

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ БІЛІМ ЖӘНЕ ҒЫЛЫМ  
МИНИСТРЛІГІ

Сәтбаев Университеті

Кибернетика және ақпараттық технологиялар институты

Киберқауіпсіздік, ақпаратты өңдеу және сақтау

кафедрасы

Керімбаев Сапархан Ғабитұлы

«Банк -Несиелік калькулятор модулі» ақпараттық жүйесін жобалау

**ДИПЛОМДЫҚ ЖҰМЫС**

5B070300 – «Ақпараттық жүйелер» мамандығы

Алматы 2020

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ

Сәтбаев Университеті

Кибернетика және ақпараттық технологиялар институты

Киберқауіпсіздік, ақпаратты өңдеу және сақтау кафедрасы

**ҚОРҒАУҒА ЖІБЕРІЛДІ**

КАӨЖС кафедрасы меңгерушісі  
техн. ғыл. канд., ассистент-  
профессор

\_\_\_\_\_ Н.А. Сейлова

«\_\_» \_\_\_\_\_ 2020 ж.

## **ДИПЛОМДЫҚ ЖҰМЫС**

Тақырыбы: «Банк -Несиелік калькулятор модулі» ақпараттық жүйесін жобалау

5B070300 – «Ақпараттық жүйелер» мамандығы

Орындаған:

Керімбаев С.Ғ.

Ғылыми жетекші:

техн.ғыл.магистрі, тьютер

\_\_\_\_\_ М.Б. Бауыржан

«\_\_» \_\_\_\_\_ 2020 ж.

Алматы 2020

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ

Сәтбаев Университеті

Кибернетика және ақпараттық технологиялар институты

Киберқауіпсіздік, ақпаратты өңдеу және сақтау кафедрасы

5B070300 – Ақпараттық жүйелер

**БЕКІТЕМІН**

КАӨЖС кафедрасы меңгерушісі  
техн. ғыл. канд., ассистент-  
профессор

\_\_\_\_\_ Н.А. Сейлова  
«\_\_» \_\_\_\_\_ 2020 ж.

**Дипломдық жұмысты орындауға  
ТАПСЫРМА**

Білім алушы: Керімбаев Сапархан Ғабитұлы

Жобаның тақырыбы: «Банк -Несиелік калькулятор модулі» ақпараттық жүйесін  
жобалау

Университет Ректорының «27» қаңтар 2020 жылғы №762-б бұйрығымен  
бекітілген.

Орындалған жұмыстың өткізу мерзімі: «29»мамыр 2020 ж.

Дипломдық жұмыстың бастапқы мәліметтері: берілген тақырып бойынша  
әдебиеттерге шолу кезінде жиналған мәліметтер, теориялық материалдар  
жинау.

Дипломдық жұмыста қарастырылатын мәселелер тізімі:

- а) тақырып аясын талдау;
- ә) несиелік калькулятор үшін есептеу әдістерін таңдау;
- б) бағдарламалық қамтамасыздандыруды жасау.

Графикалық материалдардың тізімі (міндетті түрде қажет сызбалар  
көрсетілген): жұмыстың 12 слайдтан тұратын презентациясы көрсетіледі.

Негізгі әдебиет 9 кітаптан тұрады.

Дипломдық жұмысты даярлау  
**КЕСТЕСІ**

| Бөлім атаулары,<br>қарастырылатын мәселелер тізімі    | Ғылыми жетекшіге,<br>кеңесшілерге өткізу<br>мерзімі | Ескерту |
|---|---|---------|
| Тақырып аясын талдау                                  | 10.01.2020 - 08.02.2020                             |         |
| Несиелік калькулятор үшін<br>есептеу әдістерін таңдау | 09.02.2020 - 10.03.2020                             |         |
| Бағдарламалық<br>қамтамасыздандыруды жасау            | 11.03.2020 - 28.04.2020                             |         |

Дипломдық жұмыс бөлімдерінің кеңесшілері мен  
норма бақылаушының аяқталған жұмысқа қойған  
**қолтаңбалары**

| Бөлімдердің<br>атауы | Кеңесшілер<br>(аты-жөні, тегі,<br>ғылыми дәрежесі,<br>атағы) | Қол қойылған<br>мерзімі | Қолы |
|----------------------|--|-------------------------|------|
| БҚ жасау             | Кабдуллин М.А.,<br>ассистент                                 |                         |      |
| Норма<br>Бақылаушы   | Дуйсенбаева А.Н.,<br>техн.ғыл.магистрі,<br>лектор            |                         |      |

Ғылыми жетекшісі \_\_\_\_\_ М. Б. Бауыржан

Тапсырманы орындауға қабылдаған білім алушы \_\_\_\_\_ С.Ф. Керімбаев

Күні «27» қаңтар 2020 ж.

# ҚАЗАҚСТАН РЕСПУБЛИКАСЫНЫҢ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ

Сәтбаев Университеті

## Дипломдық жобаға

Ғылыми жетекшінің сын пікірі

Керімбаев Сапархан

5B070300 – Ақпараттық жүйелер

Тақырыбы: «Банк - Несиелік калькулятор модулі» ақпараттық жүйесін жобалау

Дипломдық жұмыс түсіндірме хат кіріспеден, қорытындыдан, пайдаланылған әдебиеттер тізімінен тұрады.

Дипломдық жұмыста несие параметрлерін есептеудің ақпараттық жүйесін жобалау мәселелері сипатталған. Тақырып бойынша алдын-ала зерттеу жүргізілді. Аннуитетті және дифференциалды төлемдерді есептеудің математикалық моделдері ұсынылған. Алдын-ала төлемді қысқарту және алдын-ала мерзімін қысқарту функциялары ұсынылған. Несие төлеу кестесін word және excel бағдарламалық қамтамада жүктеп алу орнатылған. Ай сайынғы төлемді графиг түрінде көру мүмкіндігі бар. Таңдалған қосымшаны жасау ортасы сипатталған. Қосымшаның бағдарламалық жасалуы сипатталып көрсетілген.

Керімбаев Сапархан дипломдық жұмыс орындау барысында өз беттілік көрсетіп, бағдарламалық қамтамалармен терең танысып және әдебиеттермен жұмыс істеген. Дипломдық жұмыс “Ақпараттық жүйелер” мамандығының бітіру жұмыстарына қойылатын талаптарға сәйкес.

Қорытындылай келе, Керімбаев Сапарханның дипломдық жобасын қорғауға жіберуге болады.

Ғылыми жетекші:

Техника ғылымдарының магистрі, тьютор

\_\_\_\_\_ М. Б. Бауыржан

«\_\_\_» \_\_\_\_\_ 2020 ж

**Протокол анализа Отчета подобия**

**заведующего кафедрой / начальника структурного подразделения**

Заведующий кафедрой / начальник структурного подразделения заявляет, что ознакомился(-ась) с Полным отчетом подобия, который был сгенерирован Системой выявления и предотвращения плагиата в отношении работы:

**Автор:** Керімбаев Сапархан

**Название:** Банк -Несиелік калькулятор модулі» ақпараттық жүйесін жобалау

**Координатор:** Мадина Бауыржан

**Коэффициент подобия 1:**8,8

**Коэффициент подобия 2:**4,7 **Замена букв:**1 **Интервалы:**0 **Микропробелы:**0

**Белые знаки:**0

**После анализа отчета отчета подобия заведующий кафедрой / начальник структурного подразделения констатирует следующее:**

- обнаруженные в работе заимствования являются добросовестными и не обладают признаками плагиата. В связи с чем, работа признается самостоятельной и допускается к защите;
- обнаруженные в работе заимствования не обладают признаками плагиата, но их чрезмерное количество вызывает сомнения в отношении ценности работы по существу и отсутствием самостоятельности ее автора. В связи с чем, работа должна быть вновь отредактирована с целью ограничения заимствований;
- обнаруженные в работе заимствования являются недобросовестными и обладают признаками плагиата, или в ней содержатся преднамеренные искажения текста, указывающие на попытки сокрытия недобросовестных заимствований. В связи с чем, работа не допускается к защите.

Обоснование:

.....  
.....  
.....  
.....  
.....

Дата

*Подпись заведующего кафедрой /  
начальника структурного подразделения*

**Окончательное решение в отношении допуска к защите, включая обоснование:**

.....  
.....  
.....  
.....  
.....  
.....

Дата

*Подпись заведующего кафедрой /  
начальника структурного подразделения*

**Протокол анализа Отчета подобия Научным руководителем**

Заявляю, что я ознакомился(-ась) с Полным отчетом подобия, который был сгенерирован Системой выявления и предотвращения плагиата в отношении работы:

**Автор:** Керімбаев Сапархан

**Название:** Банк -Несиелік калькулятор модулі» ақпараттық жүйесін жобалау

**Координатор:** Мадина Бауыржан

**Коэффициент подобия 1:** 8,8

**Коэффициент подобия 2:** 4,7

**Замена букв:** 1 **Интервалы:** 0 **Микропробелы:** 0 **Белые знаки:** 0

**После анализа Отчета подобия констатирую следующее:**

обнаруженные в работе заимствования являются добросовестными и не обладают признаками плагиата. В связи с чем, признаю работу самостоятельной и допускаю ее к защите;

обнаруженные в работе заимствования не обладают признаками плагиата, но их чрезмерное количество вызывает сомнения в отношении ценности работы по существу и отсутствием самостоятельности ее автора. В связи с чем, работа должна быть вновь отредактирована с целью ограничения заимствований;

обнаруженные в работе заимствования являются недобросовестными и обладают признаками плагиата, или в ней содержатся преднамеренные искажения текста, указывающие на попытки сокрытия недобросовестных заимствований. В связи с чем, не допускаю работу к защите.

Обоснование:

.....

.....

*Дата*

.....

*Подпись Научного руководителя*



ДАТА ОТЧЕТА: 2020-05-23 19:23:21

**НАЗВАНИЕ:**

Банк -модулдік несиелік калькулятор» ақпараттық жүйесін жобалау

**АВТОР:**

Керімбаев Сапархан

**НАУЧНЫЙ РУКОВОДИТЕЛЬ:**

Мадина Бауыржан

**ПОДРАЗДЕЛЕНИЕ:**

ИКИИТ

**ДАТА ЗАГРУЗКИ ДОКУМЕНТА:**

2020-05-23 19:21:50

**ЧИСЛО ПРОВЕРЕК ДОКУМЕНТА:** 

1

**ПРОПУЩЕННЫЕ ВЕБ-СТРАНИЦЫ:** **Уровень заимствований**

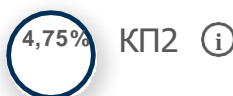
Обратите внимание! Высокие значения коэффициентов не означают плагиат. Отчет должен быть проанализирован экспертом.



% комбинаций из 5 слов, найденных во всех доступных источниках, кроме БЮА

**25**

Длина фразы для коэффициента подобия 2



% комбинации 25-слов, найденных во всех доступных источниках, кроме БЮА

**5537**

Количество слов



% найденных слов в тексте, помеченных как цитаты

**48380**

Количество символов

**Предупреждение и сигналы тревоги**

В этом разделе вы найдете информацию, касающуюся манипуляций в тексте, с целью изменить результаты проверки. Для того, кто оценивает работу на бумажном носителе или в электронном формате, манипуляции могут быть невидимы (может быть также целенаправленное вписывание ошибок). Следует оценить, являются ли изменения преднамеренными или нет.

|  |   |                   |
|--|---|-------------------|
| Замена букв  | 1 | показать в тексте |
| Использование символов из другого алфавита - может указывать на способ обойти систему, поэтому следует установить их использование.  |   |                   |
| Интервалы  | 0 | показать в тексте |
| Количество увеличенного расстояния между буквами (просим определить является ли расстояние имитацией пробела, так как исходно слова могут быть написаны слитно).   |   |                   |
| Микропробелы   | 0 | показать в тексте |
| Количество пробелов с нулевым размером - необходимо проверить влияют ли они на неправильное разделение слов в тексте.  |   |                   |
| Белые знаки  | 0 | показать в тексте |
| Количество символов, выделенных белым цветом, пожалуйста, проверьте не используются ли белые символы вместо пробела, соединяя слова (в отчете подобия система изменяет автоматически цвет букв в черный, чтобы их сделать видимыми). |   |                   |

**Заимствования по списку источников**

Посмотрите список и проанализируйте, в особенности, те фрагменты, которые превышают КП №2 (выделенные жирным шрифтом). Используйте ссылку «Обозначить фрагмент» и посмотрите, являются ли выделенные фрагменты повтоящимися короткими фразами, разбросанными в документе (совпадающие сходства), многочисленными короткими фразами расположенные рядом друг с другом (парафразирование) или обширными фрагментами без указания источника ("криптоцитаты").

**10 самых длинных фраз (6,34 %)**

Десять самых длинных фрагментов найденных во всех доступных ресурсах.

| ПОРЯДКОВЫЙ НОМЕР | НАЗВАНИЕ И АДРЕС ИСТОЧНИКА URL (НАЗВАНИЕ БАЗЫ)   | АВТОР                  | КОЛИЧЕСТВО ИДЕНТИЧНЫХ СЛОВ |
|------------------|--|------------------------|----------------------------|
| 1                | ЖОО-ның аудиториялық қорын есепке алу және рұқсаттама беру жүйесін әзірлеу<br><i>D. Serikbayev East Kazakhstan State Technical University (ОПуМУП)</i> | Оспанхан Ф.Қ. 15-ВТК-2 | 119 2,15 %                 |
| 2                | ЖОО-ның аудиториялық қорын есепке алу және рұқсаттама беру жүйесін әзірлеу<br><i>D. Serikbayev East Kazakhstan State Technical University (ОПуМУП)</i> | Оспанхан Ф.Қ. 15-ВТК-2 | 47 0,85 %                  |
| 3                | ЖОО-ның аудиториялық қорын есепке алу және рұқсаттама беру жүйесін әзірлеу<br><i>D. Serikbayev East Kazakhstan State Technical University (ОПуМУП)</i> | Оспанхан Ф.Қ. 15-ВТК-2 | 41 0,74 %                  |

|    |   |                        |    |        |
|----|---|------------------------|----|--------|
| 4  | ЖОО-ның аудиториялық қорын есепке алу және рұқсаттама беру жүйесін әзірлеу<br><i>D. Serikbayev East Kazakhstan State Technical University (ОПУМУП)</i>  | Оспанхан Ф.Қ. 15-ВТК-2 | 29 | 0,52 % |
| 5  | ЖОО-ның аудиториялық қорын есепке алу және рұқсаттама беру жүйесін әзірлеу<br><i>D. Serikbayev East Kazakhstan State Technical University (ОПУМУП)</i>  | Оспанхан Ф.Қ. 15-ВТК-2 | 27 | 0,49 % |
| 6  | <a href="https://prograide.com/pregunta/53527/comment-">https://prograide.com/pregunta/53527/comment-</a>   |                        | 23 | 0,42 % |
| 7  | <a href="https://mpt.ru/o-tehnikume/obrazovanie/files/090207-specialist-po-informacionnym-sistemam/090207-IS-PM03.pdf">https://mpt.ru/o-tehnikume/obrazovanie/files/090207-specialist-po-informacionnym-sistemam/090207-IS-PM03.pdf</a> |                        | 20 | 0,36 % |
| 8  | <a href="https://stylopedia.ru/4x5c82.html">https://stylopedia.ru/4x5c82.html</a>   |                        | 20 | 0,36 % |
| 9  | <a href="https://stylopedia.ru/4x5c82.html">https://stylopedia.ru/4x5c82.html</a>   |                        | 13 | 0,23 % |
| 10 | Разработка информационной системы ресторана «Niku»<br><i>D. Serikbayev East Kazakhstan State Technical University (ОПУМУП)</i>  | Нурпеисов А.Б 15-ВТТ-1 | 12 | 0,22 % |

### из базы данных RefBooks (0,00 %)

Все фрагменты найдены в базе данных RefBooks, которая содержит более 3 миллионов текстов от редакторов и авторов.

| ПОРЯДКОВЫЙ НОМЕР         | НАЗВАНИЕ | АВТОР | КОЛИЧЕСТВО ИДЕНТИЧНЫХ СЛОВ (КОЛИЧЕСТВО ФРАГМЕНТОВ) |
|--------------------------|----------|-------|--|
| ЗАИМСТВОВАНИЙ НЕ НАЙДЕНО |          |       |  |

### из домашней базы данных (0,00 %)

Все фрагменты найдены в базе данных вашего университета.

| ПОРЯДКОВЫЙ НОМЕР         | НАЗВАНИЕ | АВТОР | ДАТА ИНДЕКСАЦИИ (ФРАГМЕНТОВ) | ИДЕНТИЧНЫХ СЛОВ (КОЛИЧЕСТВО ФРАГМЕНТОВ) |
|--------------------------|----------|-------|------------------------------|---|
| ЗАИМСТВОВАНИЙ НЕ НАЙДЕНО |          |       |                              |   |

### из программы обмена базами данных (6,34 %)

Все фрагменты найдены в базе данных других университетов.

| ПОРЯДКОВЫЙ НОМЕР | НАЗВАНИЕ БАЗЫ ДАННЫХ   | АВТОР                               | ДАТА ИНДЕКСАЦИИ | КОЛИЧЕСТВО ИДЕНТИЧНЫХ СЛОВ (КОЛИЧЕСТВО ФРАГМЕНТОВ) |        |
|------------------|--|-------------------------------------|-----------------|--|--------|
| 1                | ЖОО-ның аудиториялық қорын есепке алу және рұқсаттама беру жүйесін әзірлеу<br><i>D. Serikbayev East Kazakhstan State Technical University (ОПУМУП)</i>                           | Оспанхан Ф.Қ. 15-ВТК-2              | 2019-05-16      | 273 (6)  | 4,93 % |
| 2                | Разработка информационной системы ресторана «Niku»<br><i>D. Serikbayev East Kazakhstan State Technical University (ОПУМУП)</i>   | Нурпеисов А.Б 15-ВТТ-1              | 2018-05-15      | 30 (3)   | 0,54 % |
| 3                | Проектирование и разработка подсистемы работы с клиентами ИП "Black&White"<br><i>D. Serikbayev East Kazakhstan State Technical University (ОПУМУП)</i>                           | Шогелов А.Е. 15-ВТТ-1               | 2018-05-16      | 22 (4)   | 0,40 % |
| 4                | Проект АРМ по работе с вкладчиками коммерческого банка<br><i>Kyrgyz National Agrarian University named after Skryabin (Кафедра прикладной информатики информационных систем)</i> | Дороева Жылдыз Узакбекович          | 2017-06-09      | 16 (2)   | 0,29 % |
| 5                | ФАКТОРНЫЙ АНАЛИЗ В ИССЛЕДОВАНИИ ФИНАНСОВОГО СОСТОЯНИЯ КОМПАНИИ<br><i>Kazakh University of Economics, Finance and International Trade (KUEFIT) (Кафедра «Финансы»)</i>            | Кашкинбекова Гульмира Каирберлиевна | 2019-06-11      | 10 (1)   | 0,18 % |

### из интернета (2,47 %)

Все фрагменты найдены в глобальных интернет-ресурсах открытого доступа.

| ПОРЯДКОВЫЙ НОМЕР | ИСТОЧНИК URL  | ИДЕНТИЧНЫХ СЛОВ (ФРАГМЕНТОВ) |        |
|------------------|---|------------------------------|--------|
| 1                | <a href="https://stylopedia.ru/4x5c82.html">https://stylopedia.ru/4x5c82.html</a>   | 33 (2)                       | 0,60 % |
| 2                | <a href="https://urait.ru/book/ekonomicheskaya-informatika-426110">https://urait.ru/book/ekonomicheskaya-informatika-426110</a> | 29 (4)                       | 0,52 % |

|   |   |        |        |
|---|---|--------|--------|
| 3 | <a href="https://prograide.com/pregunta/53527/comment-generer-des-noms-de-couleur-en-c">https://prograide.com/pregunta/53527/comment-generer-des-noms-de-couleur-en-c</a>   | 23 (1) | 0,42 % |
| 4 | <a href="https://mpt.ru/o-tehnikume/obrazovanie/files/090207-specialist-po-informacionnym-sistemam/090207-IS-PM03.pdf">https://mpt.ru/o-tehnikume/obrazovanie/files/090207-specialist-po-informacionnym-sistemam/090207-IS-PM03.pdf</a>                         | 20 (1) | 0,36 % |
| 5 | <a href="https://docplayer.ru/68477508-Upravlenie-kommunikativnymi-processami-vzaimodeystviya-institutov-vlasti-i-naseleniya.html">https://docplayer.ru/68477508-Upravlenie-kommunikativnymi-processami-vzaimodeystviya-institutov-vlasti-i-naseleniya.html</a> | 17 (2) | 0,31 % |
| 6 | <a href="https://metanit.com/sharp/windowsforms/4.20.php">https://metanit.com/sharp/windowsforms/4.20.php</a>   | 15 (2) | 0,27 % |

## АҢДАТПА

Дипломдық жұмыста несиелік параметрлерін есептеудің ақпараттық жүйесін жобалау мәселелері сипатталған. Тақырып бойынша алдын-ала зерттеу жүргізілді. Аннуитетті және дифференциалды төлемдерді есептеудің математикалық моделдері ұсынылған. Алдын-ала төлемді қысқарту және алдын-ала мерзімін қысқарту функциялары ұсынылған. Несиелік төлеу кестесін word және excel бағдарламалық қамтамада жүктеп алу орнатылған. Ай сайынғы төлемді график түрінде көру мүмкіндігі бар. Таңдалған қосымшаны жасау ортасы сипатталған. Қосымшаның бағдарламалық жасалуы сипатталып көрсетілген.

## АННОТАЦИЯ

В работе описаны вопросы проектирования информационной системы расчёта параметров кредитного займа. Проведено исследование предметной области. Приведены математические модели расчета аннуитентного и дифференцированного платежа. Предложены функции уменьшения предоплаты и уменьшения срока. График платежей по кредиту можно скачать в программном обеспечении Word и Excel. Ежемесячный платеж можно посмотреть в виде графика. Описана выбранная среда разработки приложения. Приведено описание программной разработки приложения.

## **ABSTRACT**

The graduation project describes issues of designing an information system for calculating credit loan parameters. A pre-project study of the subject area was carried out. Mathematical models for calculating annuity and differentiated payments are presented. Functions of reducing prepayment and reducing the term are proposed. Loan payment schedule can be downloaded in Word and Excel software. Monthly payment can be viewed in the form of a graph. The selected application development environment is described. The description of the software development application.

## МАЗМҰНЫ

|  |    |
|--|----|
| КІРІСПЕ  | 9  |
| 1 Несиелік калькулятордың қазіргі таңдағы жағдайы            | 10 |
| 1.1 Жұмыстың мақсаты мен міндеттерін қою                     | 10 |
| 1.2 Тақырып аясын сипаттау                                   | 10 |
| 1.3 Несиелік калькулятор – құрылымы мен жіктелуі             | 11 |
| 2 Несиелік калькулятор үшін есептеу әдістерін таңдау         | 14 |
| 2.1 Несие бойынша төлемдерді есептеудің математикалық моделі | 14 |
| 2.2 Бағдарламалық іске асыру құралы                          | 15 |
| 3 Бағдарламалық қамтамасыздандыруды жасау                    | 17 |
| 3.1 Бағдарламаны іске асыру                                  | 17 |
| 3.1.1 Төлем графигін есептеу                                 | 18 |
| 3.1.2 Төлемнің аннуитеттік түрі                              | 20 |
| 3.1.3 Төлемнің дифференциалды түрі                           | 23 |
| 3.1.4 Мерзімінен бұрын төлеу                                 | 26 |
| 3.1.5 Электронды поштаны қолдану                             | 27 |
| 3.1.6 Графикті файлда сақтау                                 | 29 |
| 3.1.7 График құру  | 32 |
| ҚОРЫТЫНДЫ  | 38 |
| ПАЙДАЛАНЫЛҒАН ӘДЕБИЕТТЕР ТІЗІМІ                              | 39 |
| Қосымшалар   | 40 |

## КІРІСПЕ

Берілген дипломдық жұмыстың мақсаты «Банк -несиелік калькулятор модулі» ақпараттық жүйесін жасау және жобалау болып табылады.

Зерттеу объектісі ретінде банктің несиелік төлемдерін есептеу қызметі алынды.

Зерттеу нысаны несиелік төлемдерін есептеу ақпараттық жүйесін жасау және жобалау болып табылады.

Біздің еліміздің тиімділігі жоғары экономикасын дамыған, аяғына нық тұрған несиелік нарығынсыз елестету мүмкін емес. Бүгінгі таңда несиелік өте танымал және экономикалық қатынастардың барлық салаларына әсер етеді.

Адамдар пайыздық мөлшерлеменің, несиелік қайтару мерзімінің және сәйкесінше ай сайынғы төлемнің қалай есептелетінін білмей-ақ банктерден несиелік ала береді.

Біздің отандастарымыздың қаржылық сауаттылығының төменділігі кесірінен көптеген адамдар несиелік қалай қайтару керектігі және қанша үстінен төлейтіні туралы ойламай несиелік үсті үстіне алып жатыр.

Несиелік калькуляторы (банктік несиелік калькуляторы) - коммерциялық банктер ашып көрсететін төлем сомасын, артық төлемді, нақты пайыздық мөлшерлемені немесе тиімді пайыздық мөлшерлемені онлайн есептеу үшін қажет. Сондай-ақ, төлемнің қай бөлігі несиелік негізгі сомасын қайтаруға, қай бөлігі несиелік бойынша сыйақы төлеуге кететінін білуге болады. Белгіленген тұрақты пайыздық мөлшерлемені таңдауға немесе пайыздық мөлшерлемені енгізуге болады. Несиелік калькуляторы төлемдердің екі түрін онлайн режимінде шешуге мүмкіндік береді: аннуитеттік төлем - банктік несиелік бойынша есептелген сыйақы сомасы мен негізгі қарыз сомасын қамтитын мөлшерде ай сайынғы несиелік төлемі, оны көптеген коммерциялық банктер пайдаланады; сараланған төлем - бұл төлем кезінде төлем сомасы өзгермейтін төлем. Несиелік калькуляторы - несиелік түрлерін салыстыру және банк мамандарының көмегінсіз қажетті ақпаратты алу үшін қолданылады. Белгіленген тұрақты пайыздық мөлшерлемені таңдауға немесе пайыздық мөлшерлемені енгізуге болады.

Несиелік калькулятор қолданушыға үйден шықпай-ақ несиелік есептеуге ғана емес, қолда бар ұсыныстарды төлем әдісі мен төлем кестесін таңдау мүмкіндігімен салыстыруға мүмкіндік береді. Мұның бәрі қолданушыларға санды шатастырмауға, қолданушыға түсетін материалдық жүктеме деңгейін объективті бағалауға мүмкіндік береді.



## **1 Несиелік калькулятордың қазіргі таңдағы жағдайы**

### **1.1 Жұмыстың мақсаты мен міндеттерін қою**

Берілген дипломдық жұмыстың мақсаты «Банк - несиелік калькулятор модулі» ақпараттық жүйесін жасау және жобалау болып табылады.

Зерттеу объектісі ретінде банктің несиелік төлемдерін есептеу қызметі алынды.

Зерттеу нысаны несиелік төлемдерін есептеу ақпараттық жүйесін жасау және жобалау болып табылады.

Алдыға қойылған мақсатқа жету үшін келесі мәселелерді қарастыру керек:

- тақырып бойынша диплом алдындағы зерттеуді жүргізу;
- несиелік төлемдерін есептеудің математикалық моделін құру;
- несиелік төлемдерін есептеудің бизнес-процесін моделдеуді орындау;
- «Банк - несиелік калькулятор модулі» несиелік төлемдерін есептеудің ақпараттық жүйесіне қойылатын талаптарды сипаттау;
- жүйені жасау ортасын таңдау;
- «Банк - несиелік калькулятор модулі» несиелік төлемдерін есептеудің ақпараттық жүйесінің визуалды интерфейсін жасау;
- ақпараттық жүйені сынақтан өткізуді іске асыру.

### **1.2 Тақырып аясын сипаттау**

Несие – бұл бір меншік иесінің, яғни несиелік берушінің несиелік берудің негізгі қағидалары болып табылатын: тауарлық, ақшалай немесе тез қайтарылуына сәйкес материалдық емес кез-келген түрдегі құндылықтарды уақытша пайдалануға басқа несиелік берушіге, яғни қарыз алушыға беруін қарастыратын экономикалық қатынастар жүйесі [1].

Несиелік бірнеше түрлері бар:

- Банк несиелік банктен немесе басқа да несиелік - қаржы ұйымдарынан ақшалай несиелік түрінде нарықта жұмыс істейтін кәсіпкерлерге беріледі. Ақша капиталы басты объект болып табылады. Банктік несиелік коммерциялық несиелік қарағанда әлдеқайда кең қолданылады, өйткені коммерциялық несиелік операциялардың бағытымен, ұзақтығымен және көлемімен шектеледі, ал банктік несиелік несиелік ыңғайлы етеді, оның шекарасы мен аясы кеңейеді, қауіпсіздік артады.

- Коммерциялық несиелік бір кәсіпкердің екіншісіне кейінге қалдырылған төлемімен тауарларды сату түрінде беріледі. Вексельмен тіркелген. Тауарлық капитал басты объект болып табылады. Оның міндеті - тауарларды сатуды жеделдету.

- Банк несиесі сияқты берілетін тұтынушылық несие, яғни тұтынушылық немесе коммерциялық қажеттіліктерге арналған несие, тұтынушыларға несие мерзімі кейінге қалдырылған тауарлар үшін беріледі.

- Ипотекалық несие - жылжымайтын мүлік үшін ұзақ мерзімді несие.

- Мемлекеттік несие - жеке немесе заңды тұлғаларға қатысты мемлекет немесе жергілікті билік органдары несие беруші немесе несие алушы болып табылатын несие.

- Халықаралық несие – бұл халықаралық экономикалық қатынастар саласындағы несиелік капиталдың қозғалысына байланысты валюталық және тауарлық ресурстар түрінде берілген қарыз.

- Ауылшаруашылық несиесі - ауылшаруашылығына ірі инвестицияларды жабу үшін банктер беретін ұзақ мерзімді несие.

- Автонесие - жеке тұлғаларға автокөлік сатып алуға берілетін несие [2].

Несиелік калькуляторда келесі үш түрлі несиені есептеу жиі қолданылады:

- тұтынушылық несие;

- ипотекалық несие;

- автонесие.

Кез-келген несиені есептеудің бірінші және негізгі қағидасы - кез-келген маманға да, бағдарламаға да несиені есептеу үшін кейбір параметрлер қажет болады:

- Талап етіліп отырған несие көлемі.

- Егер несиелік калькулятор қолданылса, жеке жеңілдіктерге байланысты несие бойынша пайыздық мөлшерлеме. Егер банкке келіп кеңес алса, маман несиелеудің бірнеше бағдарламасын ұсынады, пайыз мөлшерлемесін ескере отырып, белгілі бір бағдарламаның пайдалылығын түсіндіреді.

- Несиенің есептік мерзімі - егер несиелік калькулятор қолданылса.

Сәйкесінше, банк белгілеген мерзім ішінде төленуі керек.

- Несиенің осы түріне берілген сома - несиелік калькуляторда есептеу кезінде бұл параметр енгізілмейді, бірақ банктің несиелік бағдарламасының нақты шегінде болу үшін оны біліп алу керек.

- Бір жеке тұлғаның немесе оның отбасының табысы негізінде төлем қабілеттілігін есептеу. Несиелік калькуляторда бұл параметр қандай калькулятор қолданылатынына байланысты есептеуге енгізілмейді немесе енгізіледі.

- Аннуитеттік немесе дифференциалды төлем схемасын таңдау. Математикалық моделдері бар төлем схемаларының толық сипаттамасы осы жобаның 2-тарауында келтірілген.

### **1.3 Несиелік калькулятор – құрылымы мен жіктелуі**

Несиелік калькулятор – бұл әртүрлі несие параметрлерін есептеумен байланысты амалдарды орындауға қабілетті бағдарлама. Ол несиенің негізгі

параметрлерін тез есептеу үшін және төлем кестесің құру үшін қолданылады. Сонымен қатар, қолжетімді несиенің әр түрін салыстыруға, төлем кестесі бойынша төлеудің тиімді жоспарын құруға банкке бармай ақ қол жеткізуге болады. Бұл банкке барып уақыт кетіргісі келмейтін, кезек күтіп тұрғысы келмейтін адамдар үшін ыңғайлы, заманауи бағдарлама. Қазіргі кезде көптеген банктер өз клиенттеріне ерекше назар аударады және клиенттерінің де, өз қызметкерлерінің де уақытын үнемдеуге тырысады. Сондықтан банк сайттарынан «Несиелік калькулятор» деген бөлімді кездестіруге болады, ол жерде клиент несиені бойынша төлем сомасы мен шарттарын алдын ала есептеп алуына болады.

Әртүрлі банктердің несиелік калькулятор құрылымы жағынан ұқсас болуы мүмкін және бірнеше негізгі элементтерден тұрады, vtb-bank.kz сайтының онлайн калькуляторы мысал ретінде көрсетілген (1.1-сурет).

Несие сомасы, несиелеу мерзімі мен пайыздық мөлшерлеме негізгі параметрлер болып табылады. Банкте несиені есептеу кезінде осы үш көрсеткіш негізге алынады.

Сондай ақ, көрсетілген калькулятор өтеу схемасын есептеу кезінде аннуитеттік немесе дифференциалды төлем схемасын таңдауға мүмкіндік береді.

### Кредитный калькулятор

|                     |                     |
|---------------------|---------------------|
| Расчет 1            | + Добавить расчет   |
| Тип погашения       | Аннуитетные платежи |
| Сумма займа (тенге) | 20000               |
| Срок кредита (мес.) | 12                  |
| Процентная ставка   | 12                  |
| <b>Рассчитать</b>   |                     |

1.1 Сурет – Несиелік калькулятордың on-line интерфейсі

Қолданушы барлық қажетті параметрлерді енгізіп болған соң «Рассчитать» батырмасын басу керек. Сол кезде сервис есептеу нәтижесін көрсетеді (1.2-сурет).

|                        |              |
|------------------------|--------------|
| Ежемесячный платеж     | 1 777 тенге  |
| Общая сумма выплат:    | 21 324 тенге |
| Общая сумма переплаты: | 1 324 тенге  |

## 1.2 Сурет – Есептеу нәтижесі

on-line несиелік калькуляторды пайдалану өте ыңғайлы. Бірақ олардың да мүмкіндіктері шектеулі, жоспарланған жартылай алдын ала төлемдерді есептей және жоспарлай алмайды, төлем графигін құра алмайды. Сонымен қатар, сервис мәліметтері интернетке тәуелді, яғни тек интернет бар кезде ғана қолдана аласыз. Қазіргі уақытта несие бойынша төлемдерді есептеу функциясын іске асыратын жергілікті бағдарламалық қамтамасыз етуді әзірлеу өзектілігі туралы қорытынды жасауға болады.

Осылайша, қолданыстағы сервистерге жасалынған шолу жасалынған және жобаланған «Банк -модулдік несиелік калькулятор» ақпараттық жүйесінің жұмысына қойылатын талаптарды қалыптастыруға мүмкіндік берді.

Қосымшада:

- ыңғайлы және түсінікті графигтік интерфейс болу;
- қолданушыға төлемді есептеу схемасын таңдауға мүмкіндік беру (аннуитеттік немесе дифференциалды төлем);
- қолданушыға төлем кестесін құруға мүмкіндік беру (дөңгелек немесе баған түрінде диаграмма);
- әрбір айдың толық төлемін, қарыз сомасын және пайыз сомасын көрсету (негізгі қарыз және пайызы);
- қолданушыға жоспарланған жартылай төлемнің мөлшері мен күнін енгізу мүмкіндігін және қосымша төлемдерді қайта есептеу мүмкіндігін беру;
- пайдаланушыға төлем кестесін MS Word және MS Excel-де экспорттау мүмкіндігін беру;
- қолданушыға төлем кестесін почтаға жіберуге мүмкіндік беру;
- қолданушыға валютаны таңдауға мүмкіндік беру (теңге немесе доллар);
- қолданушыға төлеу күнін таңдауға мүмкіндік беру (несиені алған күн мен соңғы күн);
- қолданушыға пайыздық мөлшерлемені таңдауға мүмкіндік беру;
- пайдаланушыға қарыздың мерзімін көрсетуге мүмкіндік беру (айы мен жылы) керек.

## 2 Несиелік калькулятор үшін есептеу әдістерін таңдау

### 2.1 Несие бойынша төлемдерді есептеудің математикалық моделі

Несие бойынша төлемдерді есептеу үшін аннуитеттік схема мен дифференциалды схема қолданылады.

Несиеге бас кезінде дифференциалды төлем көбірек болады да, сосын біртіндеп азаяды, яғни несиенің ай сайынғы төлемдері бірдей емес. Дифференциалды төлем құрылымы екі бөлімнен тұрады:

- қарыз сомасын төлеуге кететін бүкіл мерзімге белгіленген сома;
- азаятын бөлігі - несие қалдығы сомасынан есептелетін несие бойынша пайыздар.

Несие сомасы тұрақты азайып отырғандықтан пайыздық мөлшерлеме де және ай сайынғы төлем де азаяды. Негізгі қарызды өтеу мөлшерін есептеу үшін бастапқы қарыз сомасын қарыз мерзіміне бөлу қажет (кезеңдер санына):

$$HK = \frac{BC}{KC}, \quad (1.1)$$

мұндағы НК - негізгі қарызды қайтару; BC – несиенің бастапқы сомасы; KC – кезең саны.

Бір жылда 12 ай болғандықтан, ай сайынғы пайыздық мөлшерлеме көлемі келесі формуламен анықталады:

$$TP = HK \cdot \frac{TPM}{12}, \quad (1.2)$$

мұндағы TP - төленетін пайыздар; НК - осы айдағы несиенің қалғаны; TPM — жылдық пайыздық мөлшерлеме.

Аннуитеттік, яғни тең төлемдер - бұл несиенің барлық мерзімі ішіндегі бірдей төлемдер. Бұл сома тараптардың келісімі бойынша немесе жартылай алдын-ала төлеу жағдайында ғана өзгертілуі мүмкін. Аннуитеттік төлем құрылымы екі бөліктен тұрады:

- несиені пайдаланғаны үшін пайыздар,
- несиені өтейтін сома.

Уақыт өте келе бұл қатынас өзгереді және пайыздар біртіндеп азаяды, сәйкесінше негізгі қарызды төлеуге кеткен сома аннуитеттік төлем ішінде артады.

Аннуитеттік төлем көлемі келесі формуламен есептеледі:

$$AT = BC \cdot \frac{TPM}{1 - (1 + TPM)^{-MC}}, \quad (1.3)$$

мұндағы АТ - аннуитеттік төлем; ТПМ – төлем төлеу кезіндегі пайыздық мөлшерлеме; БС – несиенің бастапқы сомасы; МС – мерзім саны.

Егер төлем ай сайын төленсе, онда МС – төленетін айлардың саны, ал ТПМ - айлық пайыздық мөлшерлеме (1/12 жылына).

## 2.2 Бағдарламалық іске асыру құралы

Microsoft Visual Studio – бұл Windows ОЖ үшін қосымшаларды жасауға арналған консолдық және графикалық интерфейсі бар бағдарламалық орта.

Кешенге келесі негізгі бөліктер кіреді:

1. Visual Basic.NET - VisualBasic-те қосымшаларды жасау үшін.
2. Visual C++ - дәстүрлі C++ тілінде.
3. Visual C# - C# тілінде (Microsoft).
4. Visual F# - F тілінде# (Microsoft Developer Division).

Ортаның функционалды құрылымына келесілер кіреді:

- IntelliSense автотолтыру сияқты көптеген қосымша функцияларды қамтитын бастапқы код редакторы, кодты рефакторлау және т.б.;

- кодты түзету құралы;

- графикалық интерфейстерді тез құрастыруға арналған форма редакторы;

- Web-редактор;

- сынып дизайнері;

- мәліметтер базасы сызбаларының дизайнері.

Сонымен қатар Visual Studio кез-келген деңгейдегі функционалдылықты кеңейтуге арналған басқа да қосымшаларды (плагиндерді) құруға және қосуға, соның ішінде бастапқы код нұсқаларын басқару жүйелерін (Subversion және VisualSourceSafe) қолдауды, жаңа құралдар жиынтығын қосуды (бағдарламалаудың пәндік-бағытталған тілдерінде кодты өңдеу және визуалды жобалау үшін немесе бағдарламалық қамтамасыздандырудың басқа да аспектілеріне арналған құралдарды өңдеу үшін) ескере отырып мүмкіндік береді.

Visual Studio-ны (IntegratedDevelopmentEnvironment-IDE) әзірлеудің интегралдық ортасы кодты базалық басқару кезіндегі жоғары деңгейлік бірқатар функционалдық мүмкіндіктерді ұсынады.

Төменде Visual Studio-ның IDE ортасының басты артықшылықтары келтірілген:

Енгізілген Web-сервер. ASP.NET Web-қосымшасына қызмет көрсету үшін Web-сұранымдарды күтетін және сәйкес парақтарды өңдейтін Web-сервер қажет. Visual Studio-да интегралданған Web-сервердің болуы Web-сайтты дизайн ортасынан тікелей іске қосуға мүмкіндік береді, сонымен қатар кез-келген сыртқы компьютерден сынақ тапсыратын Web-сайтқа кіру мүмкіндігін болдырмайды және қауіпсіздікті арттырады, өйткені тестілік сервер тек жергілікті компьютермен ғана байланыса алады.

Бағдарламаны құру кезінде көптеген тілдерге қолдау көрсету. Visual Studio әрдайым бірдей интерфейсті (IDE) қолдана отырып, өзіңіздің тіліңізде немесе кез-келген басқа тілде код жазуға мүмкіндік береді. Сонымен қатар, Visual Studio сізге әртүрлі тілдерде Web-беттерді құруға мүмкіндік береді, бірақ барлығын бірдей Web-қосымшада орналастырады. Тек бір ғана кемшілік - әрбір Web-парақта бір ғана тілді қолдануға болады (әйтпесе, керісінше жағдайда, компиляция кезінде көп проблемалар болуы мүмкін).

Бағдарлама жазу кезінде аз кодты қолдану. Көптеген қосымшалар стандартты стереотиптік кодты және Web-парақтарын талап етеді, соның бірі ASP.NET. Мысалы, басқарудың Web-элементін қосу, оқиғаны өңдеушілерді қосу және форматтауды түзету бет орналасуында бірқатар мәліметтерді орнатуды қажет етеді. Visual Studio-да мұндай бөлшектер автоматты түрде орнатылады.

Кодтаудың интуитивті стилі. Реті бойынша, Visual Studio қажетті шегіністі автоматты түрде енгізіп, түсініктеме сияқты элементтерді бөлек көрсету үшін түсті кодтауды қолдана отырып кодтарды түсуіне қарай форматтайды. Бұл кішігірім айырмашылықтар кодты оқу үшін ыңғайлы етеді және қатені аз жасауға көмектеседі. Қолданылатын Visual Studio форматтау параметрлерін автоматты түрде орнатуға мүмкіндік береді, бұл бағдарламаны жасаушы жақшаларды орналастырудың басқа стилін таңдаған кезде өте ыңғайлы (мысалы, ашылған жақша хабарландыру берілген жолда орналасатын K&R стилі).

Тезірек әзірлену жылдамдығы. Visual Studio-ның көптеген мүмкіндіктері бағдарламаны жасаушыға мүмкіндігінше тез жұмыс жасауға көмектесуге бағытталған. Ыңғайлы функциялар, мысалы, IntelliSense функциясы (қателерді анықтап, дұрыс жолын ұсына алады), іздеу және ауыстыру функциялары (кілт сөздерді бір файлда да, бүкіл жобада да іздеуге мүмкіндік береді) және пікірлерді автоматты түрде қосу және жою функциясы (код блоктарын уақытша жасыра алады) әзірлеушіге тез әрі тиімді жұмыс жасауға мүмкіндік береді.

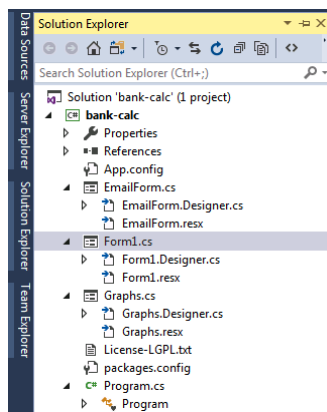
Жөндеу мүмкіндіктері. Visual Studio ұсынған жөндеу құралдары құпия қателерді бақылау және оғаш қылықты зерттеу үшін ең жақсы құрал болып табылады. Әзірлеуші өз кодын жол бойынша бірден орындай алады және кез-келген уақытта жадыдан ағымдағы ақпаратты көре алады.

Visual Studio-да жобаны басқару мүмкіндігі, бастапқы кодты басқарудың кірістірілген функциясы, кодты рефакторизациялау мүмкіндігі, кеңейтудің қуатты моделі сияқты басқа да көптеген функциялар бар. Сонымен қатар, егер Visual Studio 2008 Team System-ді қолданса, әзірлеуші блокты модулді тестілеуге, бірлесе отырып жұмыс жасауға және кодты өңдеуге арналған жетілдірілген мүмкіндіктерге ие болады (Visual SourceSafe сияқты қарапайым құралдардан мүмкіндігі көбірек).

### 3 Бағдарламалық қамтамасыздандыруды жасау

#### 3.1 Бағдарламаны іске асыру

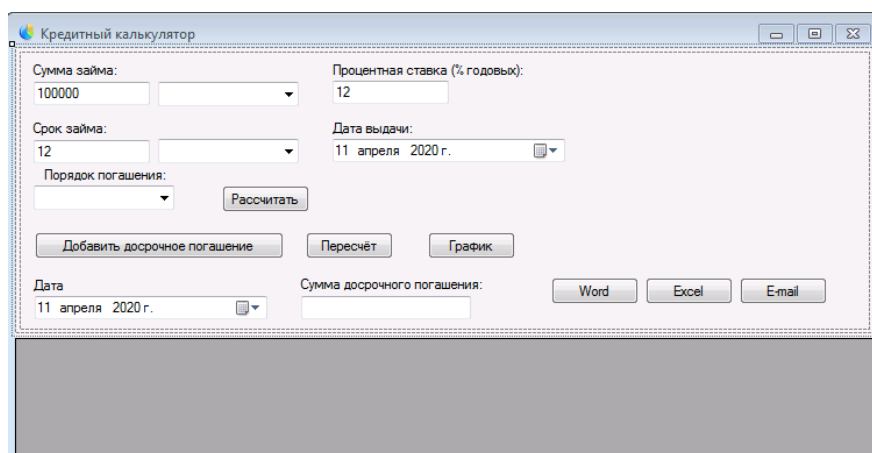
Жобаның жалпы құрылымы 3.1-суретте көрсетілген.



3.1 Сурет – Visual Studio-дағы жобаның жалпы құрылымы

Қосымшаны жасауды несиелік параметрлерін енгізу өрісі, төлеу түрін таңдау менюі, валюта мен мерзімі (жылы, айы) орналасатын негізгі формадан бастаймыз.

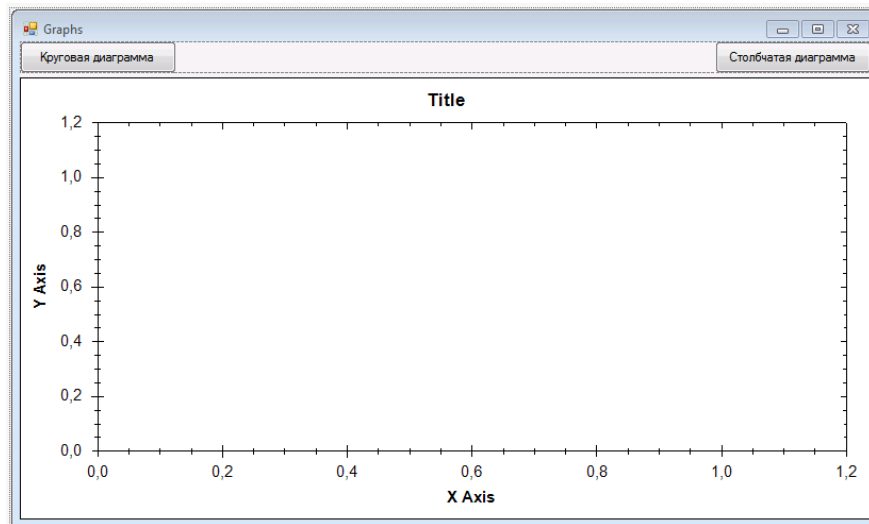
Сонымен қатар, формада несиені төлеу графигі көрсетілетін кесте де болады. Форманың сыртқы түрі 3.2-суретте берілген.



3.2 Сурет – Негізгі форманың сыртқы түрі

Негізгі формадан басқа есептеу нәтижесі көрсетілетін графигі бар форма да қажет болады. Форманың сыртқы түрі 3.3-суретте көрсетілген.





3.3 Сурет –Графигі бар форманың сыртқы түрі

Электронды поштаға жіберу формасы 3.4-суретте көрсетілген.

A screenshot of a software window titled "EmailForm". It contains several input fields: "Адрес SMTP сервера", "Порт SMTP сервера", "От кого", "Логин пользователя", "Пароль", "Кому", and "Тема письма". Below these is a "Вложение" (Attachment) section with a "Выбрать файл" (Choose file) button. At the bottom is a "Текст письма" (Message text) section with a large text area and an "Отправить" (Send) button.

3.4 Сурет – Электронды поштаға жіберу формасы

### 3.1.1 Төлем графигін есептеу

Төлем графигін есептемей тұрып жасаушы несие сомасын, пайыздық мөлшерлемесін, алу уақытын мәтін түрінде толтыру керек. Жолдар TextBox түрінде жазылады, жолға енгізілген мәндерді санға айналдыру үшін

float.Parse() әдісін қолданамыз. Көрсетілген жолдарды толтырғаннан кейін «Рассчитать» батырмасын басу керек. Сол кезде келесі жолдар шығады:

```
private void button1_Click(object sender, EventArgs e)
{
    try
    {
        calculate_Credit();
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}
```

Жасаушы графикті есептеу әдісін іске қосты:

```
calculate_Credit();
```

Бұл әдіс мәтіні А қосымшасында толық көрсетілген. График туралы барлық ақпарат арнайы топтамада сақталады:

```
List<Payment> payments;
```

Payment класы графиктің ағымдағы жазбасы туралы ақпаратты сақтайды және келесі түрде болады:

```
public class Payment
{
    public int Num { get; set; } //номер
    public DateTime date { get; set; } //мерзімі
    public double Sum { get; set; } //төлем сомасы
    public double Dolg { get; set; } //негізгі қарыз
    public double Percent { get; set; } //қосылған %
    public double Left { get; set; } //қалдығы
    public double Advance { get; set; } //мерзімінен бұрын төлеу
}
```

Бұл кластың барлық жолдары автоматты сипаттар түрінде көрсетілген. Графикті есептеудің ең басында calculate\_Credit() әдісінде топтаманы ескі мәндерден тазарту және оларды бос мәндермен инициализациялау іске асырылады:

```

payments.Clear();
n = float.Parse(timeTB.Text);
if (dateCB.SelectedItem.ToString() == "год") n *= 12;
for (int k = 0; k < (int)n; k++)
{
    payments.Add(new Payment());
}

```

flag логикалық айнымалысы жаңа есептеулер болды ма немесе мерзімінен бұрын төлем болды ма, соны анықтайды. Жаңа есептеулер болса бірінші мәнді есептеу графигімен топтамаға қосамыз:

```

payments[0].Num = 1;
payments[0].date = dateTimePicker1.Value;
payments[0].Left = float.Parse(sumTB.Text);

```

1- есептеу жолының реттік саны, күнді таңдау жолынан қай күн екені және сәйкес мәтіндік жолдан несиенің толық сомасы енгізіледі. Ары қарай қосымшаны жасаушы формада таңдаған төлем түрі анықталады.

### 3.1.2 Төлемнің аннуитеттік түрі

Аннуитеттік төлем түрі үшін келесі айнымалылар қолданылады:

```

double V;
double S;
double Q;
double Sn;
double Qn;
int i;
int j;

```

мұндағы, S- несиенің қалғаны, Q- ай сайынғы пайыз, Qn- кезекті төлемнің пайызы, Sn- кезекті төлем сомасы.

Есептеу алгоритмінің бастапқы мәндерін есептейміз:

```

S = float.Parse(sumTB.Text);
Q = float.Parse(percentTB.Text) / 100 / 12;
V = S * Q / (1 - Math.Exp(Math.Log(1 + Q) * (-n)));
i = 1;
j = 1;
Sn = S;

```

Ары қарай циклда қалған төлемдердің графигін есептейміз:

```
while (j < n)
{
    Qn = Sn * Q;
    S = V - Qn;
    payments[i].Num = i + 1;
    payments[i].date = datePicker1.Value.AddMonths(j);
    payments[i].Sum = Math.Round(V, 2);
    payments[i].Dolg = Math.Round(S, 2);
    payments[i].Percent = Math.Round(Qn, 2);
    Sn = Sn - S;
    if (payments[i].Advance > 0)
    {
        V = V - payments[i].Advance / (n - i - 1);
    }
    payments[i].Left = Math.Round(Sn, 2);

    i++;
    j++;
}
```

Әрбір қадамда несие параметрлерін қайта есептеп отырамыз, сондай ақ, осы айда мерзімінен бұрын төлем болған жоқ па, тексеріп отырамыз. Егер төлем болса, қайта есептеу жүргіземіз:

$$V = V - \text{payments}[i].\text{Advance} / (n - i - 1);$$

Қайта есептеуден кейін қалған соманың сәйкес мәнін топтамаға жазамыз:

$$\text{payments}[i].\text{Left} = \text{Math.Round}(\text{Sn}, 2);$$

Нақты мәндердің барлығын жазған кезде олар келесі әдіс көмегімен үтірден кейін екі белгіге дейін дөңгелектенеді:

$$\text{Math.Round}(\text{Sn}, 2)$$

Төлем графигінің соңғы мәні бөлек есептеледі:

```
Qn = Sn * Q;
S = V - Qn;
Sn = 0;
if (flag) payments.Add(new Payment{
```

```

    Num = i + 1,
    date = dateTimePicker1.Value.AddMonths(j),
    Sum = Math.Round(V, 2),
    Dolg = Math.Round(S, 2),
    Percent = Math.Round(Qn, 2),
    Left = Math.Round(Sn, 2)
});
else
{
    payments[i].Num = i + 1;
    payments[i].date = dateTimePicker1.Value.AddMonths(j);
    payments[i].Sum = Math.Round(V, 2);
    payments[i].Dolg = Math.Round(S, 2);
    payments[i].Percent = Math.Round(Qn, 2);
    payments[i].Left = Math.Round(Sn, 2);
}

```

Бұл мәнді есептеу кезінде бұл жаңа есептеу ма, әлде мерзімінен бұрын төлем болғаннан кейінгі қайта есептеу ма, сол анықталады. Егер жаңа есептеу болса, онда топтамаға жаңа мән қосылады:

```

if (flag) payments.Add(new Payment
{
    Num = i + 1,
    date = dateTimePicker1.Value.AddMonths(j),
    Sum = Math.Round(V, 2),
    Dolg = Math.Round(S, 2),
    Percent = Math.Round(Qn, 2),
    Left = Math.Round(Sn, 2)
});

```

Егер бұл сол графиктегі қайта есептеу болса, онда жаңа мәнің орнына бұрынғы мән өзгертіледі:

```

else
{
    payments[i].Num = i + 1;
    payments[i].date = dateTimePicker1.Value.AddMonths(j);
    payments[i].Sum = Math.Round(V, 2);
    payments[i].Dolg = Math.Round(S, 2);
    payments[i].Percent = Math.Round(Qn, 2);
    payments[i].Left = Math.Round(Sn, 2);
}

```

Осымен аннуитеттік төлем түрі аяқталады.

### 3.1.3 Төлемнің дифференциалды түрі

Дифференциалды төлем басқа алгоритммен есептеледі. Есептеу алгоритм параметрлерінің бастапқы мәндерін есептеуден басталады:

```
S = float.Parse(sumTB.Text);
n = float.Parse(timeTB.Text);
if (dateCB.SelectedItem.ToString() == "ГОД") n *= 12;
Q = float.Parse(percentTB.Text) / 100 / 12;
d = S / n;
i = 1;
j = 1;
Sn = S;
```

Есептеу графигінің бірінші мәні алдында енгізіліп қойғаннан кейін қалған мәндерге көшеміз: циклда ағымдағы мәнді есептейміз

```
while (j < n)
{
    p = Sn * Q;
    V = d + p;
    payments[i].Num = i + 1;
    payments[i].date = dateTimePicker1.Value.AddMonths(j);
    payments[i].Sum = Math.Round(V, 2);
    payments[i].Dolg = Math.Round(d, 2);
    payments[i].Percent = Math.Round(p, 2);
    if (payments[i].Advance > 0)
    {
        d = (S - payments[i].Advance) / (n);
    }
    Sn = Sn - d;
    payments[i].Left = Math.Round(Sn, 2);
    i++;
    j++;
}
```

Егер есептеудің бір жолында алдын ала төлем жасалғаны анықталса, онда төлем шамаларын азайту пропорционалды түрде болады:

```
if (payments[i].Advance > 0){
    d = (S - payments[i].Advance) / (n);
```

```
}  
Sn = Sn - d;
```

Цикл аяқталған соң графиктің соңғы мәні есептеледі:

```
p = Sn * Q;  
S = Sn - p;  
V = d + p;  
Sn = 0;  
if (flag) payments.Add(new Payment  
{  
    Num = i + 1,  
    date = dateTimePicker1.Value.AddMonths(j),  
    Sum = Math.Round(V, 2),  
    Dolg = Math.Round(d, 2),  
    Percent = Math.Round(p, 2),  
    Left = Math.Round(Sn, 2)  
});  
else  
{  
    payments[i].Num = i + 1;  
    payments[i].date = dateTimePicker1.Value.AddMonths(j);  
    payments[i].Sum = Math.Round(V, 2);  
    payments[i].Dolg = Math.Round(d, 2);  
    payments[i].Percent = Math.Round(p, 2);  
    payments[i].Left = Math.Round(Sn, 2);  
}
```

Егер жаңа график есептелінсе, онда топтамаға жаңа мән қосылады:

```
if (flag) payments.Add(new Payment  
{  
    Num = i + 1,  
    date = dateTimePicker1.Value.AddMonths(j),  
    Sum = Math.Round(V, 2),  
    Dolg = Math.Round(d, 2),  
    Percent = Math.Round(p, 2),  
    Left = Math.Round(Sn, 2)  
});
```

Егер мерзімінен бұрын төлем жасалынып, төлем графигінде қайта есептеу жүргізілсе, онда топтамадағы соңғы мән өзгертіледі:

```

else
{
    payments[i].Num = i + 1;
    payments[i].date = dateTimePicker1.Value.AddMonths(j);
    payments[i].Sum = Math.Round(V, 2);
    payments[i].Dolg = Math.Round(d, 2);
    payments[i].Percent = Math.Round(p, 2);
    payments[i].Left = Math.Round(Sn, 2);
}

```

Берілген төлем түрі бойынша графикті есептеу аяқталған соң есептеу флагын орнатамыз.

```
flag = true;
```

Сосын төлем графигі кестесін келесі формада жаңалаймыз:

```
updateGridView();
```

График кестесі қосымшаны іске қосқан кезде инициализацияланатын DataGridView түрінің элементі болып табылады:

```

dataGridView1.ColumnCount = 7;
dataGridView1.Columns[0].Name = "Р.т. №";
dataGridView1.Columns[1].Name = "Мерзімі";
dataGridView1.Columns[2].Name = "Төлем сомасы";
dataGridView1.Columns[3].Name = "Негізгі қарыз";
dataGridView1.Columns[4].Name = "Қосылған %";
dataGridView1.Columns[5].Name = "Қалдығы";
dataGridView1.Columns[6].Name = "Мерзімінен бұрын";

```

Бағанға гридті қосқаннан кейін есептеу мәліметтері бар жолды қосуға кірісе берсе болады. updateGridView әдісінде гридті тазалау тізбектей орындалады, сосын жаңа мәндер толтырылады. Әдістің коды:

```

dataGridView1.Rows.Clear();
foreach (var item in payments)
{
    string[] row = new string[] { item.Num.ToString(),
        item.date.ToString("d-m-y"),
        item.Sum.ToString(),
        item.Percent.ToString(),
        item.Dolg.ToString(),

```



```

        item.Left.ToString(),
        item.Advance.ToString());
dataGridView1.Rows.Add(row);
    }

```

Гридқа жаңа мәнді (қатарды) қосу foreach() құрастырғышы көмегімен іске асырылады және топтама арқылы итерация кезінде гридқа қосылатын массив қатары құрылады. Ағымдағы элемент мәнін жолға түрлендіру ToString() әдісі арқылы жүзеге асырылады, ал қай күн екенін білу үшін күн-ай-жыл форматында шығарылатын форматты маска қолданылады.

```

        item.Num.ToString(),
        item.date.ToString("d"),
        item.Sum.ToString(),
        item.Percent.ToString(),
        item.Dolg.ToString(),
        item.Left.ToString(),
        item.Advance.ToString()

```

Массив жолға түрлендірілгеннен кейін ол гридқа қосылады:

```

dataGridView1.Rows.Add(row);

```

### 3.1.4 Мерзімінен бұрын төлеу

Мерзімінен бұрын төлеуді қосу механизмі келесі түрде болады:

1. Қолданушы төлем мерзімін көрсетеді.
2. Қолданушы төлем сомасын көрсетеді.
3. Қолданушы «Мерзімінен бұрын төлеуді қосу» батырмасын басады.
4. Қолданушы «Қайта есептеу» батырмасын басады.

Төлемнің мерзімі мен сомасы енгізілгеннен кейін қолданушы «Төлемді қосу» батырмасын басады. Сол кезде келесі жол шығады:

```

private void button2_Click(object sender, EventArgs e)
{
    double val = float.Parse(textBox4.Text);
    DateTime dt = dateTimePicker2.Value;
    for (int j = 0; j < payments.Count - 1; j++)
    {
        if (dt >= payments[j].date && dt < payments[j + 1].date){
            payments[j].Advance = val;
            flag = false;
            updateGridView();
        }
    }
}

```

```

        return;
    }
}
}

```

Ең бірінші төлемнің мерзімі мен сомасы анықталады. Сосын төлем графигінің топтамасы бойынша циклда мерзімінен бұрын төлеуді қай жолға қосу керектігі анықталады. Ол үшін ағымдағы төлемнің қолданушы енгізген мерзімі мен төлемнің қолданушы енгізген келесі мерзімі салыстырылады. Егер сондай қатар табылса ондағы мерзімінен бұрын төлем мәні жаңаланады да, циклды орындау return командасының көмегімен аяқталады. Мәнді енгізгеннен кейін гридтің сыртқы түрін жаңалау процедурасын іске қосу мен flag айнымалысын өзгерту жүзеге асырылады.

«Қайта есептеу» батырмасын басыу flag=false мәнімен іске асырылады.

### 3.1.5 Электронды поштаны қолдану

Төлем графигін құрастырғаннан кейін қолданушы оны электронды пошта арқылы жібере алады. Ол үшін 3.4-суретте көрсетілген электронды хатты жіберу формасын ашатын E-mail батырмасы бар. Көрсетілген форманы ашқаннан кейін қолданушы керекті қатарды толтырып, керек файлды қосу керек. Содан кейін хатты жіберуге болады. Электронды поштамен жұмыс істеу үшін SmtпClient[1] класы қолданылады. Бұл класс көмегімен хатты жіберу үшін оған сервер адресі, оның порты, серверді қолдану үшін логині мен паролі сияқты міндетті параметрлерді жіберу керек. Код келесі түрде болады:

```

// smtp-сервер адресі мен хатты жіберетін порт
SmtпClient smtp = new SmtпClient(textBox1.Text,
int.Parse(textBox2.Text));
// логин мен пароль
smtp.Credentials = new NetworkCredential(textBox5.Text,
textBox6.Text);
smtp.EnableSsl = true;

```

Ары қарай хатты жіберуге дайындалу керек. Ол үшін жіберуші, хатты алушы, хат тақырыбы мен мәтіні туралы ақпарат қажет болады. Сондай ақ, жіберілуге тиіс файлды қосу керек. Хат System.Net.Mail есімдер кеңістігінің MailMessage[2] класының экземплярлары көмегімен жазылады. Хат келесі түрде құрастырылады:

```

// жіберуші – адресін жазамыз және аттағы есімді жазамыз
MailAddress from = new MailAddress(textBox3.Text);
// кімге жібереміз
MailAddress to = new MailAddress(textBox4.Text);
// хабарлама объектінің құрамын

```

```

MailMessage m = new MailMessage(from, to);
if (!String.IsNullOrEmpty(fileName)) m.Attachments.Add(new
Attachment(fileName));
// хат тақырыбы
m.Subject = textBox7.Text;
// хат мәтіні
m.Body = textBox8.Text;
// хат html кодында жазылған
m.IsBodyHtml = true;

```

Хатты қосу келесі әдіспен іске асырылады:

```
m.Attachments.Add(new Attachment(fileName));
```

Аттачды құрастыру үшін оған файл аты жіберіледі. Ол ат қолданушы «Выбрать файл на форме» батырмасын басқан кезде толтырылады. Келесі түрде болады:

```

private void button2_Click(object sender, EventArgs e)
{
    openFileDialog1.ShowDialog();
    if (openFileDialog1.ShowDialog() == DialogResult.Cancel) return;
    else fileName = openFileDialog1.FileName;
}

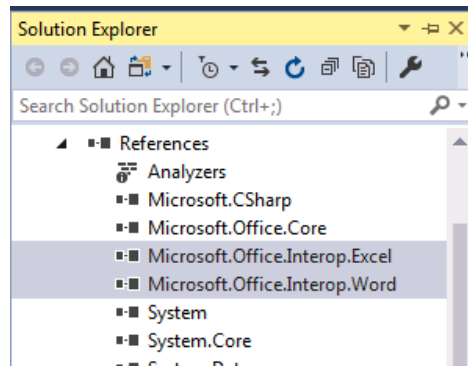
```

Батырманы басқан кезде openFileDialog1.ShowDialog() файлын таңдау диалогы ашылады және егер қолданушы файлды таңдап, ОК батырмасын басса, онда openFileDialog1 объектісінің FileName қатарында жүйедегі файлға апаратын барлық жолдар жазылады. Ол жолды fileName айнымалысында сақтаймыз.

Ары қарай келесі әдіс көмегімен хатты жіберуге болады: smtp.Send(m);  
Осымен поштаны қолдану аяқталады.

### 3.1.6 Графикті файлда сақтау

Бағдарламада төлем графигі бар файлды Microsoft Excel-де және Microsoft Word-та екі түрде құрастыруға болады. Microsoft Office-пен жұмыс істеу үшін пайдаланар алдында олардың сәйкес сілтемелерін жазып алу керек. Ол үшін жоба ішінен тышқанның оң жақ батырмасын баса отырып жоба атына кіріп, Добавить -> Ссылку дегенді басу керек және пайда болған терезеге Microsoft Word Object Library және Microsoft Excel Object Library үшін сілтемелерді қосу керек. Сосын жоба ішіндегі сілтеме тізімінде екі жаңа сілтеме пайда болуы керек (3.5-сурет).



### 3.5 Сурет – Жобадағы Microsoft Office – ке сілтеме

Excel форматында есеп беруді құру үшін формадағы сәйкес батырманы басу керек. Төлем графигіне негізделген, кестеде көрсетілген Excel форматында форма құруды орындайтын өңдеуші шақырылады. Өңдеуші коды келесі түрде болады:

```
private void button6_Click(object sender, EventArgs e)
{
    dataGridView1.ClipboardCopyMode =
DataGridViewClipboardCopyMode.EnableAlwaysIncludeHeaderText;
    dataGridView1.MultiSelect = true;
    dataGridView1.SelectAll();
    DataObject dataObj = dataGridView1.GetClipboardContent();
    if (dataObj != null)
        Clipboard.SetDataObject(dataObj);
    Microsoft.Office.Interop.Excel.Application xlexcel;
    Microsoft.Office.Interop.Excel.Workbook xlWorkBook;
    Microsoft.Office.Interop.Excel.Worksheet xlWorkSheet;
    object misValue = System.Reflection.Missing.Value;
    xlexcel = new Microsoft.Office.Interop.Excel.Application();
    xlexcel.Visible = true;
    xlWorkBook = xlexcel.Workbooks.Add(misValue);
    xlWorkSheet =
(Microsoft.Office.Interop.Excel.Worksheet)xlWorkBook.Worksheets.get_Item(1);
    Microsoft.Office.Interop.Excel.Range CR =
(Microsoft.Office.Interop.Excel.Range)xlWorkSheet.Cells[1, 1];
    CR.Select();
    xlWorkSheet.PasteSpecial(CR, Type.Missing, Type.Missing,
Type.Missing, Type.Missing, Type.Missing, true);
}
```

Басында гридтағы барлық жолдар таңдалады және олар алмасу буферіне көшіріледі. Сосын бір парақ пен бір кітапты қамтитын жаңа құжат құрастырылады. Содан кейін деректерді алмасу буферінен параққа енгізу жүреді. Енгізілгеннен кейін алынған файл экранда көрінеді. Содан кейін қолданушы файлмен не істеу керектігін таңдай алады: ол оны одан әрі жіберу немесе басып шығару үшін сақтай алады немесе жай ғана сақтамай жаба алады.

Word форматындағы файлдармен жұмыс басқаша жүргізіледі. Ең бірінші этапта есептеу кестесінің құрамын уақытша объектіге сақтаймыз:

```
int RowCount = DGV.Rows.Count;
int ColumnCount = DGV.Columns.Count;
Object[,] dataArray = new object[RowCount + 1, ColumnCount + 1];
//add rows
int r = 0;
for (int c = 0; c <= ColumnCount - 1; c++)
{
    for (r = 0; r <= RowCount - 1; r++)
    {
        dataArray[r, c] = DGV.Rows[r].Cells[c].Value;
    }
}
```

Сосын жаңа файл дайындаймыз:

```
Microsoft.Office.Interop.Word.Document wordDoc = new
Microsoft.Office.Interop.Word.Document();
wordDoc.Application.Visible = true;
Зададим портретную ориентацию страницы:
wordDoc.PageSetup.Orientation =
Microsoft.Office.Interop.Word.WdOrientation.wdOrientLandscape;
```

Содан кейін деректері бар кестені, кестенің тақырыбын, беттің тақырыбын құрастырамыз және мәтін үшін стиль орнатамыз:

```
dynamic oRange = wordDoc.Content.Application.Selection.Range;
string tmp = "";
for (r = 0; r <= RowCount - 1; r++)
{
    for (int c = 0; c <= ColumnCount - 1; c++)
    {
        tmp = tmp + dataArray[r, c] + "\t";
    }
}
```

```

//table format
oRange.Text = tmp;
object Separator =
Microsoft.Office.Interop.Word.WdTableFieldSeparator.wdSeparateByTabs;
object ApplyBorders = true;
object AutoFit = true;
object AutoFitBehavior =
Microsoft.Office.Interop.Word.WdAutoFitBehavior.wdAutoFitContent;
oRange.ConvertToTable(ref Separator, ref RowCount, ref
ColumnCount,
Type.Missing, Type.Missing, ref ApplyBorders,
Type.Missing, Type.Missing, Type.Missing,
Type.Missing, Type.Missing, Type.Missing,
Type.Missing, ref AutoFit, ref AutoFitBehavior,
Type.Missing);
oRange.Select();
wordDoc.Application.Selection.Tables[1].Select();
wordDoc.Application.Selection.Tables[1].Rows.AllowBreakAcrossPages = 0;
wordDoc.Application.Selection.Tables[1].Rows.Alignment = 0;
wordDoc.Application.Selection.Tables[1].Rows[1].Select();
wordDoc.Application.Selection.InsertRowsAbove(1);
wordDoc.Application.Selection.Tables[1].Rows[1].Select();
//стилдері
wordDoc.Application.Selection.Tables[1].Rows[1].Range.Bold = 1;
wordDoc.Application.Selection.Tables[1].Rows[1].Range.Font.Name =
"Tahoma";
wordDoc.Application.Selection.Tables[1].Rows[1].Range.Font.Size =
14;
//кесте үшін тақырып қосамыз
for (int c = 0; c <= ColumnCount - 1; c++)
{
wordDoc.Application.Selection.Tables[1].Cell(1, c + 1).Range.Text
= DGV.Columns[c].HeaderText;
}
wordDoc.Application.Selection.Tables[1].Rows[1].Select();
wordDoc.Application.Selection.Cells.VerticalAlignment =
Microsoft.Office.Interop.Word.WdCellVerticalAlignment.wdCellAlignVerticalCenter;
foreach (Microsoft.Office.Interop.Word.Section section in
wordDoc.Application.ActiveDocument.Sections)
{

```

```

        Microsoft.Office.Interop.Word.Range headerRange =
section.Headers[Microsoft.Office.Interop.Word.WdHeaderFooterIndex.wdHeaderFo
oterPrimary].Range;
        headerRange.Fields.Add(headerRange,
Microsoft.Office.Interop.Word.WdFieldType.wdFieldPage);
        headerRange.Text = "Несиелік калькулятор";
        headerRange.Font.Size = 16;
        headerRange.ParagraphFormat.Alignment =
Microsoft.Office.Interop.Word.WdParagraphAlignment.wdAlignParagraphCenter;
    }

```

Бұдан кейін экранда қолданушы шығара алатын, дискке сақтай алатын немесе жауып тастай алатын өзі құрастырған құжат пайда болады.

### 3.1.7 График құру

Төлем графигін кескіндеудің тағы бір жолы график көмегімен кескіндеу болып табылады. Графиктің екі түрін таңдауға болады: шеңбер түріндегі диаграмма және баған түріндегі диаграмма. Графикті кескіндеу үшін График батырмасын басу керек. Ол батырманы басу келесі түрде болады:

```

private void button4_Click(object sender, EventArgs e)
{
    if (payments.Count == 0) return;
    Graphs g = new Graphs(payments);
    g.Show();
}

```

Ол төлем графигін есептеу болды ма, соны тексереді және нәтиже бар топтаманы графикті кескіндеу формасын құрастырушыға жібереді. Терезені құрастырушы топтаманы жергілікті жабық айнымалыға сақтайды да, шеңбер түріндегі диаграммада көрсетеді:

```

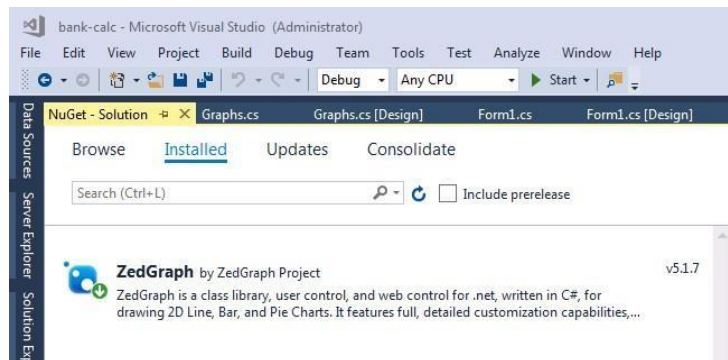
public Graphs(List<Payment> payments)
{
    if (payments.Count == 0) this.Close();
    InitializeComponent();
    this._payments = payments;
    drawPie();
}

```

Графикті кескіндеу үшін ZedGraph[3] кітапханасы қолданылады. Оны жобаға қосу үшін Инструменты -> Менеджер мәзірінен NuGet жобасын ашып, іздеу жолына ZedGraph деп жазу керек.

Сосын шыққан нәтиже тізімінен қажетті пакетті таңдап, орнатуды басу керек.

Нәтижесінде жобаға жаңа пакет қосылатын болады (3.6-сурет).



3.6 Сурет – Жаңа пакетті жобаға қосу

Ары қарай жобаға сыртында графиктер кескінделетін арнайы ZedGraphControl компоненті қосылады.

Шеңбер түріндегі диаграмманы кескіндеу drawPie() әдісінің көмегімен іске асырылады. Графикті салар алдында компонент алдыңғы нәтижелерден тазаланады:

```
zedGraph.GraphPane.CurveList.Clear();  
zedGraph.GraphPane.GraphObjList.Clear();
```

Сосын график үшін тақырып беріледі:

```
GraphPane myPane = zedGraph.GraphPane;  
myPane.Title.Text = "Төлем графигі";
```

Одан кейін циклда форма құрастырғышқа берілген төлем графигі топтамасының элементтері қайта таңдалады және графикке қарыз параметрінің мәні қосылады:

```
for (int i = 0; i < _payments.Count; i++)  
{  
    PieItem item = myPane.AddPieSlice(_payments[i].Dolg,  
getRandomColor(), 0F, _payments[i].Dolg.ToString());  
    item.LabelType = PieLabelType.Value;  
}
```

График секторы үшін түс getRandomColor() әдісінің көмегімен кездейсоқ таңдалады. Коды келесі түрде болады:



```

Random randomGen = new Random();
private Color getRandomColor()
{
    KnownColor[] names =
(KnownColor[])Enum.GetValues(typeof(KnownColor));
    KnownColor randomColorName =
names[randomGen.Next(names.Length)];
    Color randomColor = Color.FromKnownColor(randomColorName);
    return randomColor;
}

```

Бұл әдісте ең бірінші қол жетімді түстер массиві, содан кейін массивтің элементтерінің бірі кездейсоқ таңдалады және функциядан қайтарылады.

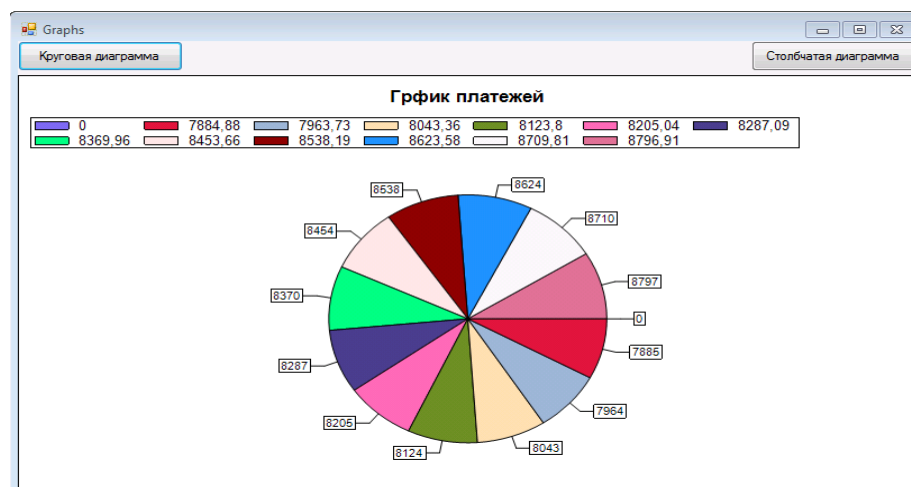
Соңғы этапта барлық деректер графикте көрсетілу үшін осьтерді масштабтау орындалады және сурет салу басталады:

```

zedGraph.AxisChange();
// Графикті жаңалаймыз
zedGraph.Invalidate();

```

Әдіс жұмысының нәтижесі 3.7-суретте көрсетілген.



3.7 Сурет – Шеңбер түріндегі диаграмма мысалы

Баған түріндегі диаграмманы салу үшін Столбчатая диаграмма батырмасын басу керек. Экранда диаграмма шығады. Коды келесі түрде болады:

Ең алдымен график алдыңғы нәтижелерден тазалану керек

```

zedGraph.GraphPane.CurveList.Clear();
zedGraph.GraphPane.GraphObjList.Clear();

```

Сосын жаңа графикті құрамыз:

```
GraphPane myPane = zedGraph.GraphPane;  
X, Y осі бойынша график тақырыбын енгіземіз.
```

```
myPane.Title.Text = "Төлем графигі";  
myPane.XAxis.Title.Text = "Төлеу";  
myPane.YAxis.Title.Text = "";
```

«Несие ақшасы» кестесіне жазамыз:

```
BarItem myBar = myPane.AddBar("Несие ақшасы", null, _payments.Select(t  
=> t.Dolg).ToArray(),  
Color.Red);
```

Бұл әдіс кірісінде сандар мәнінің массивін қабылдайды, сол үшін төлем топтамасын массивке түрлендіріп тастау керек. Ол үшін LINQ қолдануға болады және топтамадан бізге керекті массивті алуға болады:

```
_payments.Select(t => t.Dolg).ToArray()
```

Осы жағдайда төлем топтамасынан біз тек қарызды таңдаймыз және алынған құрылымды массивке түрлендіреміз.

Ары қарай, алынған бағанға келесі мәндері бар мәтіндік жазбаны қосамыз:

```
for (int i = 0; i < myBar.Points.Count; i++)  
{  
    TextObj barLabel = new TextObj(myBar.Points[i].Y.ToString(),  
myBar.Points[i].X, myBar.Points[i].Y + 20);  
    barLabel.FontSpec.Border.IsVisible = false;  
    myPane.GraphObjList.Add(barLabel);  
}
```

Алынған бағандарды толтырамыз:

```
myBar.Bar.Fill = new Fill(Color.Red, Color.White, Color.Red);
```

Пайыз мәндерін қамтитын бағандардың екінші жиынтығын құрастырамыз:

```
myBar = myPane.AddBar("Пайыздар", null, _payments.Select(t =>  
t.Percent).ToArray(), Color.Blue);
```

Сәйкесінше мәндерді қосамыз:

```
for (int i = 0; i < myBar.Points.Count; i++)
{
    TextObj barLabel = new TextObj(myBar.Points[i].Y.ToString(),
myBar.Points[i].X, myBar.Points[i].Y + 20);
    barLabel.FontSpec.Border.IsVisible = false;
    myPane.GraphObjList.Add(barLabel);
}
```

Бояймыз:

```
myBar.Bar.Fill = new Fill(Color.Blue, Color.White,Color.Blue)
```

Осьтердің сыртқы түрін және оларға жазылған жазбаларды құрастыруға кірісеміз. Көлденең осьті көрсетеміз:

```
myPane.XAxis.IsVisible = true;
```

Төлем нөмірлері бар жазбаны қосамыз:

```
myPane.XAxis.Scale.TextLabels =
_payments.Select(t=>t.Num.ToString()).ToArray();
```

Ось түрін көрсетеміз:

```
myPane.XAxis.Type = AxisType.Text;
```

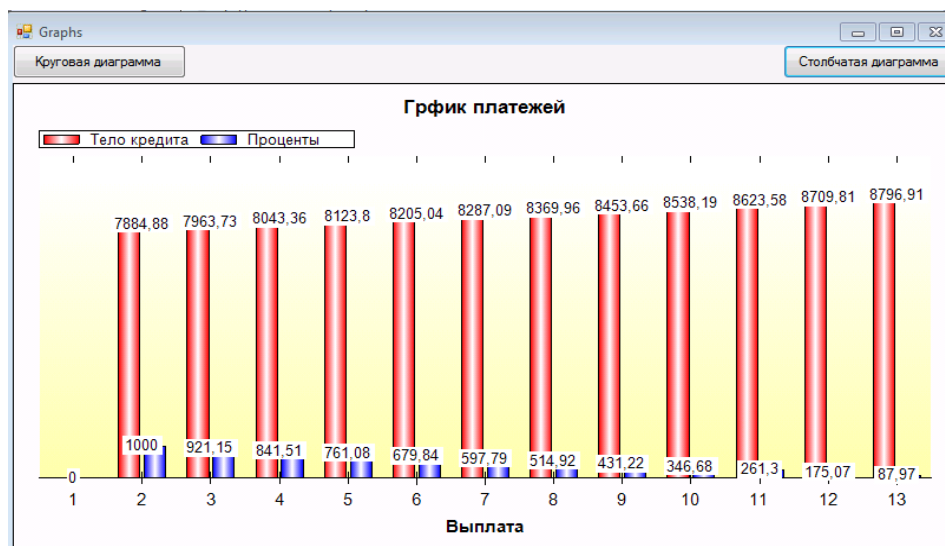
Түске бояймыз:

```
myPane.Chart.Fill = new Fill(Color.White, Color.FromArgb(255, 255,
166), 90F);
myPane.Fill = new Fill(Color.FromArgb(250, 250, 255));
```

Графикті қайта сызамыз:

```
zedGraph.AxisChange(); zedGraph.Invalidate();
```

Жұмыс нәтижесі 3.8-суретте көрсетілген.



3.8 Сурет– Төлемнің баған түріндегі диаграммасы

## ҚОРЫТЫНДЫ

Жасалған жұмыстар нәтижесінде несие калькуляторы қосымшасы жасалды. Қосымша аннуитетті және дифференциалды төлем түрлері бойынша төлем графигін құруға мүмкіндік береді.

Жасалынған несие калькуляторының бір артықшылығы – несиені уақытынан бұрын төлеуді есептеу мүмкіндігі. Формада қолданушы төлеу мерзімін көрсете отырып уақытынан бұрын төлеу сомасын жазатын бөлек жол қарастырылған. Қайта есептеу кезінде бұл сома төлемнің соңғы графигінде ескеріледі.

Нәтижені қарау үшін негізгі терезедегі кесте қолданылуы мүмкін. Сондай ақ, нәтижелер шеңбер түріндегі немесе баған түріндегі диаграммада көрсетілуі мүмкін. Экранда нәтиже кескіндемесі көрсетілумен қатар, қосымшада нәтижені Excel және Word форматтарындағы файлда сақтауға болады. Алынған нәтижелерді файлға сақтап, электрондық пошта арқылы жіберуге болады.

## ПАЙДАЛАНЫЛҒАН ӘДЕБИЕТТЕР ТІЗІМІ

- 1 Звонова, Е. А. Деньги, кредит, банки: Учебник/под ред. Е. А. Звоновой. - М: ИНФРА-М, 2015, с. 193-197
- 2 Мартыненко Н.Н. Деньги, кредит, банки: Учебник и практикум для академического бакалавриата/Под ред.: Мартыненко Н.Н., Соколов Ю.А./ Финансовый университет при Правительстве Российской Федерации (г. Москва),- М.: Издательство Юрайт, 2017, с. 181-187
- 3 Немцова Т.И., Голова С.Ю., Терентьев А.И.: Программирование на языке С++: Учебное пособие /Под ред. Гагариной Л.Г. - М.: ИД ФОРУМ: ИНФРА-М, 2012. - 512 с.: (ЭБС «Znanium»)
- 4 Гвоздева В.А. Базовые и прикладные информационные технологии: Учебник / В.А.Гвоздева – М.: ИД ФОРУМ, НИЦ ИНФРА-М, 2015. – 384 с. (ЭБС «Znanium»)
- 5 Зараменских Е.П. Основы бизнес-информатики: Учебник и практикум для бакалавриата и магистратуры / Е.П. Зараменских. – М.: Издательство Юрайт, 2017. – 407 с. – (Бакалавр и магистр. Академический курс). (ЭБС «biblio-online»)
- 6 Давыдова Н.А., Боровская Е.В. Программирование: Учебное пособие /. — 3-е изд. 241 с. — М.: БИНОМ. Лаборатория знаний, 2015
- 7 Программирование в алгоритмах [Электронный ресурс] / С.М. Окулов. - 5-е изд. (эл.). - М.: БИНОМ. Лаборатория знаний, 2014. – с. 383
- 8 Романова Ю.Д. [и др.]; Экономическая информатика: Учебник и практикум для бакалавриата и магистратуры / отв. ред. Романова Ю.Д. – М.: Издательство Юрайт, 2015. – 495 с. – (Бакалавр и магистр. Академический курс). (ЭБС «biblio-online»)
- 9 Официальный сайт «ВТБ» <https://www.vtb-bank.kz> (дата обращения: 12.03.2020)

## А Қосымшасы Бағдарламаның бастапқы коды

```
Файл Form1.cs
using System;
using System.Collections.Generic;
using System.Linq;
using System.Windows.Forms;
namespace bank_calc
{
    public partial class Form1 : Form
    {
        //int k = 160;
        bool flag;
        List<Payment> payments;
        public Form1()
        {
            InitializeComponent();
            Init();
        }
        private void Init()
        {
            payments = new List<Payment>();
            flag = true;
            dataGridView1.ColumnCount = 7;
            dataGridView1.Columns[0].Name = "№ п.п.";
            dataGridView1.Columns[1].Name = "Дата";
            dataGridView1.Columns[2].Name = "Сумма платежа";
            dataGridView1.Columns[3].Name = "Основной долг";
            dataGridView1.Columns[4].Name = "Начисленные %";
            dataGridView1.Columns[5].Name = "Остаток";
            dataGridView1.Columns[6].Name = "Досрочное";
            currencyCB.SelectedItem = "тенге";
            dateCB.SelectedItem = "месяц";
            typeCB.SelectedItem = "Аннуитетный";
        }
        private void calculate_Credit()
        {
            double V;
            double S;
            double Q;
            double Sn;
            double Qn;
```

## А қосымшасының жалғасы

```
double d;
double p;
double n;
int i;
int j;
//int np;
//double Sum;
//double SumV;
//double SumP;
if (flag)
{
    payments.Clear();
    n = float.Parse(timeTB.Text);
    if (dateCB.SelectedItem.ToString() == "год") n *= 12;
    for (int k = 0; k < (int)n; k++)
    {
        payments.Add(new Payment());
    }
    payments[0].Num = 1;
    payments[0].date = dateTimePicker1.Value;
    payments[0].Left = float.Parse(sumTB.Text);
}
else
{
    n = payments.Count - 1;
}
//Sum = 0;
//SumV = 0;
//SumP = 0;
if (typeCB.SelectedItem.ToString() == "Аннуитетный")
{
    S = float.Parse(sumTB.Text);
    Q = float.Parse(percentTB.Text) / 100 / 12;
    V = S * Q / (1 - Math.Exp(Math.Log(1 + Q) * (-n)));
    i = 1;
    j = 1;
    Sn = S;
    while (j < n)
    {
        Qn = Sn * Q;
        S = V - Qn;
        payments[i].Num = i + 1;
    }
}
```



## А қосымшасының жалғасы

```
payments[i].Sum = Math.Round(V, 2);
payments[i].Dolg = Math.Round(S, 2);
payments[i].Percent = Math.Round(Qn, 2);
Sn = Sn - S;
if (payments[i].Advance > 0)
{
    V = V - payments[i].Advance / (n - i - 1);
}
payments[i].Left = Math.Round(Sn, 2);

i++;
j++;
}
Qn = Sn * Q;
S = V - Qn;
Sn = 0;
if (flag) payments.Add(new Payment
{
    Num = i + 1,
    date = dateTimePicker1.Value.AddMonths(j),
    Sum = Math.Round(V, 2),
    Dolg = Math.Round(S, 2),
    Percent = Math.Round(Qn, 2),
    Left = Math.Round(Sn, 2)
});
else
{
    payments[i].Num = i + 1;
    payments[i].date = dateTimePicker1.Value.AddMonths(j);
    payments[i].Sum = Math.Round(V, 2);
    payments[i].Dolg = Math.Round(S, 2);
    payments[i].Percent = Math.Round(Qn, 2);
    payments[i].Left = Math.Round(Sn, 2);
}
}
//Дифференцированный
else if (typeCB.SelectedItem.ToString() == "Дифференцированный")
{
    S = float.Parse(sumTB.Text);
    n = float.Parse(timeTB.Text);
    if (dateCB.SelectedItem.ToString() == "год") n *= 12;
```

## А қосымшасының жалғасы

```
Q = float.Parse(percentTB.Text) / 100 / 12;
d = S / n;
i = 1;
j = 1;
Sn = S;
while (j < n)
{
    p = Sn * Q;
    V = d + p;
    payments[i].Num = i + 1;
    payments[i].date = dateTimePicker1.Value.AddMonths(j);
    payments[i].Sum = Math.Round(V, 2);
    payments[i].Dolg = Math.Round(d, 2);
    payments[i].Percent = Math.Round(p, 2);
    if (payments[i].Advance > 0)
    {
        d = (S - payments[i].Advance) / (n);
    }
    Sn = Sn - d;
    payments[i].Left = Math.Round(Sn, 2);
    i++;
    j++;
}
p = Sn * Q;
S = Sn - p;
V = d + p;
Sn = 0;
if (flag) payments.Add(new Payment
{
    Num = i + 1,
    date = dateTimePicker1.Value.AddMonths(j),
    Sum = Math.Round(V, 2),
    Dolg = Math.Round(d, 2),
    Percent = Math.Round(p, 2),
    Left = Math.Round(Sn, 2)
});
else
{
    payments[i].Num = i + 1;
    payments[i].date = dateTimePicker1.Value.AddMonths(j);
    payments[i].Sum = Math.Round(V, 2);
```

## А қосымшасының жалғасы

```
        payments[i].Dolg = Math.Round(d, 2);
        payments[i].Percent = Math.Round(p, 2);
        payments[i].Left = Math.Round(Sn, 2);
    }}

    flag = true;
    //dataGridView1.DataSource = payments;
    updateGridView();
}

private void updateGridView()
{
    dataGridView1.Rows.Clear();
    foreach (var item in payments)
    {
        string[] row = new string[] { item.Num.ToString(),
            item.date.ToString("d-m-y"),
            item.Sum.ToString(),
            item.Percent.ToString(),
            item.Dolg.ToString(),
            item.Left.ToString(),
            item.Advance.ToString()};
        dataGridView1.Rows.Add(row);
    }
}

private void button1_Click(object sender, EventArgs e)
{
    try{
        calculate_Credit();}
    catch (Exception ex){
        MessageBox.Show(ex.Message);
    }
}

//график
private void button4_Click(object sender, EventArgs e)
{
    if (payments.Count == 0) return;
    Graphs g = new Graphs(payments);
    g.Show();}

//добавить
```

## А қосымшасының жалғасы

```
private void button2_Click(object sender, EventArgs e)
{
    double val = float.Parse(textBox4.Text);
    DateTime dt = dateTimePicker2.Value;

    for (int j = 0; j < payments.Count - 1; j++)
    {
        if (dt >= payments[j].date && dt < payments[j + 1].date)
        {
            payments[j].Advance = val;
            flag = false;
            updateGridView();
            return;
        }
    }
}

private void button6_Click(object sender, EventArgs e)
{
    dataGridView1.ClipboardCopyMode =
DataGridViewClipboardCopyMode.EnableAlwaysIncludeHeaderText;
    dataGridView1.MultiSelect = true;
    dataGridView1.SelectAll();
    DataObject dataObj = dataGridView1.GetClipboardContent();
    if (dataObj != null)
        Clipboard.SetDataObject(dataObj);
    Microsoft.Office.Interop.Excel.Application xlexcel;
    Microsoft.Office.Interop.Excel.Workbook xlWorkBook;
    Microsoft.Office.Interop.Excel.Worksheet xlWorkSheet;
    object misValue = System.Reflection.Missing.Value;
    xlexcel = new Microsoft.Office.Interop.Excel.Application();
    xlexcel.Visible = true;
    xlWorkBook = xlexcel.Workbooks.Add(misValue);
    xlWorkSheet =
(Microsoft.Office.Interop.Excel.Worksheet)xlWorkBook.Worksheets.get_Item(1);
    Microsoft.Office.Interop.Excel.Range CR =
(Microsoft.Office.Interop.Excel.Range)xlWorkSheet.Cells[1, 1];
    CR.Select();
    xlWorkSheet.PasteSpecial(CR, Type.Missing, Type.Missing,
Type.Missing, Type.Missing, Type.Missing, true);
    for (int c = 0; c <= ColumnCount - 1; c++){
        tmp = tmp + dataArray[r, c] + "\t";
    }
}
```