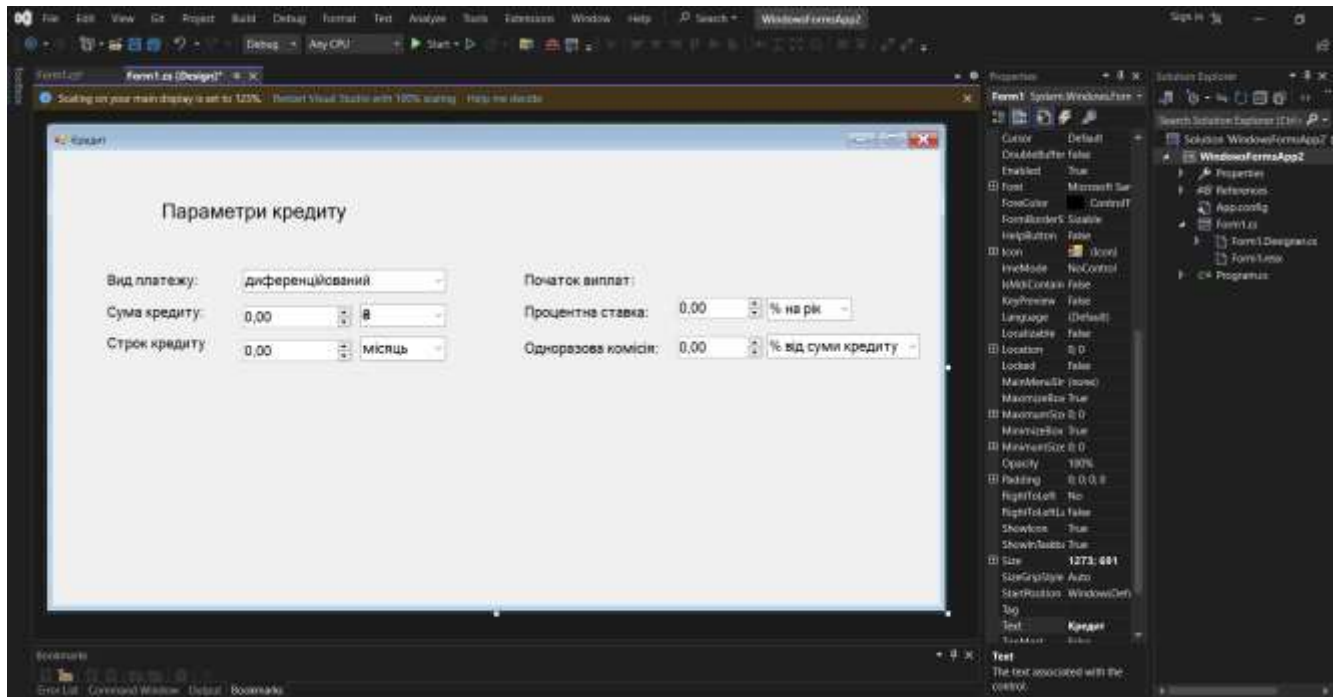


Рисунок 3.9 – Розміщення компонентів numericUpDown

Для обрання користувачем початку виплати кредиту використано компонент DateTimePicker (рисунок 3.10). Він дозволяє користувачеві вибрати дату і/або час з візуального календаря або попередньо заданого часу.

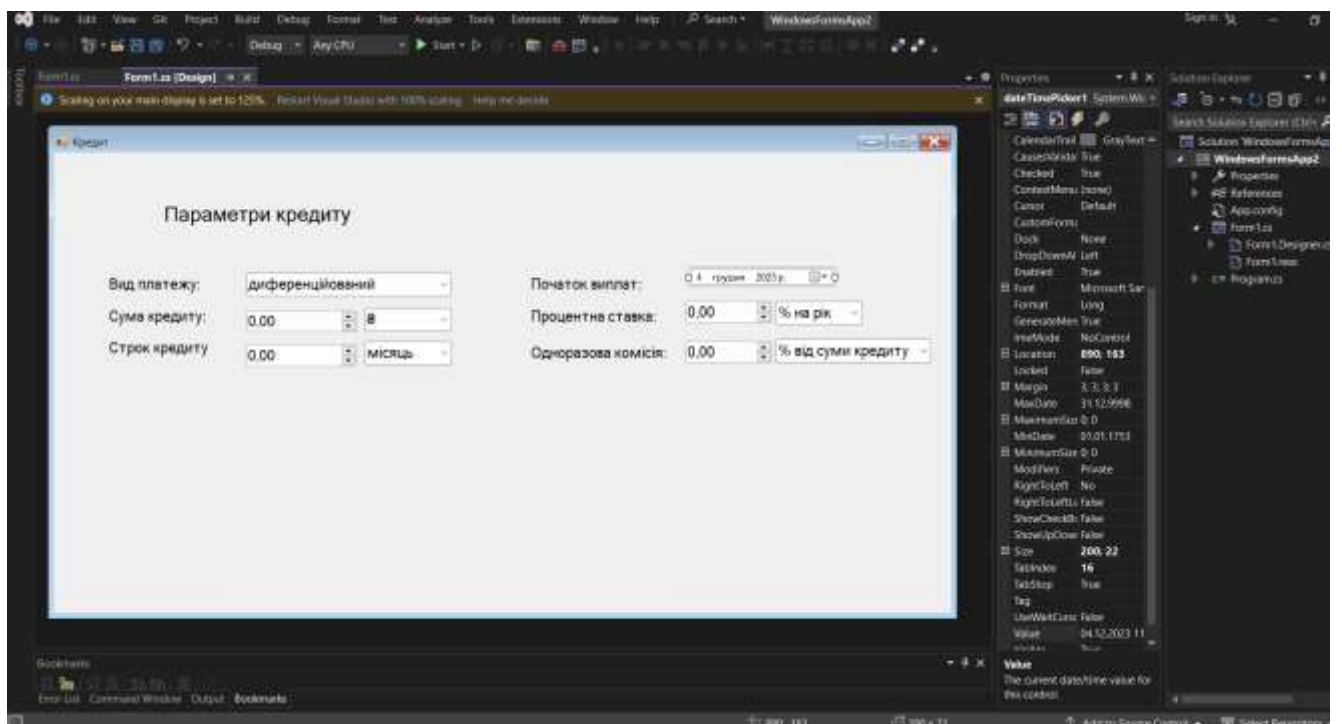


Рисунок 3.10 – Розміщення компонентів DateTimePicker

Інтерфейс DateTimePicker має текстове поле, де можна відобразити обрану дату та/або час, а також кнопку або стрілочку, за допомогою якої можна відкривати вікно для вибору дати і часу з календаря або годинника.

Код DateTimePicker:

```
// dateTimePicker1
//
this.dateTimePicker1.CalendarTitleBackColor =
System.Drawing.SystemColors.Control;
this.dateTimePicker1.CalendarTitleForeColor =
System.Drawing.SystemColors.ControlDark;
this.dateTimePicker1.CalendarTrailingForeColor =
System.Drawing.SystemColors.HighlightText;
this.dateTimePicker1.Font = new System.Drawing.Font("Microsoft Sans Serif", 12F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte) 204));
this.dateTimePicker1.Format = System.Windows.Forms.DateTimePickerFormat.Custom;
this.dateTimePicker1.Location = new System.Drawing.Point(651, 55);
this.dateTimePicker1.Margin = new System.Windows.Forms.Padding(4);
this.dateTimePicker1.Name = "dateTimePicker1";
this.dateTimePicker1.Size = new System.Drawing.Size(135, 30);
this.dateTimePicker1.TabIndex = 37;
```

Для занесення даних в базу даних у вигляді таблиці, а також аналізу графіка погашення кредиту використано компонент dataGridView (рисунк 3.11).

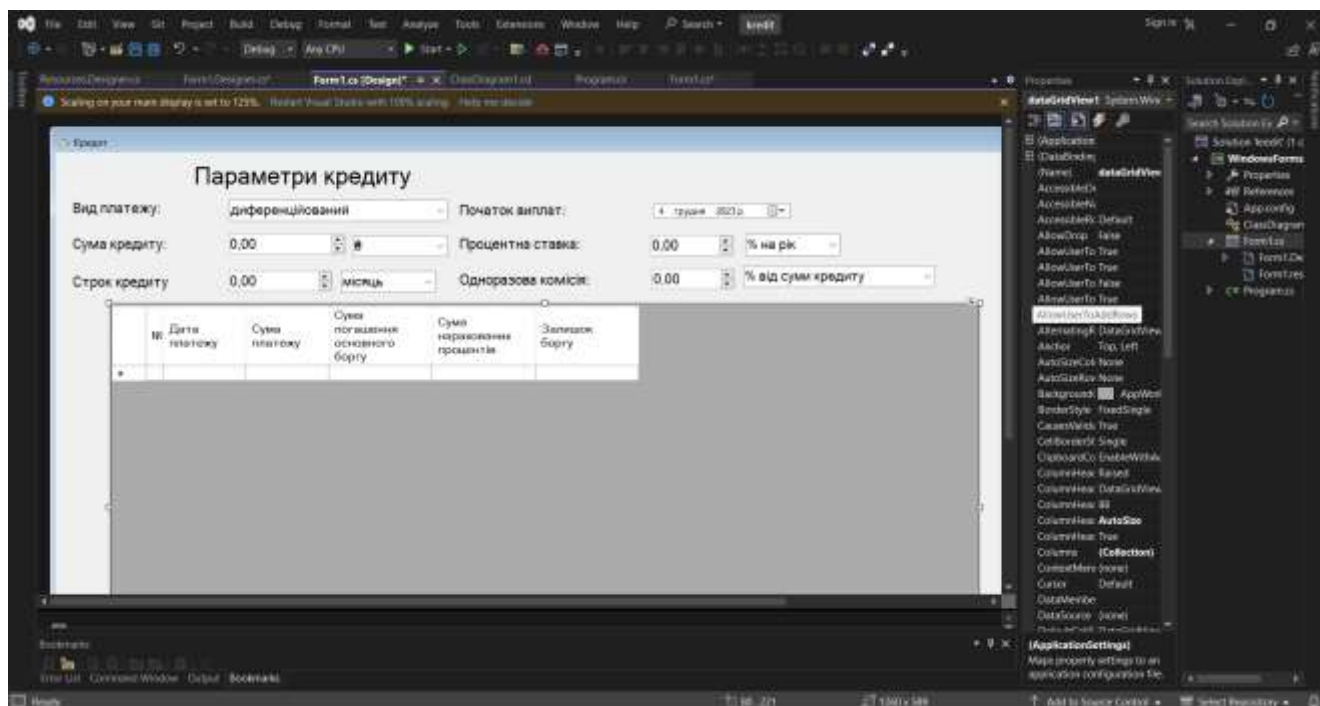


Рисунок 3.11 – Розміщення компонентів dataGridView

Код dataGridView1:

```
// dataGridView1
//
```

```

        this.dataGridView1.ColumnHeadersHeightSizeMode =
System.Windows.Forms.DataGridViewColumnHeadersHeightSizeMode.AutoSize;
        this.dataGridView1.Columns.AddRange(new
System.Windows.Forms.DataGridViewColumn[] {
            this.Column1,
            this.Column2,
            this.Column3,
            this.Column4,
            this.Column5,
            this.Column6});
        this.dataGridView1.Enabled = false;
        this.dataGridView1.Location = new System.Drawing.Point(64, 177);
        this.dataGridView1.Margin = new System.Windows.Forms.Padding(4);
        this.dataGridView1.Name = "dataGridView1";
        this.dataGridView1.RowHeadersWidth = 51;
        this.dataGridView1.Size = new System.Drawing.Size(1008, 471);
        this.dataGridView1.TabIndex = 35;
        this.dataGridView1.Visible = false;
        this.dataGridView1.CellContentClick += new
System.Windows.Forms.DataGridViewCellEventHandler(this.dataGridView1_CellContentClick_1
);

        //
        // Column1
        //
        this.Column1.HeaderText = "№";
        this.Column1.MinimumWidth = 6;
        this.Column1.Name = "Column1";
        this.Column1.Width = 25;
        //
        // Column2
        //
        this.Column2.FillWeight = 150F;
        this.Column2.HeaderText = "Дата платежу";
        this.Column2.MinimumWidth = 6;
        this.Column2.Name = "Column2";
        this.Column2.Width = 120;
        //
        // Column3
        //
        this.Column3.FillWeight = 150F;
        this.Column3.HeaderText = "Сума платежу";
        this.Column3.MinimumWidth = 6;
        this.Column3.Name = "Column3";
        this.Column3.Width = 120;
        //
        // Column4
        //
        this.Column4.HeaderText = "Сума погашення основного боргу";
        this.Column4.MinimumWidth = 6;
        this.Column4.Name = "Column4";
        this.Column4.Width = 150;
        //
        // Column5
        //
        this.Column5.HeaderText = "Сума нарахування процентів";
        this.Column5.MinimumWidth = 6;
        this.Column5.Name = "Column5";
        this.Column5.Width = 150;
        //
        // Column6
        //
        this.Column6.HeaderText = "Залишок боргу";
        this.Column6.MinimumWidth = 6;
        this.Column6.Name = "Column6";
        this.Column6.Width = 150;

```

Для обчислення платежу за кредитом використано компонент Button (рисунок 3.12). Код розрахунку кредиту представлений в Додатку В.

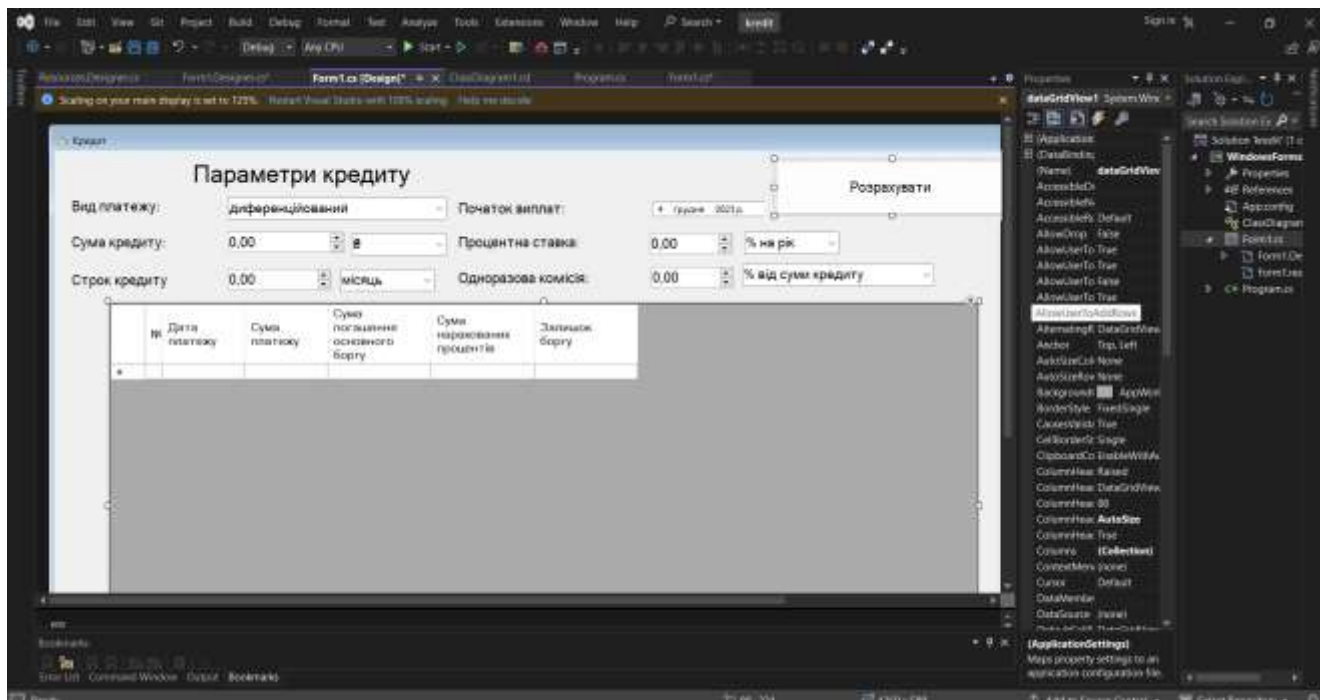


Рисунок 3.12 – Розміщення компонентів Button1

Код властивостей Button1:

```
// button1
//
this.button1.Font = new System.Drawing.Font("Microsoft Sans Serif", 12F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte) (204)));
this.button1.ForeColor = System.Drawing.SystemColors.ControlDarkDark;
this.button1.Location = new System.Drawing.Point(836, 11);
this.button1.Margin = new System.Windows.Forms.Padding(4);
this.button1.Name = "button1";
this.button1.Size = new System.Drawing.Size(269, 58);
this.button1.TabIndex = 29;
this.button1.Text = "Розрахувати";
this.button1.UseVisualStyleBackColor = true;
    this.button1.Click += new System.EventHandler(this.button1_Click);
```

Для розрахунку платежів за кредитом використано цикли. А також `dataGridView1.RowCount = Convert.ToInt32(srok)` вказує на присвоєння властивості `RowCount` об'єкта `dataGridView1` значення, яке отримується з конвертації рядка `srok` в ціле число типу `int`.

`Convert.ToInt32(srok)`: Цей вираз конвертує рядок `srok` в ціле число типу `int`. Якщо `srok` представляє собою допустиме ціле число у текстовому вигляді, то це

число буде використано; в іншому випадку, може виникнути виключення, якщо рядок містить неприпустимі символи або не може бути коректно конвертований в ціле число.

`dataGridView1.RowCount = ...`: Після конвертації отримане ціле число присвоюється властивості `RowCount` об'єкта `dataGridView1`. `RowCount` визначає кількість рядків у `DataGridView` (таблиці) і встановлює кількість видимих рядків у таблиці.

Код розрахунку:

```
dataGridView1.RowCount = Convert.ToInt32(srok);

for (int i = 0; i < srok; i++)
{
    dataGridView1.Rows[i].Cells[0].Value = i + 1;

    dataGridView1.Rows[i].Cells[1].Value = day + "." + month + "." + year;
    month++;
    if (month > 12)
    {
        month = 1;
        year++;
    }

    if (vid == "диференційований")
    {
        ss = ss + fullpay;
        if ((i + 1) == 1) { paymonth = (sum * pr); }
        else { paymonth = ((sum - ss) * pr); }
        fullpay = (sum / srok);
        pay = (paymonth + fullpay);
        ostatok = (ostatok - fullpay);
        obsum += pay;
        obproz += paymonth;
        itog = obsum * komiss + obproz;
    }
    else if (vid == "ануїтетний")
    {
        ss = +pay;
        /*

        if ((i + 1) == 1) { paymonth = (sum * pr); }
        else { paymonth = ((sum - ss) * pr); }
        pay = ((pr * Math.Pow((1 + pr), srok)) / (Math.Pow((1 + pr), srok) - 1)) *
sum);

        */
        paymonth = ostatok * pr;
        pay = sum * (pr + pr / (Math.Pow((1 + pr), srok) - 1));
        fullpay = (pay - paymonth);
        fullpaysum = +fullpay;
        ostatok = (ostatok - fullpay);
        obsum = +pay;
        obproz = +paymonth;
        itog = obsum * komiss + obproz;
    }
}
```

Для збереження таблиці використано компонента Button2 (рисунок 3.13)

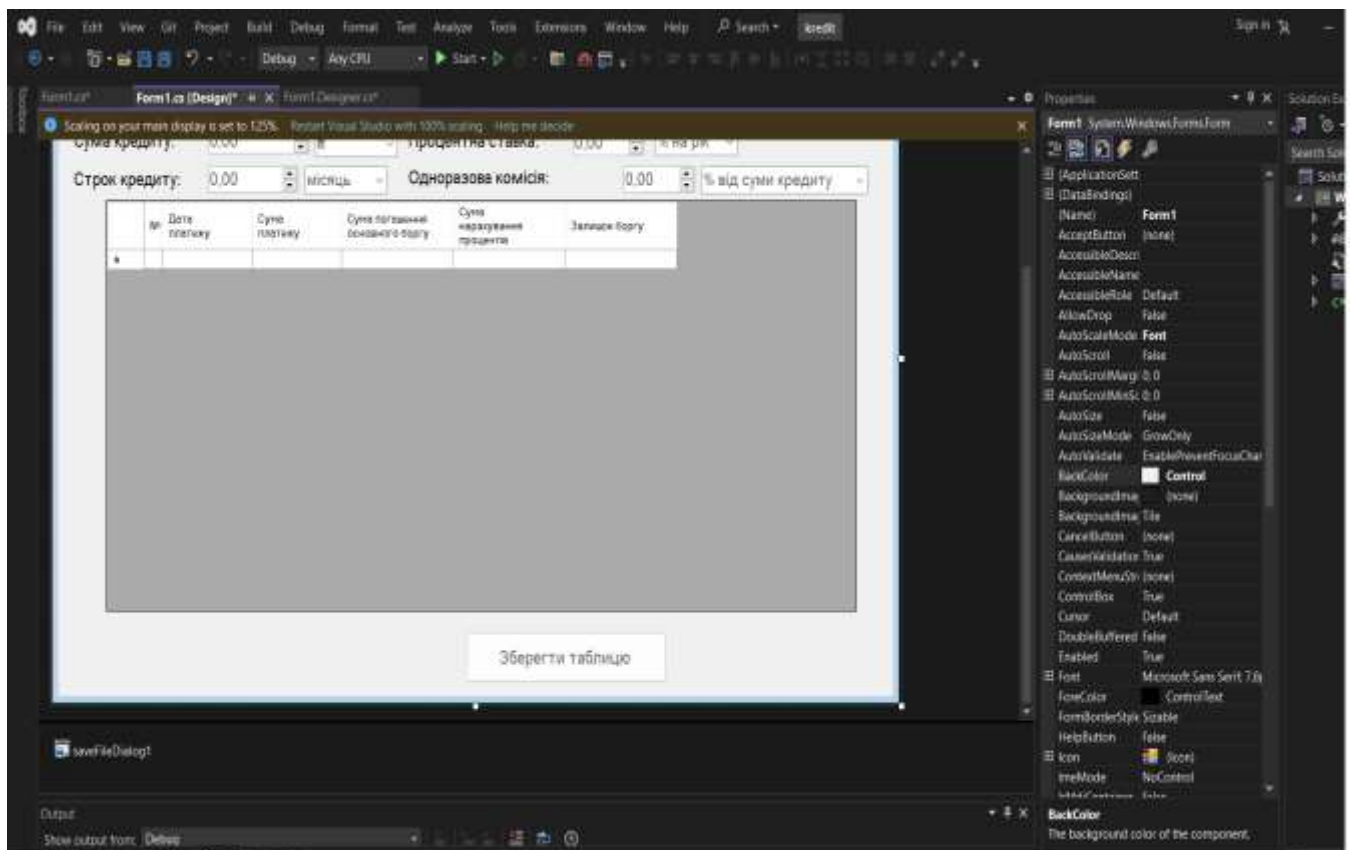


Рисунок 3.13 – Розміщення компонентів Button2

Код Button2:

```
// button2
//
this.button2.Font = new System.Drawing.Font("Microsoft Sans Serif", 12F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte) (204)));
this.button2.ForeColor = System.Drawing.SystemColors.ControlDarkDark;
this.button2.Location = new System.Drawing.Point(547, 672);
this.button2.Margin = new System.Windows.Forms.Padding(4);
this.button2.Name = "button2";
this.button2.Size = new System.Drawing.Size(269, 58);
this.button2.TabIndex = 47;
this.button2.Text = "Зберегти таблицю";
this.button2.UseVisualStyleBackColor = true;
this.button2.Visible = false;
    this.button2.Click += new System.EventHandler(this.button2_Click_1);
```

Для зміни даних використано компонента Button3 (рисунок 3.14).

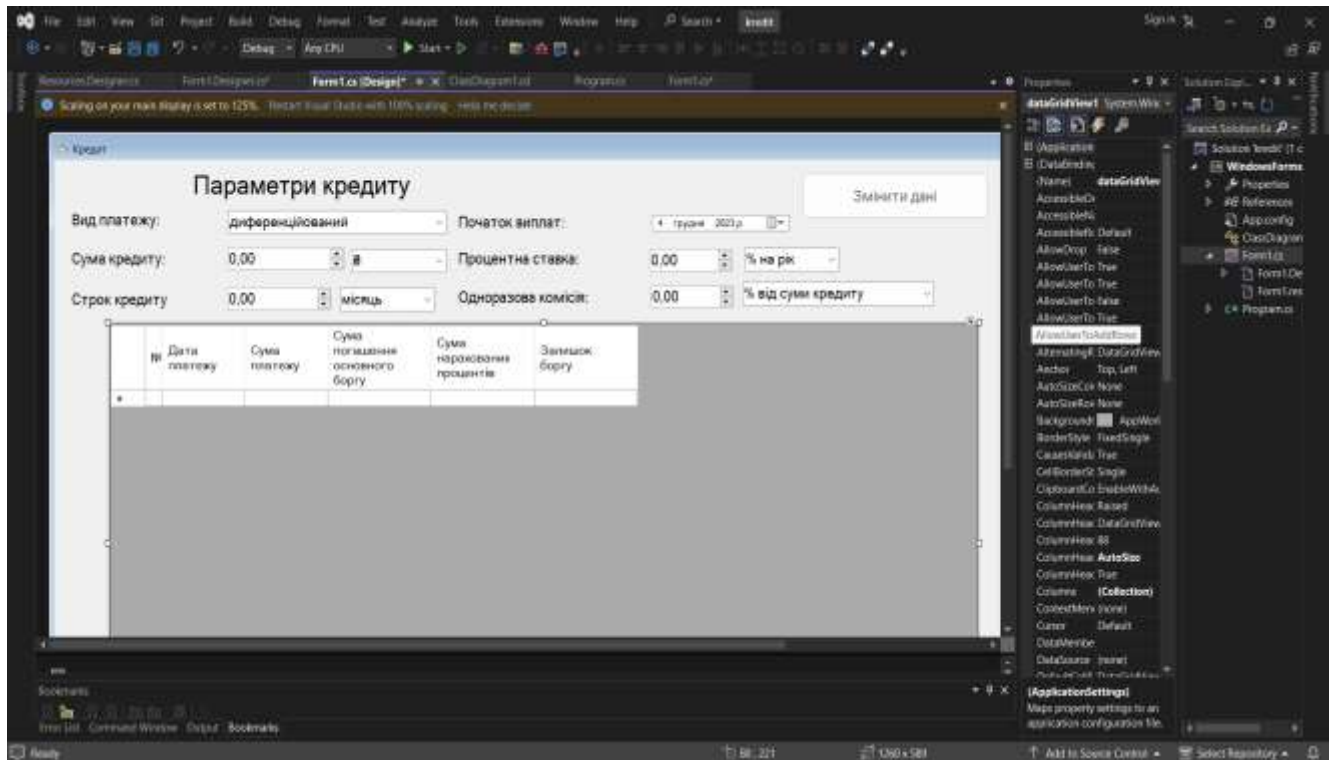


Рисунок 3.14 – Розміщення компонентів Button3

Код Button3:

```
// button3
//
this.button3.Font = new System.Drawing.Font("Microsoft Sans Serif", 12F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte) (204)));
this.button3.ForeColor = System.Drawing.SystemColors.ControlDarkDark;
this.button3.Location = new System.Drawing.Point(836, 9);
this.button3.Margin = new System.Windows.Forms.Padding(4);
this.button3.Name = "button3";
this.button3.Size = new System.Drawing.Size(269, 62);
this.button3.TabIndex = 48;
this.button3.Text = "Змінити дані";
this.button3.UseVisualStyleBackColor = true;
this.button3.Visible = false;
    this.button3.Click += new System.EventHandler(this.button3_Click);
```

Для закриття програми використано компонента Button4 (рисунок 3.15)

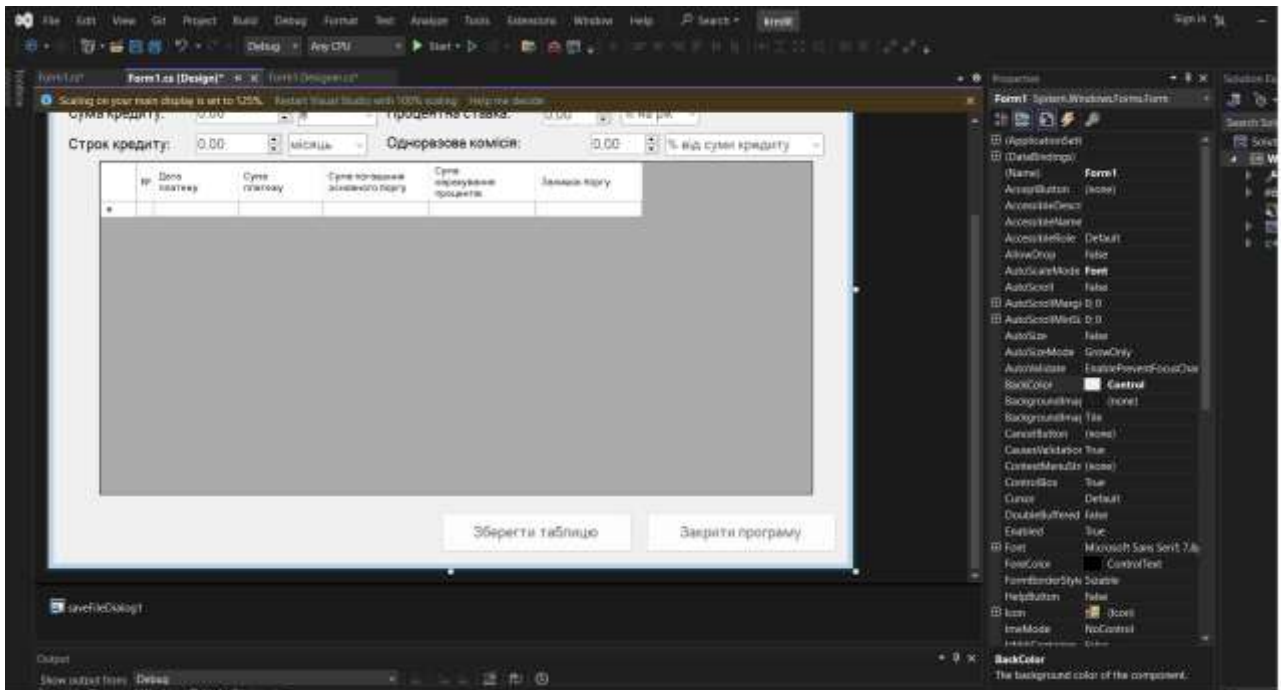


Рисунок 3.15 – Розміщення компонентів Button4

Код Button4:

```
// button4
//
this.button4.Font = new System.Drawing.Font("Microsoft Sans Serif", 12F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte) (204)));
this.button4.ForeColor = System.Drawing.SystemColors.ControlDarkDark;
this.button4.Location = new System.Drawing.Point(836, 672);
this.button4.Margin = new System.Windows.Forms.Padding(4);
this.button4.Name = "button4";
this.button4.Size = new System.Drawing.Size(269, 58);
this.button4.TabIndex = 49;
this.button4.Text = "Закрити програму";
this.button4.UseVisualStyleBackColor = true;
    this.button4.Click += new System.EventHandler(this.button4_Click);
```

3.4 Інструкція користувача

Інструкція користувача для програмного забезпечення потрібна для того, щоб користувачі могли правильно і ефективно використовувати програмне забезпечення. Вона містить інформацію про те, як інсталювати, запускати, налаштовувати та використовувати програмне забезпечення. Інструкції також містить інформацію про функції та можливості програмного забезпечення.

Для роботи з програмним забезпеченням для обчислення, аналізу платежів та графіка погашення кредитів у вигляді програмного додатку необхідно його

інсталювати на комп'ютер, який підтримує операційну систему Windows версії 7, 10, 11, а також Visual Studio з .NET Framework 4.7 і вище.

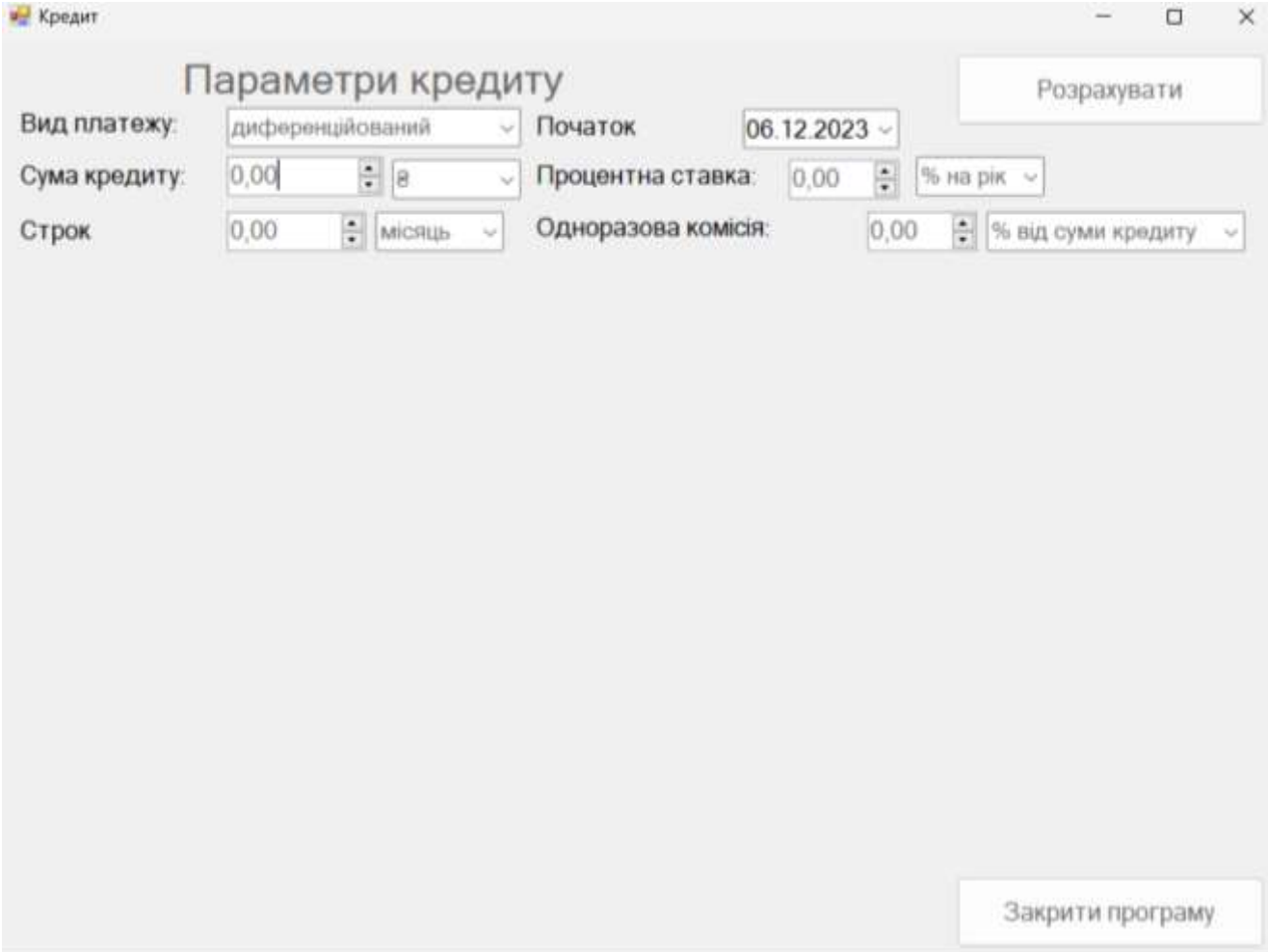
Далі необхідно запуснути програмний додаток клацанням миші по піктограмі додатку (рисунок 3.16)



Kredit

Рисунок 3.16 – Піктограма додатка

Після запуску програми з'являється вікно програми для обчислення, аналізу платежів та графіка погашення кредитів (рисунок 3.17)



Кредит

Параметри кредиту

Вид платежу: диференційований Початок: 06.12.2023

Сума кредиту: 0,00 в Процентна ставка: 0,00 % на рік

Строк: 0,00 місяць Одноразова комісія: 0,00 % від суми кредиту

Розрахувати

Закрити програму

Рисунок 3.17 – Вікно програми

Наступний крок користувачеві потрібно обрати із списку вид платежу за кредитом (рисунок 3.18).

Кредит

Параметри кредиту

Вид платежу: диференційований

Сума кредиту: диференційований

Строк: 0,00 місяць

Початок: 06.12.2023

Процентна ставка: 0,00 % на рік

Одноразова комісія: 0,00 % від суми кредиту

Розрахувати

Закрити програму

Рисунок 3.18 – Вибір виду платежу

У програмі пропонується два основних види платежу за кредитом: диференційований, ануїтетний. В даний час найбільш поширеним варіантом складання графіка платежів за кредитами є варіант ануїтетних платежів. При цьому варіанті сума щомісячної виплати не змінюється протягом усього терміну погашення заборгованості. Але в початковому періоді основна частина внесеної суми списується на погашення відсотків. І лише ближче до кінця терміну кредитного договору сума відсотків зменшується, а гаситися починає основний борг.

При диференційованій системі платежів вигода клієнта очевидна. Сума загального боргу рівномірно поділена на кількість місяців відповідно до терміну, на який береться кредит. Щомісячні платежі складають суми обов'язкового гасіння боргу плюс відсотки на залишок заборгованості. Відповідно, чим більшу суму зміг внести клієнт у звітному періоді, тим більша кількість грошових коштів списалося на гасіння основного боргу. І тим менша сума відсотків буде нарахована в наступному місяці.

Наступним кроком потрібно ввести суму кредиту, яку планується отримати і обрати в якій грошовій одиниці (рисунок 3.19).

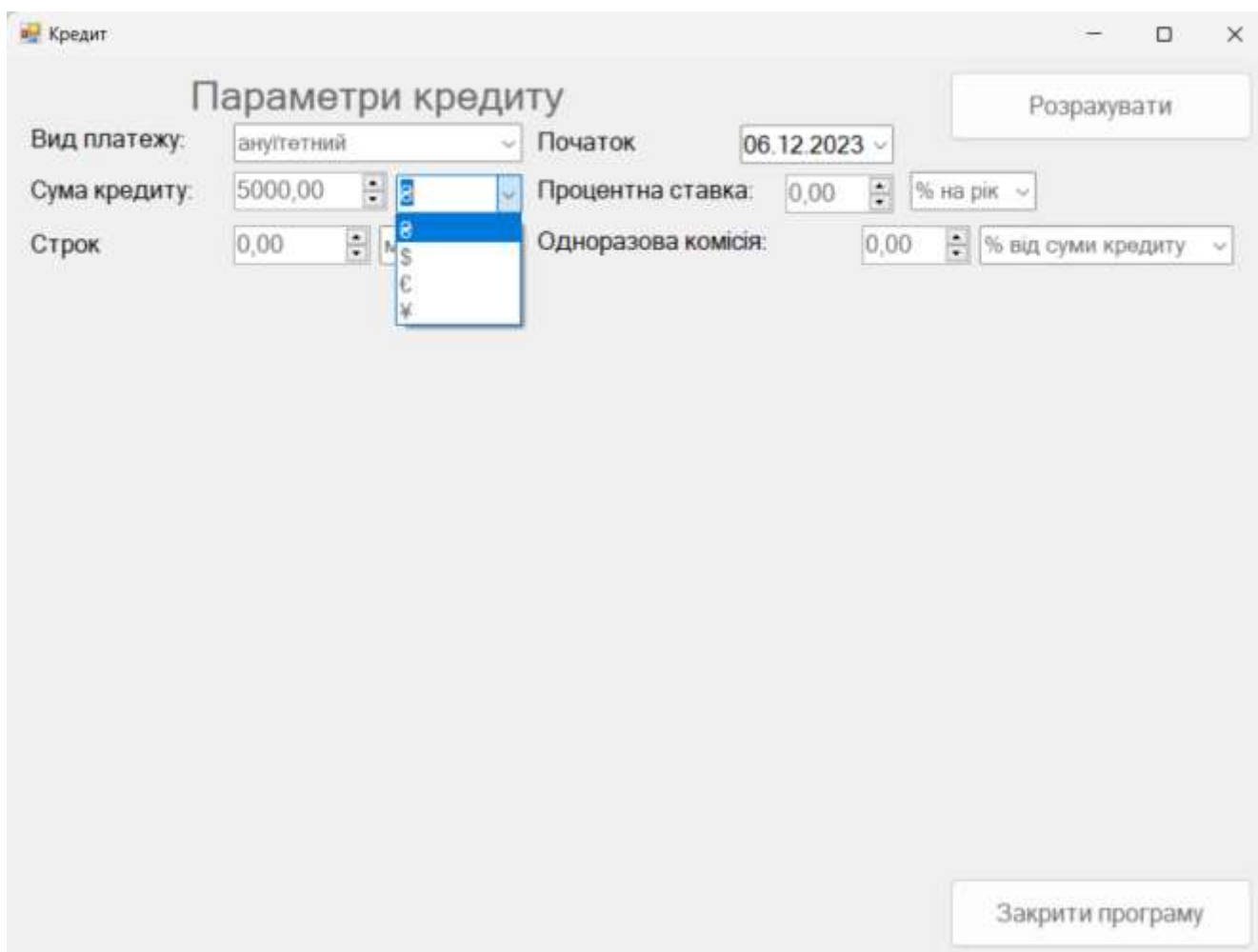


Рисунок 3.19 – Введення суми кредиту та вибір грошової одиниці

Далі користувачеві необхідно ввести строк погашення кредиту та обрати місяць або рік (рисунок 3.20).

Кредит

Параметри кредиту

Розрахувати

Вид платежу: Початок:

Сума кредиту: Процентна ставка: % на рік

Строк: Одноразова комісія: % від суми кредиту

Закрити програму

Рисунок 3.20 – Вибір строку погашення кредиту

Наступним кроком потрібно обрати за допомогою календаря початок сплати кредитної заборгованості (рисунок 3.21).

Кредит

Параметри кредиту

Розрахувати

Вид платежу: Початок:

Сума кредиту: Процентна ставка: % на рік

Строк: Одноразова комісія: % від суми кредиту

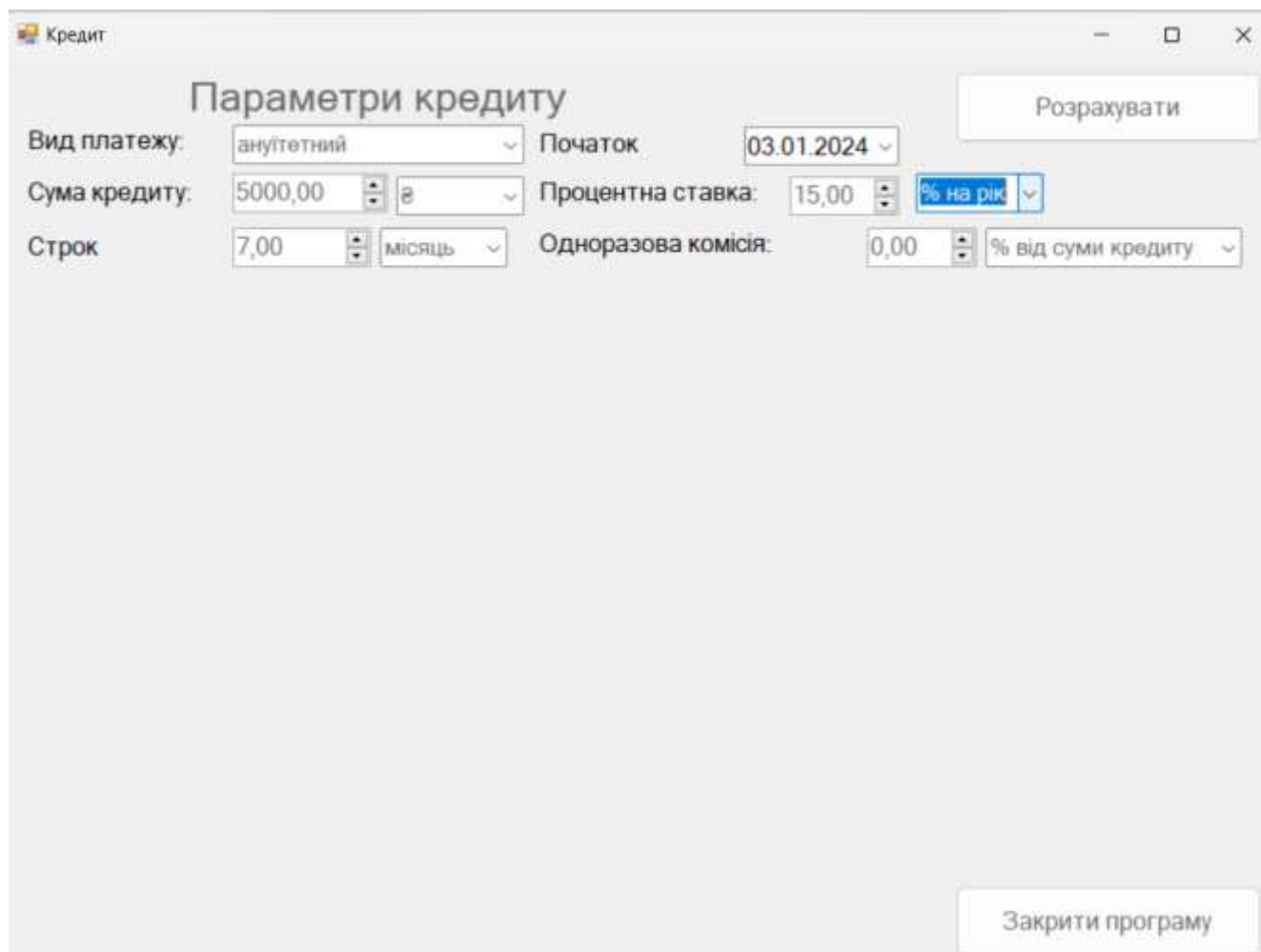
Закрити програму

Пн	Вт	Ср	Чт	Пт	Сб	Нд
25	26	27	28	29	30	31
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	1	2	3	4

Сьогодні: 21.01.2024

Рисунок 3.21 – Вибір дати початку сплати кредитної заборгованості

Далі необхідно ввести процентну ставку за кредитом, яка вираховується в розрахунку % на рік (рисунок 3.22)



The screenshot shows a software window titled "Кредит" with a form titled "Параметри кредиту". The form contains the following fields and controls:

- Вид платежу:** dropdown menu with "ануїтетний" selected.
- Початок:** date field with "03.01.2024" selected.
- Сума кредиту:** numeric field with "5000,00" and a currency dropdown with "₴" selected.
- Процентна ставка:** numeric field with "15,00" and a dropdown menu with "% на рік" selected.
- Строк:** numeric field with "7,00" and a dropdown menu with "місяць" selected.
- Одноразова комісія:** numeric field with "0,00" and a dropdown menu with "% від суми кредиту" selected.

Buttons: "Розрахувати" (top right) and "Закрити програму" (bottom right).

Рисунок 3.22 – Введення % ставки

Процентна ставка за кредитом, яка вираховується в розрахунку % на рік, називається річною процентною ставкою. Вона вказує, скільки відсотків від суми кредиту кредитор отримує за користування цим кредитом протягом одного року.

Річна процентна ставка є важливою характеристикою кредиту, яку необхідно враховувати при виборі кредитної пропозиції. Чим вища річна процентна ставка, тим дорожчим буде кредит для позичальника.

Програмою також передбачено введення одноразової комісії або вибір варіанту, що комісія не враховується (рисунок 3.23).

Рисунок 3.23 – Одноразова комісія

Одноразова комісія при кредитуванні – це додаткова плата, яку позичальник сплачує кредитору при отриманні кредиту. Вона може бути фіксованою або вираженою в процентах від суми кредиту або вартості застави.

Одноразова комісія може включати в себе такі витрати:

Плату за оформлення кредиту. Ця комісія покриває витрати кредитора на підготовку документів, проведення юридичної перевірки позичальника та інші витрати, пов'язані з видачею кредиту.

Плату за страхування. Ця комісія покриває витрати кредитора на страхування життя позичальника, майна, яке виступає заставою, або інших ризиків, пов'язаних з кредитуванням.

Інші витрати. Це можуть бути різні комісії, пов'язані з кредитуванням, наприклад, плата за дострокове погашення кредиту, плата за перекредитування або плата за обслуговування кредитного договору.

Одноразова комісія є важливою характеристикою кредиту, яку необхідно враховувати при виборі кредитної пропозиції. Чим вища одноразова комісія, тим дорожчим буде кредит для позичальника.

Після заданих параметрів потрібно натиснути кнопку Розрахувати, з'явиться графік погашення кредиту у вигляді таблиці за введеними параметрами (рисунок 3.24).

Кредит

Вид платежу: **ануїтетний** Змінити дані

Загальна сума платежу: 750,44 ₴

Переплата за процентами за кредит: 9,26 ₴

Підсумкова переплата з врахуванням комісі 61,8 ₴

	№	Дата платежу	Сума платежу	Сума погашення основного боргу	Сума нарахування процентів	Залишок боргу
▶	1	3.1.2024	750,44 ₴	687,94 ₴	62,5 ₴	4312,06 ₴
	2	3.2.2024	750,44 ₴	696,54 ₴	53,9 ₴	3615,51 ₴
	3	3.3.2024	750,44 ₴	705,25 ₴	45,19 ₴	2910,26 ₴
	4	3.4.2024	750,44 ₴	714,07 ₴	36,38 ₴	2196,2 ₴
	5	3.5.2024	750,44 ₴	722,99 ₴	27,45 ₴	1473,21 ₴
	6	3.6.2024	750,44 ₴	732,03 ₴	18,42 ₴	741,18 ₴
•	7	3.7.2024	750,44 ₴	741,18 ₴	9,26 ₴	0 ₴

Зберегти таблицю Закрити програму

Рисунок 3.24 – Результат натискання кнопки Розрахувати

Для проведення аналізу розробкою передбачено кнопка Змінити дані. Для порівняння змінено лише вид платежу (рисунок 3.25). Користувач може зберегти

дві таблиці з різними видами платежу і провести їх аналіз для найкращої умови кредитування.

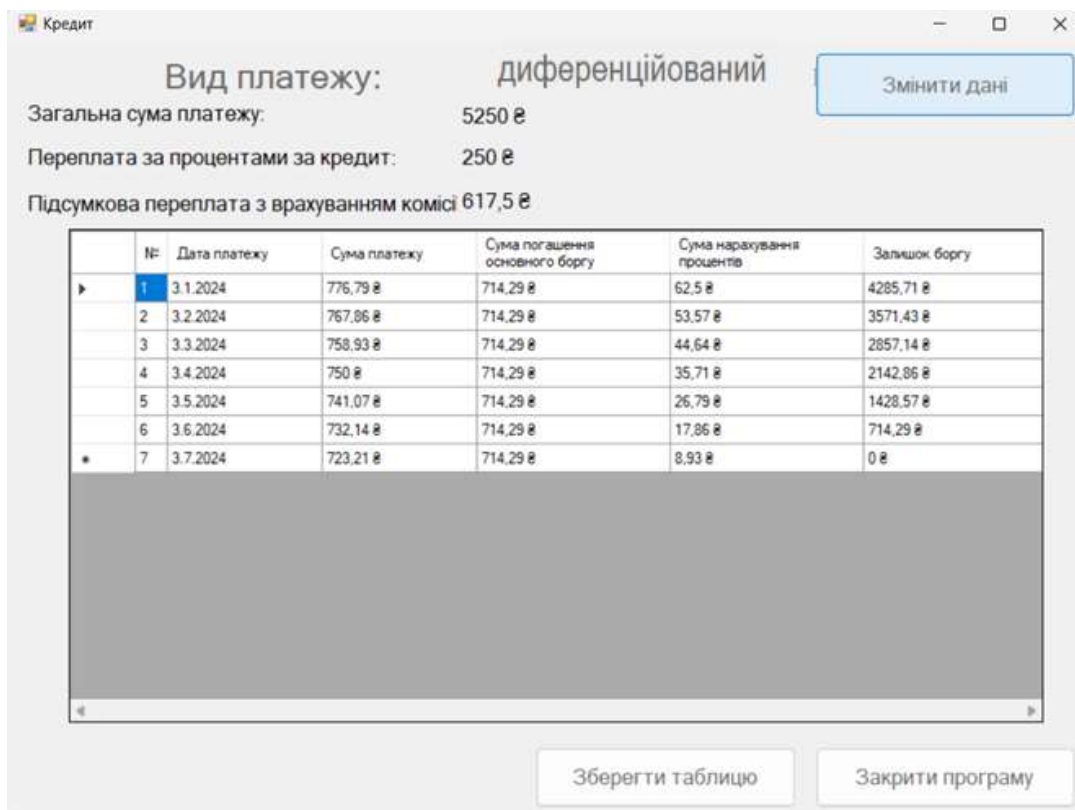


Рисунок 3.25 – Змінити дані

Для збереження і перегляду графіка погашення кредиту програмою передбачено кнопка Зберегти таблицю (рисунок 3.26). Додаток Г.

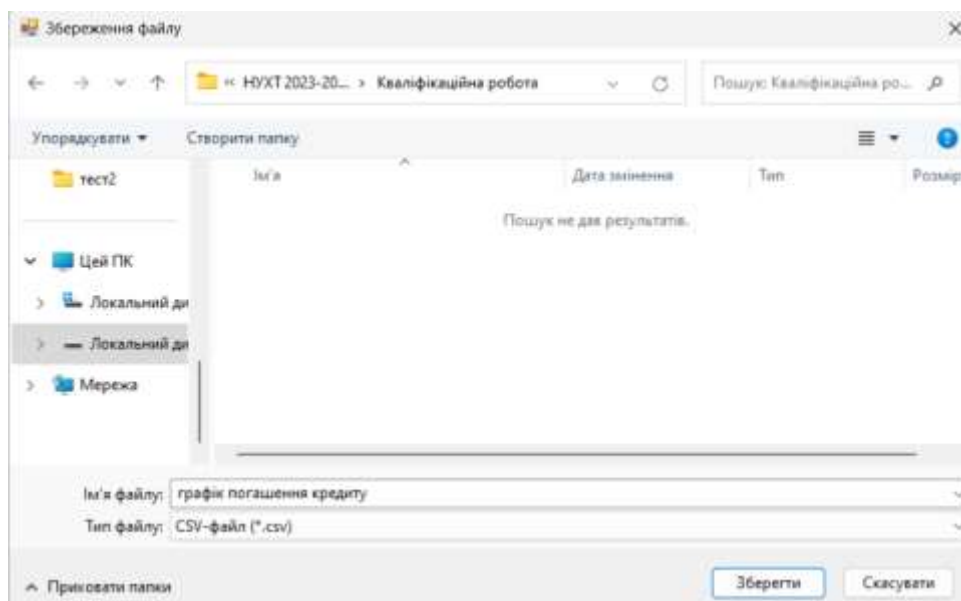


Рисунок 3.26 – Збереження таблиці

Після натискання кнопки Зберегти – з’являється вікно з повідомленням (рисунок 3.27).

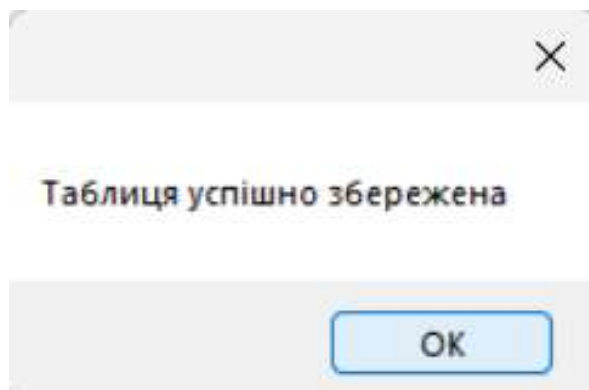


Рисунок 3.27 – Вікно з повідомленням про збереження

Якщо користувач ввів не вірні дані, з’являється вікно з повідомленням про виникнення помилки (рисунок 3.28).

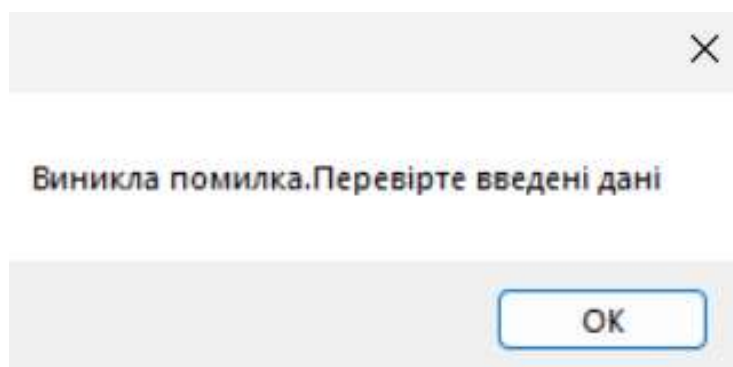


Рисунок 3.28 – Виникла помилка

При закритті програми з’являється вікно з повідомленням (рисунок 3.29)

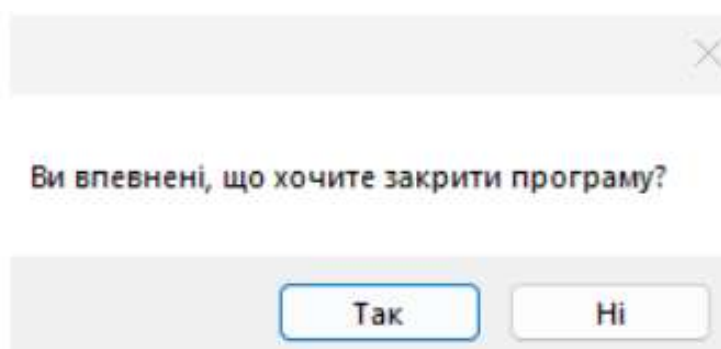


Рисунок 3.29 – Повідомлення про закриття програми

3.5 Технічне та системне забезпечення системи

3.5.1 Обґрунтування вибору технічних засобів

Вибір технічних засобів для розробки програмного забезпечення для обчислення, аналізу платежів та графіка погашення кредитів включав декілька ключових факторів і залежав від їхніх можливостей, сумісності з іншими компонентами та специфіки завдання. Детально розглянемо кожен засіб.

1. Draw.io. Цей інструмент був використаний для створення концептуальної діаграми програмного забезпечення. Draw.io – це онлайн-інструмент для створення різноманітних діаграм, включаючи концептуальні діаграми програмного забезпечення. Він надає можливість створювати діаграми різних типів, такі як UML-діаграми, діаграми потоків, ER-діаграми.

2. UML (Unified Modeling Language). Цей засіб використаний для створення діаграми класів у Visual Studio 2022 для проекту C#. За допомогою інструмента Class Designer у Visual Studio 2022 можна проєктувати, візуалізувати та рефакторювати діаграми класів у проєктах C#. Діаграми класів показують елементи коду та їхні зв'язки у візуальній поверхні дизайну, що допомагає краще зрозуміти структуру проєкту та реорганізувати код Class Designer був обраний через можливості, які він надає для моделювання класів та інтеграцією з іншими інструментами.

4. Visual Studio 2022 та мова C#: Visual Studio 2022 є інтегроване середовище розробки (IDE), яке надає широкий набір інструментів для розробки програмного забезпечення. Visual Studio 2022 має зручний інтерфейс та підтримує мову програмування C#. C# базується на об'єктно-орієнтованому підході до програмування, що дозволяє створювати модульні, повторно використовувані та легко зрозумілі програми. Крім того, використання мови C# дозволяє вам отримати доступ до всіх можливостей та ресурсів платформи .NET.

5. Платформа .NET Framework: Забезпечує середовище для виконання програм на мові C#, а також багато корисних бібліотек і функцій для швидкого розгортання ІС.

Комбінація цих засобів дозволила ефективно розробляти програмне забезпечення для обчислення, аналізу платежів та графіка погашення кредитів, враховуючи бізнес-процеси, структуру та програмну реалізацію.

3.5.2 Обґрунтування вибору операційної системи та протоколу обміну даними

Windows є однією з найпопулярніших операційних систем у світі, що означає, що багато користувачів вже працюють з цією платформою. Це робить Windows зручним вибором для розробки програмного забезпечення з урахуванням широкого кола користувачів.

Розгалужена інфраструктура та інтеграційні можливості.

Windows має розгалужену інфраструктуру, що дозволяє легко інтегрувати програмне забезпечення з іншими додатками та службами, такими як бази даних, хмарні сервіси і інше.

Розробка на .NET платформі.

.NET є потужним фреймворком для розробки програмного забезпечення на платформі Windows. Використання .NET спрощує створення, розгортання та управління програмами.

Підтримка багатьох мов програмування.

Windows підтримує багато мов програмування, включаючи C#. Це дає можливість вибору мови, яка найкраще відповідає потребам розробки.

Обґрунтування вибору протоколу обміну даними

HTTPS (Hypertext Transfer Protocol Secure)

Для забезпечення безпеки обміну даними між клієнтом та сервером рекомендується використовувати протокол HTTPS, який забезпечує шифрування даних та захист від несанкціонованого доступу.

Обрана операційна система та протокол обміну даними для розробки програмного забезпечення для обчислення, аналізу платежів та графіка погашення кредитів враховує специфіку проєкту та вимоги користувачів, а також забезпечує надійність, ефективність та безпеку системи.

РОЗДІЛ 4. ОХОРОНА ПРАЦІ

4.1 Вимоги безпеки під час виконання робіт за ПК

В межах даного розділу роботи розглянемо вимоги безпеки під час виконання робіт за ПК, так як розробка програмного забезпечення для обчислення, аналізу платежів та графіка погашення кредитів відбувалося безпосередньо із застосуванням ПК.

Робота за комп'ютером передбачає ряд шкідливих факторів та загроз. Що пов'язано з можливістю отримання травм та професійних захворювань, то закон України «Про охорону праці» визначає основні положення відносно реалізації конституційного права громадян на охорону їх життя та здоров'я в процесі трудової діяльності, регулює за допомогою відповідних органів державної влади відносини між роботодавцем і працівником з питань безпеки, гігієни працівника виробничого середовища та встановлює єдиний порядок організації охорони праці в Україні.

Охорона праці - це система правових, соціально-економічних, організаційно-технічних, санітарно-гігієнічних і лікувально-профілактичних заходів та засобів, спрямованих на збереження життя, здоров'я і працездатності людини у процесі трудової діяльності.

Дана робота розроблялась на робочому місці, яке знаходиться на третьому поверсі триповерхової адміністративної будівлі.

Площа приміщення складає 50 м^2 , що відповідає санітарним нормам, згідно з якими норма на одного працюючого повинна бути не менш 6 м^2 . Висота приміщення $3,0 \text{ м}$. таким чином обсяг приміщення складає 150 м^3 , по нормам – не менш 20 м^3 .

Перелік шкідливих та небезпечних факторів, які діють при роботі на ПЕОМ наведений в таблиці 4.1 згідно з та ДсанПіН 3.3.2.007-98 «Державні санітарні

правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» [8] та ДСН 33.6.042-99 [9].

Таблиця 4.1 – Шкідливі та небезпечні фактори виробничого середовища

Найменування факторів	Джерела їх виникнення	Параметр, що нормується, та нормативне значення
1	2	3
1. Рівень електромагнітного випромінювання	ЕПТ монітора, системний блок, мережа живлення	Відстань - 50 см, навколо ПК 2-5 кГц – 25 В/м
2. Ультрафіолетове випромінювання	Комп'ютер	Щільність потоку ультрафіолетового випромінювання 10 Вт/м
3. Емоційні перенавантаження, напруга зорового аналізатору	Складність виконання завдань	Зниження реакції користування на звук і світло на 40-50 %
4. Підвищений рівень шуму	Вентилятор, система освітлення, друкувальні прилади	Рівень звуку LA=50 дБ (А)
5. Підвищене значення напруги в електричній мережі	Блок живлення	I=0,6 мА; U=36 В
6. Недолік природнього освітлення	Неправильне планування розташування комп'ютера	КПО, №, Е, лк
7. Вібрація	Вентиляційна система	Віброприскорення, м/с ² , віброшвидкість, м/с або їх рівні LA, LV, дБ
8. Виробничий пил	Статична електрика, накопичена на поверхні комп'ютера	ГДК=4 мг/м ³
9. Несприятливі температури мікроклімату	Не задовільна робота опалення або вентиляції	Температура (t, °С), вологість (φ, %), швидкість руху повітря (V, м/с).

Оптимальні параметри мікроклімату, відображені в таблиці 4.2, встановлюються залежно від категорії робіт по фізичному навантаженню.

Оптимальні мікрокліматичні умови – це такі параметри мікроклімату, які за тривалого і систематичного впливу на людину забезпечують збереження

нормального теплового стану організму без активації терморегуляції, тобто забезпечують стан теплового комфорту, створюючи таким чином умови для високого рівня працездатності.

Забезпечення необхідних параметрів мікроклімату досягається у холодний період кондиціонуванням та системою опалення, а в теплий період лише системою кондиціонування, згідно з ДБН В.2.5.- 67: 2013.

Таблиця 4.2 – Оптимальні параметри мікроклімату

Категорія робіт по вазі	Період року	Температура, °С	Відносна вологість, %	Швидкість руху повітря, м/с
Легка – Іа	теплий	23-25	40-60	0,1-0,2
	холодний	22-24	40-60	0,1

Освітлення виробничих, службових і допоміжних приміщень регламентується ДБН В.2.5.-28-2006.

Штучне та природне освітлення.

В даному випадку використовується комбіноване освітлення: природне бокове вранці та штучне ввечері.

Виконувана робота відноситься до ІІІ розряду зорової праці. Мінімальний розмір об'єктів від 0,3 до 0,5 мм, фон світлий, контраст великий, підрозряд зорових робіт – «Г» в приміщенні забезпеченому комбінованим освітленням: у світлий час доби – бокове однобічне природне освітлення – три віконних прорізи, у темний час загальне чи місцеве рівномірне штучне.

В таблиці 4.3 наведені норми освітлення для даного розряду і точності зорових робіт.

Таблиця 4.3 – Характеристики виробничого освітлення

Точність зорових робіт	Мінімальний розмір об'єкта розрізнення, мм	Розряд зорових робіт	Характеристика типу фону	Контраст об'єкта розрізнення з фоном	Розряд зорових робіт	Нормативне значення параметрів освітлення	
						Природне, %	Штучне, лк
Високої точності	0,3-0,5	ІІІ	Світлий	Великий	Г	1,2	400

Джерелом шуму в кабінеті може слугувати така техніка: телефон, принтер, факс, комп'ютер, кондиціонер. Рівень шуму не повинен перевищувати 50 дБ. Рівень вібрації в умовах комфортної роботи, не повинен перевищувати 75 дБ.

Для зменшення рівня звуку та вібрації застосовуються демпфуючі матеріали.

Основними методами захисту є: зниження шуму та вібрації в джерелі (підставки, шумопоглинальні корпуси) і на шляху розповсюдження (ширми, шумопоглинальні стійки), застосування індивідуальних засобів та організаційно-профілактичних методів захисту.

Для захисту від електромагнітного випромінювання застосовується спеціальне покриття екрану. Напряга електромагнітних полів у діапазоні 1 – 12 кГц, 60 – 300кГц по магнітній і електричній складовій повинні відповідати вимогам до ДСанПіН 3.3.2-007-98 [14].

Повітря зовнішнього середовища містить параметри представлені в таблиці 4.4) [9].

Таблиця 4.4 – Рівень іонізації повітря при роботі на ПК

Рівні	Кількість іонів в 1 см ³ повітря	
	Позитивні	Негативні
Мінімально необхідні	400	600
Оптимальні	1500-3000	3000-5000
Максимально припустимі	50000	50000

Напряга електромагнітних полів у діапазоні 1 – 12 кГц, 60 – 300 кГц по магнітній і електричній складовій повинні відповідати вимогам до ДСанПіН 3.3.2-007-98 [9].

При проектуванні систем електропостачання, монтаж силового електроустаткування й електричного освітлення в будинках і приміщеннях для ПК необхідно дотримуватися вимог нормативно-технічної документації.

Комплекс необхідних заходів щодо техніки безпеки визначається, виходячи з видів електроустановки, її номінальної напруги, умов середовища, типу приміщення й доступності електроустаткування.

Документом «Правил улаштування електроустановок» ПУЕ 2017 передбачені наступні міри електробезпеки:

1) конструктивні заходи: персональна ПК відноситься як електроустановка до 1000 В закритого виконання, усі рубильники встановлені в закритих корпусах, усі струмоведучі частини розміщені в захисних коробах або покриті шаром ізоляції, який виключає можливість дотику до них. Комп'ютер має робочу ізоляцію і елементи заземлення;

2) експлуатаційні міри: при роботі на ЕОМ необхідно дотримувати правила техніки безпеки при роботі з високою напругою, не підключати і не відключати кабелі при включеній напрузі мережі, технічне обслуговування і ремонт проводити тільки при вимкненому живленні;

3) схемно – конструктивні міри: в електричних мережах із глухо заземленим нейтральним провідником як схемно – конструктивну міру безпеки застосовують занулення – навмисне з'єднання металевих неструмоведучих частин комп'ютера з нейтральним провідником.

Відповідно до вимог ДБН В 1.1.7-2016 пожежна безпека забезпечується наступними заходами, які застосовуються до категорії В: системою пожежного захисту; організаційними заходами щодо пожежної безпеки; системою запобігання пожеж: запобігання утворенню горючого середовища, та запобігання утворення у горючому середовищі джерел запалювання.

Для зменшення небезпеки утворення в сталюму середовищі джерел запалювання передбачено:

1) використання устаткування, що відповідає класу пожежобезпечної зони ПШа: ступінь захисту електроапаратури повинна бути не менш IP-44, ступінь захисту світильників IP-23, відповідно до ДБН В 1.1.7-2016;

2) блискавковідвід будинків, споруджень і устаткування для даного класу пожежонебезпеки, зони П-Па і місцевості із середньою грозовою діяльністю 20 і більше грозових годин у рік, тобто встановлена III категорія блискавко захисту відповідно до ДСТУ EN 62305-1:2012;

3) застосування заземлення захисного екрану для стоку статичної електроенергії; використання для гасіння пожежі вуглекислого вогнегасника ВВ-2.

Організаційними заходами протипожежної профілактики є: навчання виробничого персоналу протипожежним правилам; видання необхідних інструкцій, плакатів, засобів наочної агітації, плану евакуації персоналу у випадку пожежі.

4.2 Дії персоналу в надзвичайних ситуаціях

Розглянемо дії персоналу у різних надзвичайних ситуаціях. Дії у разі вибуху.

Вибух. Основні вражаючі фактори вибуху: повітряна ударна хвиля та уламкові поля, що утворюються уламками зруйнованих об'єктів, що летять, технологічного обладнання, вибухових пристроїв.

При загрозі вибуху слід лягти на живіт, захищаючи голову руками, подалі від вікон, закслених дверей, проходів, сходів.

Якщо стався вибух, вжити заходів щодо недопущення пожежі та паніки; надати першу допомогу постраждалим. Кожен працівник при виявленні вогнища або ознак горіння (задимлення, запах гару, підвищення температури тощо) повинен негайно повідомити про це за телефоном «101».

При цьому назвати найменування об'єкта, місце вибуху, пожежі, а також своє прізвище; вжити заходів щодо евакуації людей, гасіння пожежі та збереження матеріальних цінностей.

Вимоги щодо використання первинних засобів пожежогасіння: Вуглекислотні вогнегасники (ОУ-2, ОУ-3, ОУ-5, ОУ-6, ОУ-7 тощо) призначені для гасіння загорянь різних горючих речовин, виключення, коли горіння яких відбувається без доступу повітря.

Для приведення в дію вуглекислотних вогнегасників необхідно розтруб направити на палаючий предмет, зірвати пломбу, висмикнути чеку, натиснути на важіль (або повернути маховик вентиля вліво вцент), направити струмінь на полум'я. Тримати вогнегасник вертикально, перевертати його не потрібно.

Щоб уникнути обморожування, не торкатися металевої частини розтрубу оголеними частинами тіла. При гасінні електроустановок, що знаходяться під напругою, не допускається підводити до них розтруб ближче 1м.

Внутрішні пожежні крани призначені для подачі води.

Дії у разі хімічної аварії.

Небезпека хімічної аварії для людей полягає в порушенні нормальної життєдіяльності організму та можливості віддалених генетичних наслідків, а за певних обставин – у летальному результаті при потраплянні АХНР в організм через органи дихання, шкіру, слизові оболонки, рани та разом з їжею.

При отриманні сигналу про хімічну аварію включити радіоприймач для отримання достовірної інформації про аварію та рекомендовані дії.

Закрити вікна, вимкнути електропобутові прилади. Для захисту органів дихання використовувати ватно-марлеву пов'язку або підручні вироби з тканини, змочені у воді, 2-5%-ному розчині харчової соди (для захисту від хлору), 2%-ному розчині лимонної або оцтової кислоти (для захисту від аміаку).

При неможливості залишити зону зараження щільно закрити двері, вікна, вентиляційні отвори та димарі; щілини в них заклеїти папером чи скотчем.

Не ховатися на перших поверхах будівель, у підвалах та напівпідвалах.

При підозрі на поразку АХНР виключити будь-які фізичні навантаження, прийняти рясне питво (молоко, чай) та негайно звернутися до лікаря.

Утримуватися від вживання водопровідної води – до офіційного висновку щодо її безпеки. На зараженій місцевості рухатися швидко, але не бігти, піднімаючи пил, не торкатися навколишніх предметів, не наступати пролиту рідину або порошкоподібні розсипи невідомих речовин.

Дії у разі обвалення будівель, споруд. Повне або часткове раптове обвалення будівлі – це надзвичайна ситуація природного або техногенного характеру, яка також виникає за причини помилок, допущених на етапі проектування.

Руйнування комунально-енергетичних мереж, утворення завалів, травмуванню та загибелі людей. Почувши вибух або виявивши, що будівля втрачає свою стійкість, негайно покинути його.

Залишаючи приміщення, спускатися сходами, а не на ліфті: він у будь який момент може зупинитись.

Не панікувати, не влаштовувати тисняву у дверях під час евакуації. Зупиняти тих, хто збирається стрибати з балконів (поверхів вище першого) та через засклені вікна.

Якщо відсутня можливість покинути будівлю, зайняти безпечне місце: отвори капітальних внутрішніх стін, кути, утворені капітальними внутрішніми стінами, під балконами каркасу (вони захищають від падаючих предметів та уламків). Відкрити двері з приміщення, щоб забезпечити вихід.

Не піддаватися паніці та зберігати спокій. Триматися подалі від вікон, електроприладів.

Якщо виникла пожежа, негайно спробувати загасити її. Телефон використовувати лише для виклику представників правоохоронних органів, пожежної охорони, лікарів, рятувальників.

Не користуватися сірниками: існує небезпека вибуху внаслідок витоку газу. Опинившись надворі, не стояти поблизу будинку. Перейти на відкритий простір.

Дії у разі знаходження під завалом.

Дихати глибоко, не піддаватися паніці.

По можливості надати собі першу допомогу. Пристосуватися до обстановки та озирнутися, пошукати вихід. Спробувати визначити, де знаходиться людина, чи немає інших людей: прислухатись, подати голос.

Слід пам'ятати: людина здатна витримати спрагу і голод протягом тривалого часу, якщо не марно витрачати енергію.

Пошукати в кишенях або поблизу предмети, щоб подати світлові або звукові сигнали: ліхтарик або металеві предмети, якими можна постукати по трубі чи стіні (привернути увагу рятувальників).

Якщо єдиним виходом є вузький лаз - протиснутися через нього. Для цього розслабити м'язи та рухатися, притиснувши лікті до тіла.

У всіх випадках ураження електричним струмом виклик лікаря є обов'язковим незалежно від стану потерпілого.

Якщо потерпілий знаходиться при свідомості, його треба покласти у зручне положення і до прибуття лікаря забезпечити спокій, обов'язково спостерігаючи за диханням і пульсом.

Не можна дозволяти потерпілому рухатись, продовжувати роботу. Якщо лікаря швидко викликати не можна, необхідно терміново доставити потерпілого у медичний пункт.

Якщо потерпілий знаходиться у непритомному стані, його необхідно покласти, розстебнути одяг, забезпечити приплив свіжого повітря, дати понюхати нашатирний спирт, бризнути на нього водою і забезпечити спокій. У той же час потрібно викликати лікаря. Якщо потерпілий дихає погано, рідко і судомно, йому необхідно робити штучне дихання і непрямий масаж серця.

У разі відсутності в потерпілого ознак життя не можна вважати його померлим. Якщо в такому стані потерпілому не буде надано негайну першу допомогу у вигляді штучного дихання і зовнішнього масажу серця, то настане смерть.

Оживлення організму, ураженого електричним струмом, може бути проведено кількома способами. Всі вони базуються на штучному диханні.

Починати штучне дихання слід негайно після вивільнення потерпілого від електричного струму і проводити безперервно до досягнення позитивного результату.

Штучне дихання необхідно робити безперервно, до прибуття лікаря.

Переносити потерпілого до іншого місця треба тільки в тих випадках, коли йому, чи особі, яка надає допомогу, продовжує загрожувати небезпека.

Ураженого електричним струмом можна визнати померлим тільки за наявності видимих тяжких зовнішніх ушкоджень: роздроблення черепа у разі падіння чи обпалення всього тіла.

В інших випадках констатує смерть лише в лікарні.

Також розглянемо дії при пожежі. Про виникнення пожежі в приміщеннях негайно повідомити пожежну охорону.

При цьому необхідно назвати адресу, зазначити кількість поверхів будівлі, місце виникнення пожежі, обстановку на пожежі, наявність людей, а також повідомити своє прізвище.

Вжити (по можливості) заходи на евакуацію людей, гасіння (локалізацію) пожежі з використанням первинних засобів пожежогасіння та на збереження матеріальних цінностей.

Повідомити про виникнення пожежі керівника (заступників керівника) чи відповідальну компетентну посадову особу та чергового охорони.

У разі необхідності, викликати інші аварійно-рятувальні служби (медичну, газорятувальну тощо).

ВИСНОВКИ

Ця кваліфікаційна робота становить результат інтенсивної розробки та аналізу програмного забезпечення, яке спрямоване на ефективне управління кредитними платежами та аналіз платежів з використанням сучасних технологій. Розробка здійснена в середовищі Visual Studio 2022, використовуючи мову програмування C# та засоби .NET Framework.

Однією з ключових переваг розробленого програмного продукту є його інтеграція з відомим та потужним інтегрованим середовищем розробки Visual Studio 2022. Використання цього інструментарію дозволило не лише ефективно створювати програмний код, а й забезпечити високу якість та надійність розробленого продукту. Редагування, налагодження та тестування коду стали більш доступними та зручними завдяки широкому набору функцій, які надає Visual Studio.

Основною функціональністю розробленого програмного продукту є обчислення, аналіз та візуалізація платежів та графіка погашення кредитів. Використання мови програмування C# сприяло швидкому розгортанню логіки обчислень та взаємодії з даними. Також, використання багатofункціональних бібліотек .NET дозволило ефективно реалізувати операції обчислення кредитного платежу та зберігання даних для кращого візуального розуміння.

Процес розробки включав в себе тісну співпрацю з елементами інтерфейсу користувача, щоб забезпечити інтуїтивне використання та привабливий вигляд програмного забезпечення. Використання Visual Studio Designer та можливостей C# для роботи з Windows Forms або WPF дозволило створити дружлюбний та зручний інтерфейс для кінцевого користувача.

У процесі аналізу платежів та графіка погашення кредитів було приділено особливу увагу оптимізації використання ресурсів та врахуванню можливостей швидкісного та надійного виконання. Використання сучасних методологій

розробки та підходів до оптимізації коду сприяло підвищенню продуктивності та реактивності програмного забезпечення.

Узагальнюючи, дана кваліфікаційна робота є вагомим внеском у сферу розробки фінансових програм та відзначається високою якістю при реалізації функціоналу, оптимізації та використання сучасних інструментів. Розроблене програмне забезпечення володіє потужними можливостями для роботи з розрахунку кредитних платежів та може використовуватися як ефективний інструмент для управління та аналізу погашення кредитів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Credit Karma [Електронний ресурс] – Режим доступу до ресурсу:
<https://www.creditkarma.com/>
2. Loan Calculator [Електронний ресурс] – Режим доступу до ресурсу:
<https://www.nerdwallet.com/calculator/loan-calculator>
3. Mint [Електронний ресурс] – Режим доступу до ресурсу:
<https://mint.intuit.com/>
4. Quicken [Електронний ресурс] – Режим доступу до ресурсу:
<https://www.quicken.com/>
5. You Need a Budget [Електронний ресурс] – Режим доступу до ресурсу:
<https://www.ynab.com/>
6. ДСТУ 3008-2015. Документація. Звіти у сфері науки і техніки. Структура та правила оформлювання.
7. ДСТУ Б А.2.4-4:2009 «Основні вимоги до проектної та робочої документації.
8. ДСанПН 3.3.2.007-98. Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин. Затверджено Постановою Головного державного санітарного лікаря України 10 грудня 1998 р. №7.
9. ДСН 3.3.6.037-99. Державні санітарні норми. Шум. Загальні вимоги безпеки. – К.: Затверджено Постановою Головного державного санітарного лікаря України 1 грудня 1999 р. №37.
10. ДСТУ ISO/IEC 12207:2014. Інженерія систем і програмного забезпечення. Процеси життєвого циклу програмного забезпечення.
11. ДСТУ ISO/IEC 15910:2012. Інформаційні технології. Документування програм. Документація користувача
12. ДСТУ ISO/IEC 29155-1:2015. Розроблення систем і програмного забезпечення. Платформи для тестування проектів з розроблення інформаційних

систем. Частина 1. Концепції та визначення.

13. Закон України «Про банки і банківську діяльність». Редакція від 17.10.2019. URL : <http://zakon3.rada.gov.ua> (дата звернення: 04.04.2020)

14. Коноваленко І.В. Платформа .NET та мова програмування C#: навчальний посібник / І. В. Коноваленко, П. О. Марущак. – Тернопіль: ФОП Паляниця В. А., 2020. – 320 с. [Електронний ресурс] – Режим доступу до ресурсу: <http://elartu.tntu.edu.ua/handle/lib/32825>

15. Кредитні операції банків [Електронний ресурс] – Режим доступу до ресурсу: http://bses.in.ua/journals/2020/52_2_2020/21.pdf

16. Методичні рекомендації до виконання випускної кваліфікаційної роботи на здобуття освітнього ступеня «Бакалавр» спеціальності 122 «Комп'ютерні науки», освітньо-професійних програми «Комп'ютерні науки» ден. та заоч. форми навчання [Електрон. ресурс] / уклад. О. М. М'якшило, М. П. Костіков. – К.: НУХТ, 2022. – 34 с.

17. Розвиток кредитного ринку в Україні [Електронний ресурс] – Режим доступу до ресурсу: [40.pdf \(market-infr.od.ua\)](http://market-infr.od.ua)

18. Як погасити кредит [Електронний ресурс] – Режим доступу до ресурсу: <https://informer.ua/uk/yak-pogasiti-kredit-tri-shemi-rozrahunku-platezhiv>

Додаток Б. Код Form

```
// Form1
//
this.AutoScaleDimensions = new System.Drawing.SizeF(8F, 16F);
this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
this.BackColor = System.Drawing.SystemColors.Control;
this.ClientSize = new System.Drawing.Size(1121, 745);
this.Controls.Add(this.button4);
this.Controls.Add(this.button3);
this.Controls.Add(this.button2);
this.Controls.Add(this.label14);
this.Controls.Add(this.dateTimePicker1);
this.Controls.Add(this.dataGridView1);
this.Controls.Add(this.numericUpDown4);
this.Controls.Add(this.numericUpDown3);
this.Controls.Add(this.numericUpDown2);
this.Controls.Add(this.numericUpDown1);
this.Controls.Add(this.button1);
this.Controls.Add(this.comboBox5);
this.Controls.Add(this.comboBox4);
this.Controls.Add(this.comboBox3);
this.Controls.Add(this.comboBox2);
this.Controls.Add(this.comboBox1);
this.Controls.Add(this.label7);
this.Controls.Add(this.label6);
this.Controls.Add(this.label5);
this.Controls.Add(this.label4);
this.Controls.Add(this.label3);
this.Controls.Add(this.label2);
this.Controls.Add(this.label1);
this.Margin = new System.Windows.Forms.Padding(4);
this.Name = "Form1";
this.Text = "Кредит";
this.Load += new System.EventHandler(this.Form1_Load);

((System.ComponentModel.ISupportInitialize)(this.numericUpDown1)).EndInit();

((System.ComponentModel.ISupportInitialize)(this.numericUpDown2)).EndInit();

((System.ComponentModel.ISupportInitialize)(this.numericUpDown3)).EndInit();

((System.ComponentModel.ISupportInitialize)(this.numericUpDown4)).EndInit();
    ((System.ComponentModel.ISupportInitialize)(this.dataGridView1)).EndInit();
    this.ResumeLayout(false);
    this.PerformLayout();

}

#endregion
private System.Windows.Forms.Label label1;
private System.Windows.Forms.Label label2;
private System.Windows.Forms.Label label3;
private System.Windows.Forms.Label label4;
private System.Windows.Forms.Label label5;
private System.Windows.Forms.Label label6;
private System.Windows.Forms.Label label7;
private System.Windows.Forms.ComboBox comboBox1;
private System.Windows.Forms.ComboBox comboBox2;
private System.Windows.Forms.ComboBox comboBox3;
private System.Windows.Forms.ComboBox comboBox4;
private System.Windows.Forms.ComboBox comboBox5;
```

```
private System.Windows.Forms.Button button1;
private System.Windows.Forms.NumericUpDown numericUpDown1;
private System.Windows.Forms.NumericUpDown numericUpDown2;
private System.Windows.Forms.NumericUpDown numericUpDown3;
private System.Windows.Forms.NumericUpDown numericUpDown4;
private System.Windows.Forms.DataGridView dataGridView1;
private System.Windows.Forms.SaveFileDialog saveFileDialog1;
private System.Windows.Forms.DateTimePicker dateTimePicker1;
private System.Windows.Forms.Label label14;
private System.Windows.Forms.Button button2;
private System.Windows.Forms.DataGridViewTextBoxColumn Column1;
private System.Windows.Forms.DataGridViewTextBoxColumn Column2;
private System.Windows.Forms.DataGridViewTextBoxColumn Column3;
private System.Windows.Forms.DataGridViewTextBoxColumn Column4;
private System.Windows.Forms.DataGridViewTextBoxColumn Column5;
private System.Windows.Forms.DataGridViewTextBoxColumn Column6;
private System.Windows.Forms.Button button3;
private System.Windows.Forms.Button button4;
}
}
```

Додаток В. Код для обчислення, аналізу платежів та погашення кредитів

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

namespace WindowsFormsApp1
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void textBox2_TextChanged(object sender, EventArgs e)
        {
        }

        private void label5_Click(object sender, EventArgs e)
        {
        }

        private void label7_Click(object sender, EventArgs e)
        {
        }

        private void label6_Click(object sender, EventArgs e)
        {
        }

        private void comboBox5_SelectedIndexChanged(object sender, EventArgs e)
        {
        }

        public void monthCalendar1_DateChanged(object sender, DateRangeEventArgs e)
        {
            DateTime data = e.Start;
        }

        private void button1_Click(object sender, EventArgs e)
        {
            try
            {

                string val = comboBox1.Text;
                string srk = comboBox4.Text;
                string stav = comboBox3.Text;
                string komis = comboBox2.Text;
            }
        }
    }
}
```

```

string vid = comboBox5.Text;
string data = dateTimePicker1.Text;

double srok = 0;
double sum = Convert.ToDouble(numericUpDown1.Text);
double komiss = 0;

if (srk == "місяць")
{
    srok = Convert.ToDouble(numericUpDown4.Text);
}
else if (srk == "рік")
{
    srok = Convert.ToDouble(numericUpDown4.Text) * 12;
}

if (komis == "не враховується")
{
    komiss = 0;
}
else { komiss = (Convert.ToDouble(numericUpDown2.Text) / 100); }

double stavka = Convert.ToDouble(numericUpDown3.Text);

double pay = 0;
double paymonth = 0;
double fullpay = 0;
double pr = (stavka / (12 * 100));
double ostatek = sum;
double ss = 0;
double fullpaysum = 0;
double obsum = 0;
double obproz = 0;
double itog = 0;

int day = dateTimePicker1.Value.Day;
int month = dateTimePicker1.Value.Month;
int year = dateTimePicker1.Value.Year;

dataGridView1.RowCount = Convert.ToInt32(srok);

for (int i = 0; i < srok; i++)
{
    dataGridView1.Rows[i].Cells[0].Value = i + 1;

    dataGridView1.Rows[i].Cells[1].Value = day + "." + month + "." +
year;

    month++;
    if (month > 12)
    {
        month = 1;
        year++;
    }
}

```

```

if (vid == "диференційований")
{
    ss = ss + fullpay;
    if ((i + 1) == 1) { paymonth = (sum * pr); }
    else { paymonth = ((sum - ss) * pr); }
    fullpay = (sum / srok);
    pay = (paymonth + fullpay);
    ostatek = (ostatok - fullpay);
    obsum += pay;
    obproz += paymonth;
    itog = obsum * komiss + obproz;

}
else if (vid == "ануїтетний")
{
    ss = +pay;
    /*

    if ((i + 1) == 1) { paymonth = (sum * pr); }
    else { paymonth = ((sum - ss) * pr); }
    pay = ((pr * Math.Pow((1 + pr), srok)) / (Math.Pow((1 + pr),
srok) - 1)) * sum);

    */
    paymonth = ostatek * pr;
    pay = sum * (pr + pr / (Math.Pow((1 + pr), srok) - 1));
    fullpay = (pay - paymonth);
    fullpaysum = +fullpay;
    ostatek = (ostatok - fullpay);
    obsum = +pay;
    obproz = +paymonth;
    itog = obsum * komiss + obproz;

}
label6.AutoSize = true;
label2.AutoSize = true;
label3.AutoSize = true;
label6.Text = "Загальна сума платежу:";
label2.Text = "Переплата за процентами за кредит:";
label3.Text = "Підсумкова переплата з врахуванням комісії: ";
label7.Text = Convert.ToString(Math.Round(obsum, 2)) + " " + val;
label4.Text = Convert.ToString(Math.Round(obproz, 2)) + " " + val;
label5.Text = Convert.ToString(Math.Round(itog, 2)) + " " + val;

dataGridView1.Rows[i].Cells[2].Value = Math.Round(pay, 2) + " " +
val;

dataGridView1.Rows[i].Cells[3].Value = Math.Round(fullpay, 2) + " "
+ val;

dataGridView1.Rows[i].Cells[4].Value = Math.Round(paymonth, 2) + "
" + val;

dataGridView1.Rows[i].Cells[5].Value = Math.Round(ostatok, 2) + " "
+ val;

button3.Visible = true;
button2.Visible = true;
comboBox1.Visible = false;
comboBox2.Visible = false;
comboBox3.Visible = false;
comboBox4.Visible = false;
comboBox5.Visible = false;
numericUpDown1.Visible = false;
numericUpDown2.Visible = false;
numericUpDown3.Visible = false;
numericUpDown4.Visible = false;
dateTimePicker1.Visible = false;

```

```

        dataGridView1.Visible = true;
        label14.Visible = true;
        label14.Text = vid;
        label1.Text = "Вид платежу: ";
    }

}
catch
{
    MessageBox.Show("Виникла помилка.Перевірте введені дані");

    comboBox1.Visible = true;
    comboBox2.Visible = true;
    comboBox3.Visible = true;
    comboBox4.Visible = true;
    comboBox5.Visible = true;
    numericUpDown1.Visible = true;
    numericUpDown2.Visible = true;
    numericUpDown3.Visible = true;
    numericUpDown4.Visible = true;
    dateTimePicker1.Visible = true;
    button1.Visible = true;
    button2.Visible = false;
    button3.Visible = false;

    label14.Visible = false;
    dataGridView1.Visible = false;
    label1.Text = "Параметри кредиту";
}

}

private void comboBox1_SelectedIndexChanged(object sender, EventArgs e)
{
}

private void dataGridView1_CellContentClick(object sender,
DataGridViewCellEventArgs e)
{
}

private void tableLayoutPanel1_Paint(object sender, PaintEventArgs e)
{
}

private void dataGridView1_CellContentClick_1(object sender,
DataGridViewCellEventArgs e)
{
}

private void button2_Click_1(object sender, EventArgs e)
{
    Stream myStream;
    saveFileDialog1.Filter = "Текстовий файл (*.txt)|*.txt|CSV-файл
(*.csv)|*.csv|Bin-файл (*.bin)|*.bin|Bitmap files (*.bmp)|*.bmp|Image files
(*.jpg)|*.jpg";
    saveFileDialog1.FilterIndex = 2;
    saveFileDialog1.RestoreDirectory = true;

    if (saveFileDialog1.ShowDialog() == DialogResult.OK)
    {

```

```

        if ((myStream = saveFileDialog1.OpenFile()) != null)
        {
            StreamWriter myWritet = new StreamWriter(myStream);
            try
            {
                for (int i = 0; i < dataGridView1.RowCount-1 ; i++)
                {
                    var first = true;
                    for (int j = 0; j < dataGridView1.ColumnCount; j++)
                    {
                        if (!first) myWritet.Write(";");
myWritet.Write(dataGridView1.Rows[i].Cells[j].Value.ToString());
                        first = false;
                    }
                    myWritet.WriteLine();
                }
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.Message);
            }
            finally
            {
                MessageBox.Show("Таблица успешно збережена");
                myWritet.Close();
            }
            myStream.Close();
        }
    }

private void label1_Click(object sender, EventArgs e)
{
}

private void Form1_Load(object sender, EventArgs e)
{
}

private void button3_Click(object sender, EventArgs e)
{
    comboBox1.Visible = true;
    comboBox2.Visible = true;
    comboBox3.Visible = true;
    comboBox4.Visible = true;
    comboBox5.Visible = true;
    numericUpDown1.Visible = true;
    numericUpDown2.Visible = true;
    numericUpDown3.Visible = true;
    numericUpDown4.Visible = true;
    dateTimePicker1.Visible = true;
    button1.Visible = true;
    button2.Visible = false;
    button3.Visible = false;
    label6.Text = "Вид платежа:";
    label2.Text = "Сумма кредита:";
    label3.Text = "Срок кредита:";
    label7.Text = "Начало выплат:";
    label4.Text = "Процентная ставка: ";
    label5.Text = "Единовременная комиссия:";

    label14.Visible = false;
}

```

```
        dataGridView1.Visible = false;
        label1.Text = "Параметры кредита";
    }

    private void button4_Click(object sender, EventArgs e)
    {
        if (MessageBox.Show("Вы уверены, что хотите закрыть калькулятор?",
            Application.ProductName, MessageBoxButtons.YesNo) != DialogResult.No)
            Application.Exit();
    }

    private void numericUpDown1_ValueChanged(object sender, EventArgs e)
    {
    }

    private void numericUpDown3_ValueChanged(object sender, EventArgs e)
    {
    }
}
}
```

Додаток Г. Аналіз диференційованого платежу та графіка погашення кредиту

Кредит

Вид платежу: **диференційований** Змінити дані

Загальна сума платежів: 17343,75 \$

Переплата за процентами за кредит: 2343,75 \$

Підсумкова переплата з врахуванням комісії 3557,81 \$

	№	Дата платежу	Сума платежу	Сума погашення основного боргу	Сума нарахованих процентів	Залишок боргу
▶	1	3.1.2024	812,5 \$	625 \$	187,5 \$	14375 \$
	2	3.2.2024	804,69 \$	625 \$	179,69 \$	13750 \$
	3	3.3.2024	796,88 \$	625 \$	171,88 \$	13125 \$
	4	3.4.2024	789,06 \$	625 \$	164,06 \$	12500 \$
	5	3.5.2024	781,25 \$	625 \$	156,25 \$	11875 \$
	6	3.6.2024	773,44 \$	625 \$	148,44 \$	11250 \$
	7	3.7.2024	765,62 \$	625 \$	140,62 \$	10625 \$
	8	3.8.2024	757,81 \$	625 \$	132,81 \$	10000 \$
	9	3.9.2024	750 \$	625 \$	125 \$	9375 \$
	10	3.10.2024	742,19 \$	625 \$	117,19 \$	8750 \$
	11	3.11.2024	734,38 \$	625 \$	109,38 \$	8125 \$
	12	3.12.2024	726,56 \$	625 \$	101,56 \$	7500 \$
	13	3.1.2025	718,75 \$	625 \$	93,75 \$	6875 \$
	14	3.2.2025	710,94 \$	625 \$	85,94 \$	6250 \$
	15	3.3.2025	703,12 \$	625 \$	78,12 \$	5625 \$

Зберегти таблицю Закрити програму

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	1	03.01.2024	812,5 \$	625 \$	187,5 \$	14375,00											
2	2	03.02.2024	804,69 \$	625 \$	179,69 \$	13750,00											
3	3	03.03.2024	796,88 \$	625 \$	171,88 \$	13125,00											
4	4	03.04.2024	789,06 \$	625 \$	164,06 \$	12500,00											
5	5	03.05.2024	781,25 \$	625 \$	156,25 \$	11875,00											
6	6	03.06.2024	773,44 \$	625 \$	148,44 \$	11250,00											
7	7	03.07.2024	765,62 \$	625 \$	140,62 \$	10625,00											
8	8	03.08.2024	757,81 \$	625 \$	132,81 \$	10000,00											
9	9	03.09.2024	750 \$	625 \$	125 \$	9375,00											
10	10	03.10.2024	742,19 \$	625 \$	117,19 \$	8750,00											
11	11	03.11.2024	734,38 \$	625 \$	109,38 \$	8125,00											
12	12	03.12.2024	726,56 \$	625 \$	101,56 \$	7500,00											
13	13	03.01.2025	718,75 \$	625 \$	93,75 \$	6875,00											
14	14	03.02.2025	710,94 \$	625 \$	85,94 \$	6250,00											
15	15	03.03.2025	703,12 \$	625 \$	78,12 \$	5625,00											
16	16	03.04.2025	695,31 \$	625 \$	70,31 \$	5000,00											
17	17	03.05.2025	687,5 \$	625 \$	62,5 \$	4375,00											
18	18	03.06.2025	679,69 \$	625 \$	54,69 \$	3750,00											
19	19	03.07.2025	671,88 \$	625 \$	46,88 \$	3125,00											
20	20	03.08.2025	664,06 \$	625 \$	39,06 \$	2500,00											
21	21	03.09.2025	656,25 \$	625 \$	31,25 \$	1875,00											
22	22	03.10.2025	648,44 \$	625 \$	23,44 \$	1250,00											
23	23	03.11.2025	640,62 \$	625 \$	15,62 \$	625,00											
24																	



Додаток Д. Аналіз ануїтетного платежу та графіка погашення кредиту

Кредит

Вид платежу: ануїтетний

Змінити дані

Загальна сума платежів: 727,3 \$

Переплата за процентами за кредит: 8,98 \$

Підсумкова переплата з врахуванням комісії 59,89 \$

	№	Дата платежу	Сума платежу	Сума погашення основного боргу	Сума нарахованих процентів	Залишок боргу
▶	1	3.1.2024	727,3 \$	539,8 \$	187,5 \$	14460,2 \$
	2	3.2.2024	727,3 \$	546,55 \$	180,75 \$	13913,65 \$
	3	3.3.2024	727,3 \$	553,38 \$	173,92 \$	13360,27 \$
	4	3.4.2024	727,3 \$	560,3 \$	167 \$	12799,98 \$
	5	3.5.2024	727,3 \$	567,3 \$	160 \$	12232,68 \$
	6	3.6.2024	727,3 \$	574,39 \$	152,91 \$	11658,29 \$
	7	3.7.2024	727,3 \$	581,57 \$	145,73 \$	11076,72 \$
	8	3.8.2024	727,3 \$	588,84 \$	138,46 \$	10487,87 \$
	9	3.9.2024	727,3 \$	596,2 \$	131,1 \$	9891,67 \$
	10	3.10.2024	727,3 \$	603,65 \$	123,65 \$	9288,02 \$
	11	3.11.2024	727,3 \$	611,2 \$	116,1 \$	8676,82 \$
	12	3.12.2024	727,3 \$	618,84 \$	108,46 \$	8057,98 \$
	13	3.1.2025	727,3 \$	626,57 \$	100,72 \$	7431,41 \$
	14	3.2.2025	727,3 \$	634,41 \$	92,89 \$	6797 \$
	15	3.3.2025	727,3 \$	642,34 \$	84,96 \$	6154,66 \$

Зберегти таблицю

Закрити програму

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	1	03.01.2024	727,3 \$	539,8 \$	187,5 \$	14460,2									
2	2	03.02.2024	727,3 \$	546,55 \$	180,75 \$	13913,65									
3	3	03.03.2024	727,3 \$	553,38 \$	173,92 \$	13360,27									
4	4	03.04.2024	727,3 \$	560,3 \$	167 \$	12799,98									
5	5	03.05.2024	727,3 \$	567,3 \$	160 \$	12232,68									
5	6	03.06.2024	727,3 \$	574,39 \$	152,91 \$	11658,29									
7	7	03.07.2024	727,3 \$	581,57 \$	145,73 \$	11076,72									
3	8	03.08.2024	727,3 \$	588,84 \$	138,46 \$	10487,87									
9	9	03.09.2024	727,3 \$	596,2 \$	131,1 \$	9891,67									
0	10	03.10.2024	727,3 \$	603,65 \$	123,65 \$	9288,02									
1	11	03.11.2024	727,3 \$	611,2 \$	116,1 \$	8676,82									
2	12	03.12.2024	727,3 \$	618,84 \$	108,46 \$	8057,98									
3	13	03.01.2025	727,3 \$	626,57 \$	100,72 \$	7431,41									
4	14	03.02.2025	727,3 \$	634,41 \$	92,89 \$	6797									
5	15	03.03.2025	727,3 \$	642,34 \$	84,96 \$	6154,66									
6	16	03.04.2025	727,3 \$	650,37 \$	76,93 \$	5504,29									
7	17	03.05.2025	727,3 \$	658,5 \$	68,8 \$	4845,8									
8	18	03.06.2025	727,3 \$	666,73 \$	60,57 \$	4179,07									
9	19	03.07.2025	727,3 \$	675,06 \$	52,24 \$	3504,01									
0	20	03.08.2025	727,3 \$	683,5 \$	43,8 \$	2820,51									
1	21	03.09.2025	727,3 \$	692,04 \$	35,26 \$	2128,47									
2	22	03.10.2025	727,3 \$	700,69 \$	26,61 \$	1427,77									
3	23	03.11.2025	727,3 \$	709,45 \$	17,85 \$	718,32									
4															

