Одеський національний політехнічний університет

Інститут комп’ютерних систем

Кафедра системного програмного забезпечення

Пояснювальна записка

до Розрахунково-графічної роботи

з дисципліни «Проектний практикум»

на тему «Клієнт-серверна програма розрахунку перспективного прибутку від інвестиційної діяльності»

Виконав: ст. гр. АС-123

Буслюк П.С.

Перевірив:

доц. Блажко О. А.

Одеса – 2016 року

**ЗМІСТ**

[1 ВИМОГИ ДО ПРОГРАМНОГО ПРОДУКТУ 3](#_Toc449537641)

[1.1 Визначення бізнес-вимог 3](#_Toc449537642)

[1.2 Визначення власних вимог 4](#_Toc449537643)

[1.3 Функціональні вимоги 9](#_Toc449537644)

[1.4 Нефункціональні вимоги 11](#_Toc449537645)

[1.5 Планування розробки 12](#_Toc449537646)

[2 ПРОЕКТУВАННЯ ПРОГРАМНОГО ПРОДУКТУ 12](#_Toc449537647)

[2.1 Концептуальне проектування 17](#_Toc449537648)

[2.2 Логічне проектування 18](#_Toc449537649)

[2.2.1 Структури даних 18](#_Toc449537650)

[2.2.2 Алгоритми 19](#_Toc449537651)

[2.2.3 Інтерфейс користувача 21](#_Toc449537652)

[3 КОНСТРУЮВАННЯ ПРОГРАМНОГО ПРОДУКТУ 21](#_Toc449537653)

[3.1 Опис програмних технологій 21](#_Toc449537654)

[3.2 Опис програмних бібліотек 23](#_Toc449537655)

[3.3 Особливості створення програмних модулів з урахуванням мови програмування 23](#_Toc449537656)

[3.4 Особливості створення структур даних 23](#_Toc449537657)

[3.5 Модульне тестування 24](#_Toc449537658)

[4 ТЕСТУВАННЯ ПРОГРАМНОГО ПРОДУКТУ 27](#_Toc449537659)

[4.1 Функціональне тестування 27](#_Toc449537660)

[5 РОЗГОРТАННЯ ПРОГРАМНОГО ПРОДУКТУ 29](#_Toc449537661)

[5.1 Інструкція з встановлення 29](#_Toc449537662)

[5.2 Інструкція з використання 29](#_Toc449537663)

# 1 ВИМОГИ ДО ПРОГРАМНОГО ПРОДУКТУ

## Визначення бізнес-вимог

Предметна область: Клієнт-серверна програма розрахунку перспективного прибутку від інвестиційної діяльності.

Суб’єкти автоматизації, як зацікавлені лиця: Клієнт-користувач (перспективний інвестор), який зацікавлений в тому, щоб не розраховувати вручну дані щодо інвестуємих компаній за певний період; Адміністратор – збирає данні щодо клієнтів (перспективних інвесторів), щоб в наступному можна було пропонувати цікаві пропозиції щодо інвестування в проекти.

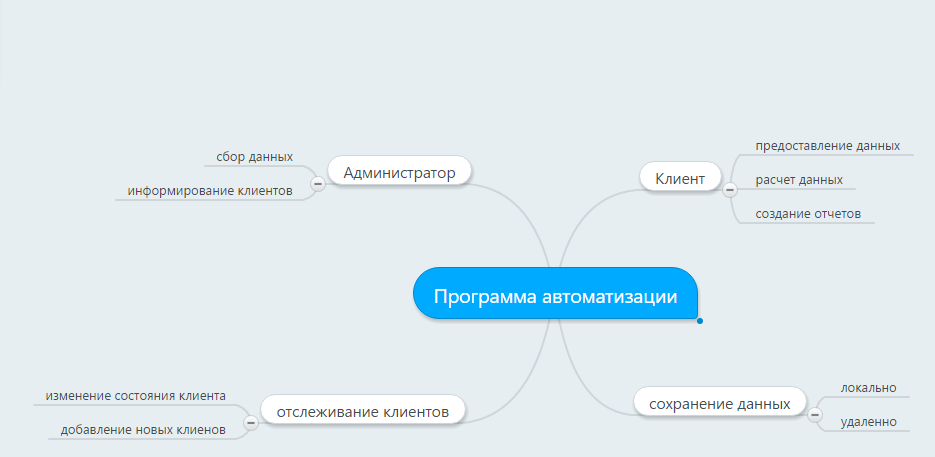


Рис 1 ­ - Метальна карта зв’язку між об’єктом та суб’єктами

Проблеми, що виникають: можлива відсутність необхідних функцій щодо розрахунку та обробки даних.

Аналіз існуючих програмних продуктів(список продуктів):

1) <http://invite-invest.ru/calculator-investitsii.html>

2) <http://soft.delovar.info/>

3) [http://www.denega.ru/portcalc162.zip](http://mmgp.ru/redirect/away.php?url=http%3A%2F%2Fwww.denega.ru%2Fportcalc162.zip)

4) <http://www.firmasoft.ru/hide/finans/clear.htm>

Таблиця рішень проблеми:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Продукти | Збереження даних віддалено | Розрахунок більше 1 компанії | Можливість довкладення/знімання сумми | Відновлення даних |
| 1 | - | - | + | - |
| 2 | - | - | + | - |
| 3 | - | - | - | - |
| 4 | - | - | - | - |

Ціль програми – довести до автоматизації процес розрахунку, збереження та відновлення розраховуємих даних, із подальшим отриманням новин щодо цікавих перспективних проектів, в які можна внести кошти для отримання додаткового прибутку.

Унікальна назва програми: Invest Profit v.2.0

## Визначення власних вимог

Рис 2. – Діаграма прецедентів

Типи акторів в системі:

Клієнт – кінцевий користувач.

Оператор ­­– кінцевий користувач.

soketIO – система взаємодії по протоколу TCP / IP.

H2 – БД, взаємодіюча по протоколу TCP / IP.

javaMail ­– бібліотека, взаємодіюча по протоколу TCP / IP.

Таблиця 1.1 – Сценарії використання

|  |  |  |
| --- | --- | --- |
| Варіант використання №1:  авторизація користувача | Передумова | К заповнив усі необхідні поля профілю |
|  | Післяумова | Сервер дав згоду на користування системою |
| Суб'єкт | Користувач(К), Сервер(С), Система Користувача (СК) |
| Основний потік | 1. К входить до програми та заповнює профіль. СК пропонує збереження даних профілю. 2. СК відправляє дані авторизації на С. 3. С шукає в БД дані К та відсилає відповідь. |
| Альтернативний потік | 1а.1 К не згоджується із збереженням даних. СК не зберігає дані. Вихід з программи.  1б.1 К вже заповнив дані. Перехід до 2 пункту.  2а.1 С не знайшла К в БД. С створює К в БД та надає доступ. |
| Варіант використання №2:  Збереження локальних даних локально | Передумова | К авторизований у СК; К вказав персональні дані. |
| Післяумова | Дані збережені |
| Суб'єкт | Користувач(К), Система Клієнта(СК) |
|  | Основний потік | 1. К обирає збереження даних. СК перевіряє дані та зберігає їх у XML файл |
| Альтернативний потік | 1а.1 Дані не коректні. СК видає повідомлення щодо невірних даних |
| Варіант використання №3:  Збереження даних віддалено | Передумова | К авторизований у СК; у К є доступ до Інтернет мережі; К вказав у персональні дані. |
| Післяумова | Обліковий запис збережений. |
| Суб'єкт | Користувач(К), Система Клієнта(СК), Сервер (С) |
| Основний потік | 1. К обирає збереження даних віддалено. СК перевіряє дані та відсилає їх на С. 2. С отримує їх та зберігає в БД |
| Альтернативний потік | 1а.1 К вказав невірні дані. СК пропонує ввести дані ще раз. |
| Варіант використання №4:Розрахунок даних | Передумова | К авторизований у СК; Дані заповнені; Дані вірні; |
|  | Післяумова | Дані розраховані |
| Суб'єкт | Користувач(К), Система Клієнта(СК) |
| Основний потік | 1. К обирає компанії для розрахунку. СК розраховує дані та видає результат. |
| Альтернативний потік |  |
| Варіант використання №5:  Надіслання повідомлення | Передумова | Є доступ до БД та мережі інтеренет. |
| Післяумова | Повідомлення відіслані вказаним адресатам. |
| Користувачі | Сервер(С), БД, Оператор(О), Клієнт(К) |
| Основний потік | 1. О обирає К для оповіщення та вводить саме повідомлення. С встановлює email адресу кожного К та надсилає повідомлення. |
| Альтернативний потік | 1а.1 Невірна адреса. С пропускає К. |

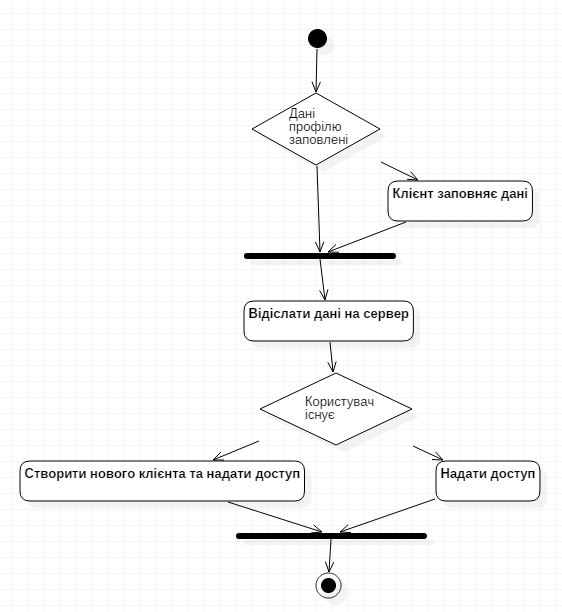


Рис 3 – Діаграма діяльності «Авторизація користувача»



Рис 4 – Діаграма діяльності «Збереження локальних даних»

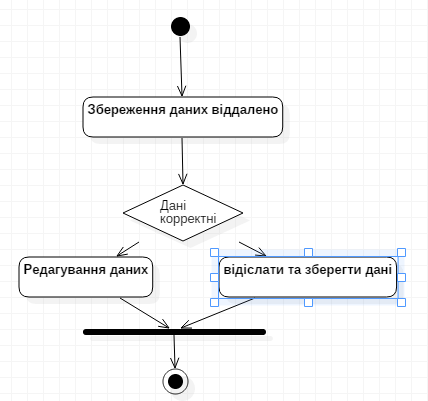


Рис 5 – Діаграма діяльності «Збереження даних віддалено»

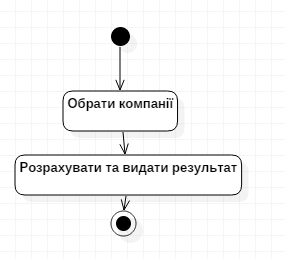


Рис 6 – Діаграма діяльності «Розрахунок даних»

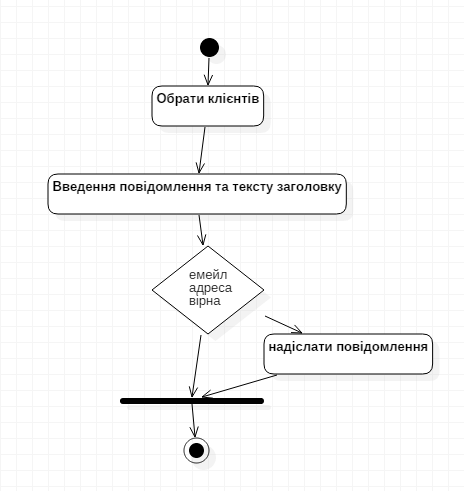


Рис 7 – Діаграма діяльності «Надіслати повідомлення»

## Функціональні вимоги

Тезаріус:

Архітектура клієнт-сервер - передбачає взаємодію та обмін даними між вузлами. Вона передбачає такі основні компоненти:

1. набір [серверів](https://uk.wikipedia.org/wiki/%D0%A1%D0%B5%D1%80%D0%B2%D0%B5%D1%80), які надають інформацію або інші послуги програмам, які звертаються до них;
2. набір [клієнтів](https://uk.wikipedia.org/wiki/%D0%9A%D0%BB%D1%96%D1%94%D0%BD%D1%82_(%D1%96%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0)), які використовують сервіси, що надаються серверами;
3. [мережа](https://uk.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D1%83%D0%BD%D1%96%D0%BA%D0%B0%D1%86%D1%96%D0%B9%D0%BD%D0%B0_%D0%BC%D0%B5%D1%80%D0%B5%D0%B6%D0%B0), яка забезпечує взаємодію між клієнтами та серверами.

Авторизація - керування рівнями та засобами [доступу](https://uk.wikipedia.org/w/index.php?title=%D0%94%D0%BE%D1%81%D1%82%D1%83%D0%BF&action=edit&redlink=1) до певного захищеного ресурсу, як в фізичному розумінні (доступ до кімнати готелю за карткою), так і в галузі цифрових технологій (наприклад, автоматизована система контролю доступу) та ресурсів системи залежно від [ідентифікатора](https://uk.wikipedia.org/wiki/%D0%86%D0%B4%D0%B5%D0%BD%D1%82%D0%B8%D1%84%D1%96%D0%BA%D0%B0%D1%82%D0%BE%D1%80) і [пароля](https://uk.wikipedia.org/wiki/%D0%9F%D0%B0%D1%80%D0%BE%D0%BB%D1%8C) користувача або надання певних повноважень (особі, програмі) на виконання деяких дій у системі обробки даних.

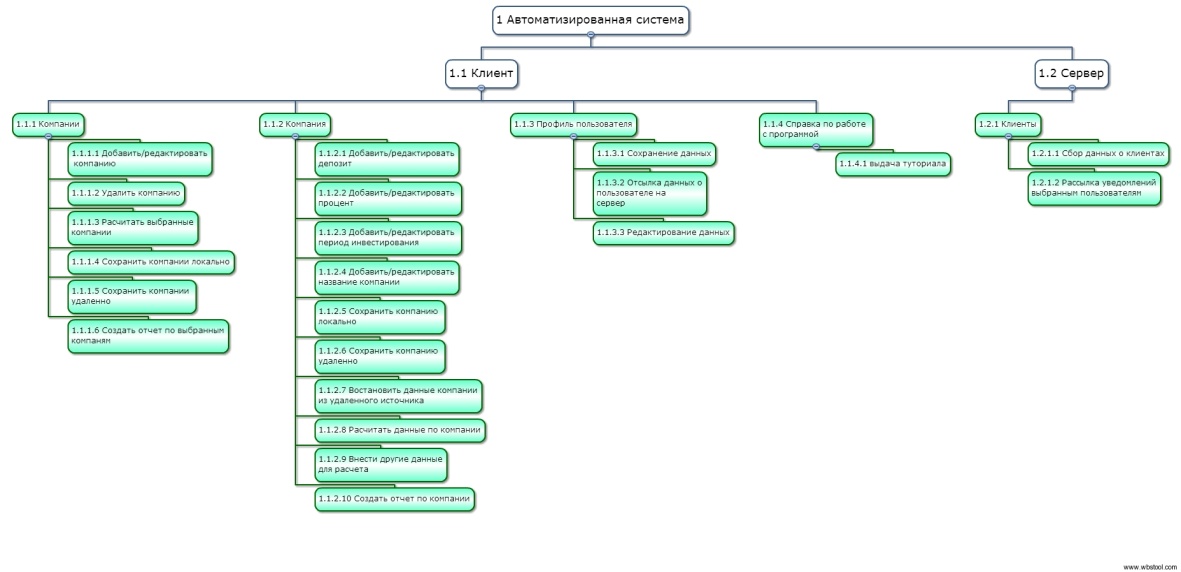
База даних ([англ.](https://uk.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D1%96%D0%B9%D1%81%D1%8C%D0%BA%D0%B0_%D0%BC%D0%BE%D0%B2%D0%B0) database) – це сукупність [даних](https://uk.wikipedia.org/wiki/%D0%94%D0%B0%D0%BD%D1%96_(%D0%BE%D0%B1%D1%87%D0%B8%D1%81%D0%BB%D1%8E%D0%B2%D0%B0%D0%BB%D1%8C%D0%BD%D0%B0_%D1%82%D0%B5%D1%85%D0%BD%D1%96%D0%BA%D0%B0)), організованих відповідно до концепції, яка описує характеристику цих даних і взаємозв'язки між їх елементами; ця сукупність підтримує щонайменше одну з областей застосування (за стандартом ISO/IEC 2382:2015). В загальному випадку база даних містить схеми, таблиці, подання, збережені процедури та інші об'єкти. Дані у базі організовують відповідно до моделі організації [даних](https://uk.wikipedia.org/wiki/%D0%94%D0%B0%D0%BD%D1%96). Таким чином, сучасна база даних, крім саме даних, містить їх опис та може містити засоби для їх обробки.

XML - розши́рювана мо́ва розмі́тки ([англ.](https://uk.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D1%96%D0%B9%D1%81%D1%8C%D0%BA%D0%B0_%D0%BC%D0%BE%D0%B2%D0%B0) Extensible Markup Language) — запропонований консорціумом World Wide Web ([W3C](https://uk.wikipedia.org/wiki/W3C)) стандарт побудови [мов розмітки](https://uk.wikipedia.org/wiki/%D0%9C%D0%BE%D0%B2%D0%B0_%D1%80%D0%BE%D0%B7%D0%BC%D1%96%D1%82%D0%BA%D0%B8_%D0%B4%D0%B0%D0%BD%D0%B8%D1%85) ієрархічно структурованих [даних](https://uk.wikipedia.org/wiki/%D0%94%D0%B0%D0%BD%D1%96) для обміну між різними [застосунками](https://uk.wikipedia.org/wiki/%D0%97%D0%B0%D1%81%D1%82%D0%BE%D1%81%D1%83%D0%BD%D0%BE%D0%BA), зокрема, через [Інтернет](https://uk.wikipedia.org/wiki/%D0%86%D0%BD%D1%82%D0%B5%D1%80%D0%BD%D0%B5%D1%82).[[1]](https://uk.wikipedia.org/wiki/XML#cite_note-XmlOriginsGoals-1) Є спрощеною підмножиною мови розмітки [SGML](https://uk.wikipedia.org/wiki/SGML). XML-документ складається із текстових знаків, і придатний до читання людиною.

Інтерне́т (від [англ.](https://uk.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D1%96%D0%B9%D1%81%D1%8C%D0%BA%D0%B0_%D0%BC%D0%BE%D0%B2%D0%B0) Internet), міжмере́жжя — всесвітня система взаємосполучених [комп'ютерних мереж](https://uk.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D1%83%D0%BD%D1%96%D0%BA%D0%B0%D1%86%D1%96%D0%B9%D0%BD%D0%B0_%D0%BC%D0%B5%D1%80%D0%B5%D0%B6%D0%B0), що базуються на [комплекті Інтернет-протоколів](https://uk.wikipedia.org/wiki/TCP/IP). Інтернет також називають мережею мереж. Інтернет складається з мільйонів локальних і глобальних приватних, публічних, академічних, ділових і урядових мереж, пов'язаних між собою з використанням різноманітних дротових,[оптичних](https://uk.wikipedia.org/wiki/%D0%9E%D0%BF%D1%82%D0%BE%D0%B2%D0%BE%D0%BB%D0%BE%D0%BA%D0%BD%D0%BE) і [бездротових технологій](https://uk.wikipedia.org/wiki/%D0%91%D0%B5%D0%B7%D0%B4%D1%80%D0%BE%D1%82%D0%BE%D0%B2%D0%B0_%D0%BC%D0%B5%D1%80%D0%B5%D0%B6%D0%B0). Інтернет становить фізичну основу для розміщення величезної кількості інформаційних ресурсів і послуг, таких як взаємопов'язані гіпертекстові документи [Всесвітньої павутини](https://uk.wikipedia.org/wiki/%D0%92%D1%81%D0%B5%D1%81%D0%B2%D1%96%D1%82%D0%BD%D1%8F_%D0%BF%D0%B0%D0%B2%D1%83%D1%82%D0%B8%D0%BD%D0%B0) (World Wide Web — WWW) та [електронна пошта](https://uk.wikipedia.org/wiki/%D0%95%D0%BB%D0%B5%D0%BA%D1%82%D1%80%D0%BE%D0%BD%D0%BD%D0%B0_%D0%BF%D0%BE%D1%88%D1%82%D0%B0).

Електро́нна по́шта ([англ.](https://uk.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D1%96%D0%B9%D1%81%D1%8C%D0%BA%D0%B0_%D0%BC%D0%BE%D0%B2%D0%B0) e-mail, або email, скорочення від electronic mail) — популярний [сервіс](https://uk.wikipedia.org/wiki/%D0%95%D0%BB%D0%B5%D0%BA%D1%82%D1%80%D0%BE%D0%BD%D0%BD%D0%B8%D0%B9_%D1%81%D0%B5%D1%80%D0%B2%D1%96%D1%81) в [інтернеті](https://uk.wikipedia.org/wiki/%D0%86%D0%BD%D1%82%D0%B5%D1%80%D0%BD%D0%B5%D1%82), що робить можливим обмін даними будь-якого змісту ([текстові документи](https://uk.wikipedia.org/wiki/%D0%A2%D0%B5%D0%BA%D1%81%D1%82%D0%BE%D0%B2%D0%B8%D0%B9_%D1%84%D0%B0%D0%B9%D0%BB), [аудіо](https://uk.wikipedia.org/wiki/%D0%90%D1%83%D0%B4%D1%96%D0%BE%D1%84%D0%B0%D0%B9%D0%BB)-, [відеофайли](https://uk.wikipedia.org/w/index.php?title=%D0%92%D1%96%D0%B4%D0%B5%D0%BE%D1%84%D0%B0%D0%B9%D0%BB&action=edit&redlink=1), [архіви](https://uk.wikipedia.org/wiki/%D0%90%D1%80%D1%85%D1%96%D0%B2_(%D1%96%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0)), [програми](https://uk.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%27%D1%8E%D1%82%D0%B5%D1%80%D0%BD%D0%B0_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%B0)).

Багаторівнева ієрархія функціональних вимог:



Виключення функціональних вимог-дублікатів:

Можна виключити такі вимоги, адже вони зберігаються у вимогах Компаній:

1.1.2.5 – Зберегти компанію локально , 1.1.2.6-– Зберегти компанію віддалено, 1.1.2.7 – Відновити данні з віддаленого джерела.

Визначення критеріїв за типом MoSCoW: Усі вимоги мають тип M, однак можна додати ще пару вимог типу W – додати оновлення програми та ввести оплату за ПЗ.

Функціональний аналіз існуючих продуктів:

1) <http://invite-invest.ru/calculator-investitsii.html>

2) <http://soft.delovar.info/>

3) [http://www.denega.ru/portcalc162.zip](http://mmgp.ru/redirect/away.php?url=http%3A%2F%2Fwww.denega.ru%2Fportcalc162.zip)

4) <http://www.firmasoft.ru/hide/finans/clear.htm>

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Продукти | Збереження даних віддалено | Розрахунок більше 1 компанії | Створити звіт | Відновлення даних | Збір даних про користувачів |
| 1 | - | - | - | - | - |
| 2 | - | - | + | - | - |
| 3 | - | - | - | - | - |
| 4 | - | - | - | - | - |

## Нефункціональні вимоги

Зовнішні інтерфейси: Інтерфейс повинен бути інтуїтивно зрозумілим, із однаковим оформленням на усіх формах. Також повинні використовуватись компоненти меню та таблиць для надання повноцінної інформації, розбитої на певні блоки із мінімальною затратою місця на формах для кращого зовнішнього вигляду. Тож, будуть використовуватись наступні стилі оформлення: шрифт – Segoe UI, назви кнопок - російською мовою, задній фон: сірий (RGB(240,240,240)).

За управління будуть відповідати наступні компоненти: стандартні елементи swing. Також налаштовані гарячі клавіші: F1 – довідка,Ctrl+R- Формування звіту , Ctrl+A – вибір усіх компаній,Alt+C – розрахунок даних обраних компаній, Ctrl+N – створення нової компанії,Ctrl+D – видалення обраної компанії,Ctrl+Alt+S – збереження компаній віддалено, Ctrl+Alt+R – відновлення даних з серверу, Alt+N – редагування назви компанії, Alt+D– редагування депозиту, Alt+P – редагування відсотку, Alt+I – редагування тривалості інвестування. Відображення повідомлень буде відбуватись в стандартних MessageBox ОС.

Програмні інтерфейси: робота в OS: Windows - XP SP1 та вище.

Обов’язкова наявність JDK version 8u77.

Інтерфейси передачі даних між клієнтом та сервером: Wifi або Ethernet.

Продуктивність:

максимальний час відгуку програми на запит - до 2 секунд;

максимальна кількість обслуговуємих клієнтів одночасно – 2000;

Надійність:

мінімальний час безвідмовної роботи – 10 хвилин;

також передбачено відновлення даних після збоїв у системі.

Технології розробки:

ОС: Windows 7, Windows 10

СУБД: H2

Програмні технологии: Desktop

Структури даних: xml, files, db

Інструменти розробки:

IDE: NetBeans

Збирач програмних модулів: JDK version 8u77.

VCS (система контролю версій): Git

Бібліотеки модульного тестування: jUnit4

Автоматизоване тестування: Jubula

## 1.5 Планування розробки

Структура декомпозицій робіт (WBS, дані сформовані за допомогою сайту app.ganttpro.com. Зсилання на проект: https://app.ganttpro.com/shared/token/2df8ca27c089b7f96b54ed7707240acc5743b7b4ebbc8e78aaf972be6d0a927b):

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Структура задач | Наименование задач | Дата начала | Время начала | Дата окончания | Время окончания | Задачи, предшествующие данной | Прогресс | Длительность (часы) |
|  | Общая продолжительность | 04.01.2016 | 08:00 | 25.04.2016 | 15:00 |  |  | 807 |
| 1 | Имя проекта: Invest Profit | 04.01.2016 | 08:00 | 25.04.2016 | 15:00 |  | 84.22 % | 807 |
| 1.1 | Создание архитектуры | 04.01.2016 | 08:00 | 27.01.2016 | 13:00 |  | 100.00 % | 175 |
| 1.1.1 | Определение функциональных требований | 04.01.2016 | 08:00 | 13.01.2016 | 18:00 |  | 100.00 % | 80 |
| 1.1.2 | Определение структуры данных | 18.01.2016 | 08:00 | 21.01.2016 | 18:00 |  | 100.00 % | 40 |
| 1.1.3 | Определение взаимосвязей между элементами | 22.01.2016 | 08:00 | 27.01.2016 | 13:00 |  | 100.00 % | 35 |
| 1.2 | Создание прототипа | 28.01.2016 | 17:34 | 08.02.2016 | 17:00 |  | 100.00 % | 70 |
| 1.2.1 | Создание форм | 28.01.2016 | 17:34 | 02.02.2016 | 17:34 |  | 100.00 % | 30 |
| 1.2.2 | Настройка связи к БД | 02.02.2016 | 17:00 | 04.02.2016 | 17:00 |  | 100.00 % | 20 |
| 1.2.3 | Настройка взаимосвязей между элементами | 05.02.2016 | 17:00 | 08.02.2016 | 17:00 |  | 100.00 % | 10 |
| 1.3 | Внедрение основного функционала | 08.02.2016 | 17:34 | 06.04.2016 | 13:00 |  | 100.00 % | 416 |
| 1.3.1 | Создание методов расчета данных | 24.02.2016 | 12:00 | 29.02.2016 | 12:00 |  | 100.00 % | 30 |
| 1.3.2 | Настройка клиент-серверной архитектуры | 08.02.2016 | 17:34 | 22.02.2016 | 17:34 |  | 100.00 % | 100 |
| 1.3.3 | Сохранение данных локально | 29.02.2016 | 12:00 | 02.03.2016 | 17:00 |  | 100.00 % | 25 |
| 1.3.4 | Сохранение данных удаленно | 02.03.2016 | 12:00 | 08.03.2016 | 17:00 |  | 100.00 % | 45 |
| 1.3.5 | Восстановление данных из удаленного источника | 08.03.2016 | 12:00 | 14.03.2016 | 17:00 |  | 100.00 % | 45 |
| 1.3.6 | Создание отчетов | 14.03.2016 | 12:00 | 15.03.2016 | 17:00 |  | 100.00 % | 15 |
| 1.3.7 | Рефакторинг кода | 15.03.2016 | 12:00 | 24.03.2016 | 12:00 |  | 100.00 % | 70 |
| 1.3.8 | Создание профиля клиента | 24.03.2016 | 13:00 | 25.03.2016 | 13:00 |  | 100.00 % | 10 |
| 1.3.9 | Передача данных профиля клиента на сервер | 25.03.2016 | 13:00 | 31.03.2016 | 18:00 |  | 100.00 % | 45 |
| 1.3.10 | Настройка уведомлений на почту для выбранных клиентов | 30.03.2016 | 13:00 | 06.04.2016 | 13:00 |  | 100.00 % | 50 |
| 1.4 | Тестирование | 06.04.2016 | 13:00 | 25.04.2016 | 15:00 | 1.3 | 5.19 % | 132 |
| 1.4.1 | Модульное тестирование | 06.04.2016 | 13:00 | 13.04.2016 | 13:00 |  | 14.00 % | 50 |
| 1.4.2 | Автоматизированное тестирование | 13.04.2016 | 13:00 | 20.04.2016 | 13:00 |  | 0.00 % | 50 |
| 1.4.3 | Устранение неисправностей | 20.04.2016 | 10:00 | 25.04.2016 | 15:00 |  | 0.00 % | 35 |
| 1.5 | Написание тех документации | 25.04.2016 | 15:00 | 25.04.2016 | 15:00 | 1.4 | 0.00 % | 0 |

UCP метод оцінки об’єму виконуємих робіт:

Таблиця 1.1 – Визначення типів акторів

|  |  |  |  |
| --- | --- | --- | --- |
| Тип актора | Вага | Примітка | Актори |
| Простий | 1 | Представляє іншу систему, взаємодіючу по зумовленій API (REST, SOAP, dll) | 1 |
| Середній | 2 | Представляє іншу систему, що взаємодіє по протоколу типу TCP / IP | 3 |
| Складний | 3 | Кінцевий користувач, що взаємодіє через GUI або Webpage | 2 |
| ∑ UAW = | | | 13 |

Таблиця 1.2 – Визначення типів варіантів використання за кількістю транзакцій

|  |  |  |  |
| --- | --- | --- | --- |
| Тип варіанту використання | Вага | Кількість транзакцій | Кількість варіантів використання |
| Простий | 5 | До 4 | 2 |
| Середній | 10 | Від 4 до 7 | 2 |
| Складний | 15 | Від 7 | 1 |
| Розрахунок UnadjustedUseCaseWaiting (UUCW) | | | 45 |

UCP = UAW + UUCW = 13+ 45 = 58

Таблиця 1.3 - Технічні фактори

| Фактор | Опис | Вага | Пояснення | Оцінка |
| --- | --- | --- | --- | --- |
| T1 | Розподіленість системи | 2 | Інформує про потребу системи в розподілених обчисленнях | 1 |
| T2 | Час відгуку | 1 | Визначає ефективність системи з точки зору часу відгуку, потоку робіт тощо | 4 |
| T3 | Ефективність кінцевого користувача | 1 | Визначає ефективність користувача з точки зору його (її) сприйняття | 5 |

Продовження таблиця 1.3 - Технічні фактори

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Фактор | Опис | Вага | Пояснення | Оцінка |
| T4 | Складність обробки | 1 | Визначає, чи будуть застосовуватись складні алгоритми для обробки даних | 5 |
| T5 | Фокус на повторному використанні коду | 1 | Визначає, чи будуть елементи коду системи використовуватись знову | 2 |
| T6 | Простота інсталяції | 0,5 | Визначає метод інсталяції та простоту інсталяції для кінцевого користувача, чи буде потреба в спеціалісті для інсталяції системи | 3 |
| T7 | Простота використання | 0,5 | Визначає узгодженість інтерфейсу користувача з його потребами | 4 |
| T8 | Портативність | 2 | Визначає, чи має застосування працювати в різних середовищах | 3 |
| T9 | Простота змін | 1 | Визначає, чи будуватиметься система в такий спосіб, щоб спростити її модифікації в майбутньому | 3 |
| T10 | Паралельні обчислення | 1 | Інформує, чи будуть мати в системі місце паралельні обчислення | 0 |
| T11 | Засоби захисту | 1 | Визначає чи потребуватиме система спеціальні засоби захисту даних чи системи | 2 |
| T12 | Доступ до третьої сторони | 1 | Визначає ступень використання системи зовнішніми системами або акторами | 2 |
| T13 | Потреби в спеціальному навчанні | 1 | Визначає, чи потрібно організувати тренінги для користувачів | 1 |
| TechnicalComplexityFactor (TCF) | | | | 0,955 |

TCF = 0.6+(0.01\*SUM(Ki\*Vi)) = 0,955

Таблиця 1.4 – Оцінка зовнішніх факторів

|  |  |  |  |
| --- | --- | --- | --- |
| Фактор | Описание | Коэф-т(Ki) | Значение (Vi) |
| T1 | Familiarity With The Project Process | 1,5 | 4 |
| T2 | ApplicationExperience | 0,5 | 5 |
| T3 | OO ProgrammingExperience | 1 | 5 |
| T4 | LeadAnalystCapability | 0,5 | 4 |
| T5 | Motivation | 1 | 5 |
| T6 | StableRequirements | 2 | 5 |
| T7 | PartTimeStaff | -1 | 3 |
| T8 | DifficultProgrammingLanguage | -1 | 2 |
|  | EnvironmentFactor (EF) |  | 0,635 |

EF = 1.4+(-0.03\*SUM(Ki\*Vi)) = 0,635

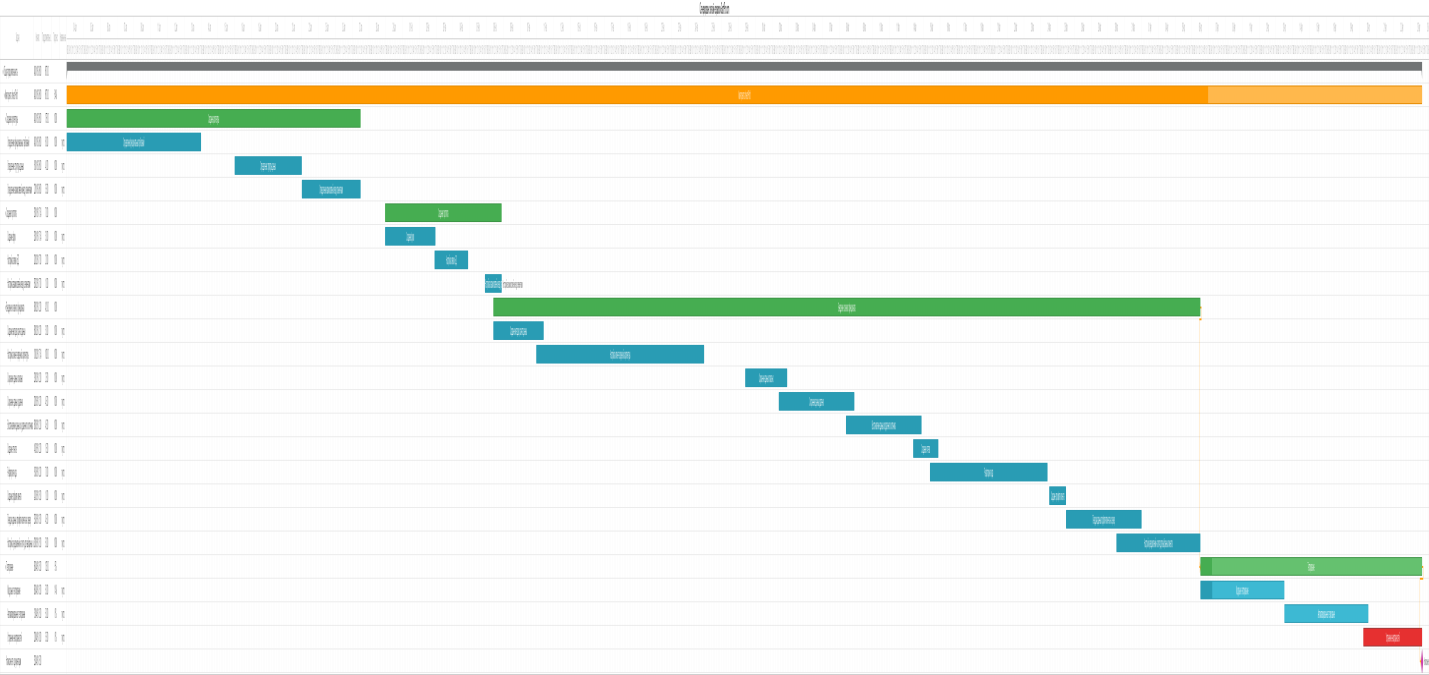
Скоректовані UCP обчислюються за формулою

AUCP = UCP \* TCF \* EF = 58 \* 0,955 \* 0,635 = 35,17

Більшість значень T1-T8 перевищують 3, тому одному UCP відповідає 35 робочих годин.

Hours of Effort per Use Case Point = 35 \* 41,2 = 1442 години.

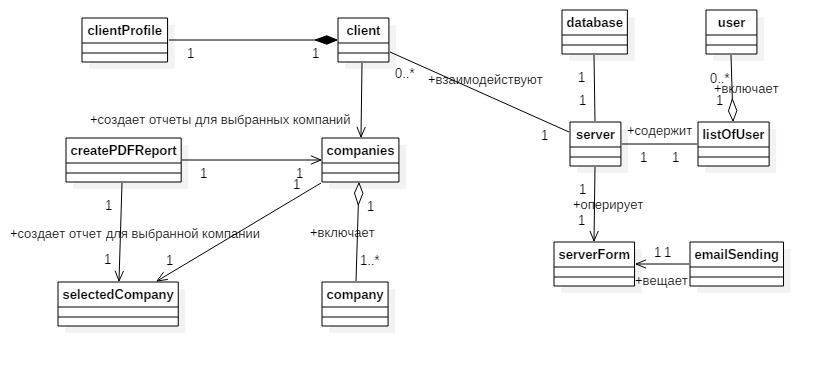
Діаграма Ганта:



# 2 ПРОЕКТУВАННЯ ПРОГРАМНОГО ПРОДУКТУ

## 2.1 Концептуальне проектування

Діаграма концептуальних класів:



## 2.2 Логічне проектування

Діаграма програмних класів:

## C:\Diplom\report\диаграма програмных класов.jpg

### 2.2.1 Структури даних

Структура даних по компаніям (для розрахунку даних) повинна бути така:

<companies>

<company>

<name>Назва компанії</name>

<depo>Початковий депозит</depo>

<persent>Середній відсоток інвестування</persent>

<period>Період інвестування</period>

<intervals>

<i>

<plus>Сумма довкладень за i-й період</plus>

<minus>Сумма знімань коштів за i-й період </minus>

<reinvesting>Перенесення поточної сумми у наступний період інвестування</reinvesting>

</i>

</intervals>

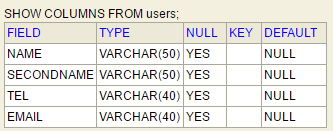
</company>

</companies>

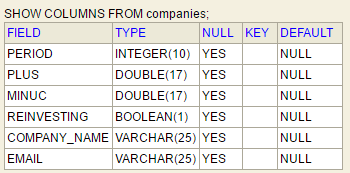
Структура даних в БД:

Усі дані у нижчезазначених таблицях описані у xml-тегах, які знаходяться вище.

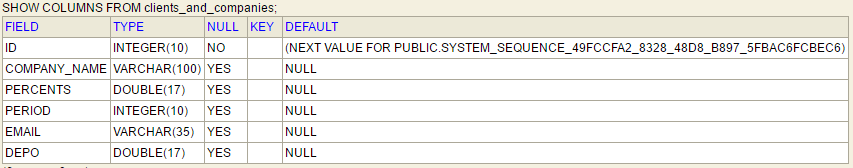
Таблиця користувачів системи:



Таблиця компаній користувача:



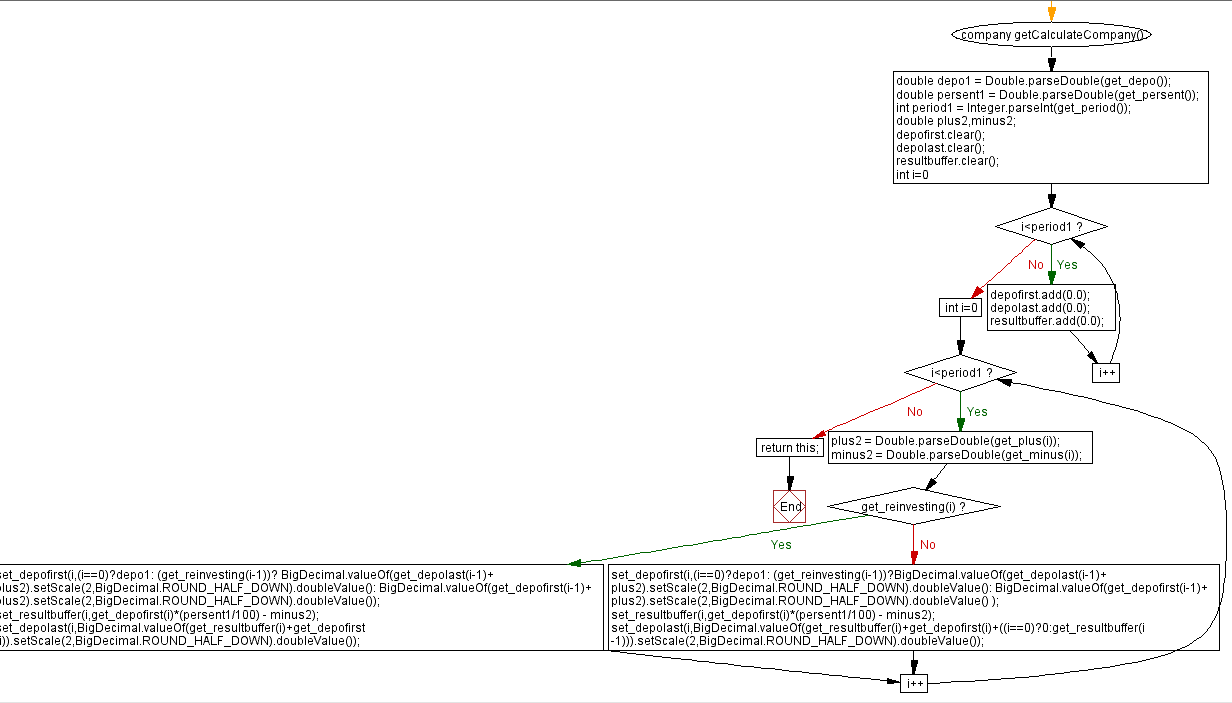
Таблиця доповнень для окремих компаній:



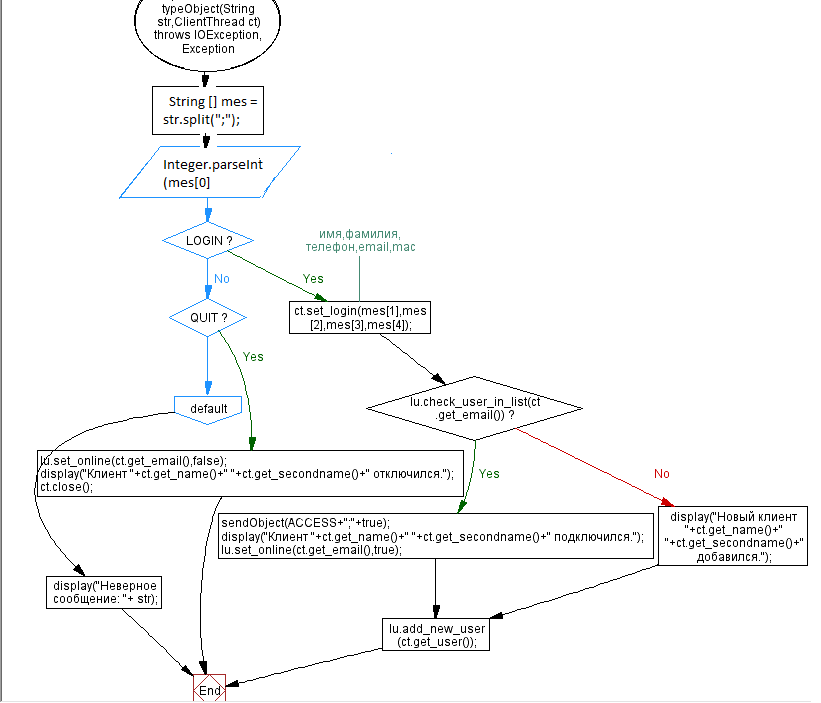
### 2.2.2 Алгоритми

Блок-схема 2 важливих для проекту функцій:

1 Блок-схема алгоритму розрахунку даних (блок-схема автоматично створена в програмі на основі коду програми, надалі блок-схема буде перерисована):



2 Блок-схема обміну даних між клієнтом та сервером. Серверна частина. (блок-схема автоматично створена в програмі, надалі блок-схема буде перерисована):



Оцінка може бути оцінена як часом (часова складність) так і об’ємом памяті (ємкісна складність).

Оцінка складності кожного елементарного блоку:

Оператори введення/виведення , проста умова - О (1)

Складна умова - O (m), m - кількість полів.

Оцінка алгоритмічної складності циклів:

Лінійний час - О (n), де n – кількість операторів .

Загальна оцінка алгоритмічної складності :

По першому алгоритму буде квадратична складність, так як в функції присутній один цикл, по якому розраховуються певні оператори, після виконання необхідних умов.

По другому – така сама складність і з тим же описом.

Оцінка алгоритмічної складності доступу до даних:

В даних алгоритмах не використовуються зчитування даних із xml-файлів та БД, бо це зроблено раніше і дані вже завантажені в оперативну пам'ять комп’ютера.

### 2.2.3 Інтерфейс користувача

Зовнішні інтерфейси: Інтерфейс інтуїтивно зрозумілим, із однаковим оформленням на усіх формах. Також використовувались компоненти меню та таблиць для надання повноцінної інформації, розбитої на певні блоки із мінімальною затратою місця на формах для кращого зовнішнього вигляду. Тож, будуть використовуватись наступні стилі оформлення: шрифт – Segoe UI, назви кнопок - російською мовою, задній фон: сірий (RGB(240,240,240)).

За управління відповідать наступні компоненти: стандартні елементи swing. Також налаштовані гарячі клавіші: F1 – довідка,Ctrl+R- Формування звіту , Ctrl+A – вибір усіх компаній,Alt+C – розрахунок даних обраних компаній, Ctrl+N – створення нової компанії,Ctrl+D – видалення обраної компанії,Ctrl+Alt+S – збереження компаній віддалено, Ctrl+Alt+R – відновлення даних з серверу, Alt+N – редагування назви компанії, Alt+D– редагування депозиту, Alt+P – редагування відсотку, Alt+I – редагування тривалості інвестування. Відображення повідомлень відбувається в стандартних MessageBox ОС.

Також був застосований золотий перетин – це співвідношення, найвідповідніше для

[естетичного](https://uk.wikipedia.org/wiki/%D0%95%D1%81%D1%82%D0%B5%D1%82%D0%B8%D0%BA%D0%B0) сприйняття зображення, вперше запропоноване давньогрецьким математиком [Евклідом](https://uk.wikipedia.org/wiki/%D0%95%D0%B2%D0%BA%D0%BB%D1%96%D0%B4).

Програмні інтерфейси: робота в OS: Windows - XP SP1 та вище.

Інтерфейси передачі даних між клієнтом та сервером: Wifi або Ethernet.

# 3 КОНСТРУЮВАННЯ ПРОГРАМНОГО ПРОДУКТУ

## 3.1 Опис програмних технологій

NetBeans IDE — [вільне](https://uk.wikipedia.org/wiki/%D0%92%D1%96%D0%BB%D1%8C%D0%BD%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BD%D0%B5_%D0%B7%D0%B0%D0%B1%D0%B5%D0%B7%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%BD%D1%8F) [інтегроване середовище розробки](https://uk.wikipedia.org/wiki/%D0%86%D0%BD%D1%82%D0%B5%D0%B3%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B5_%D1%81%D0%B5%D1%80%D0%B5%D0%B4%D0%BE%D0%B2%D0%B8%D1%89%D0%B5_%D1%80%D0%BE%D0%B7%D1%80%D0%BE%D0%B1%D0%BA%D0%B8) (IDE) для [мов програмування](https://uk.wikipedia.org/wiki/%D0%9C%D0%BE%D0%B2%D0%B0_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F) [Java](https://uk.wikipedia.org/wiki/Java), [JavaFX](https://uk.wikipedia.org/wiki/JavaFX), [C](https://uk.wikipedia.org/wiki/%D0%A1_(%D0%BC%D0%BE%D0%B2%D0%B0_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F))/[C++](https://uk.wikipedia.org/wiki/C%2B%2B), [PHP](https://uk.wikipedia.org/wiki/PHP), [JavaScript](https://uk.wikipedia.org/wiki/JavaScript),[HTML5](https://uk.wikipedia.org/wiki/HTML5), [Python](https://uk.wikipedia.org/wiki/Python), [Groovy](https://uk.wikipedia.org/wiki/Groovy). Середовище може бути встановлене і для підтримки окремих мов, і у повній конфігурації. Середовище розробки NetBeans за замовчуванням підтримує розробку для платформ [J2SE](https://uk.wikipedia.org/w/index.php?title=J2SE&action=edit&redlink=1) і [J2EE](https://uk.wikipedia.org/wiki/J2EE).

Поширюється у open source code під ліцензіями [GPLv2](https://uk.wikipedia.org/wiki/GPLv2) і [CDDL](https://uk.wikipedia.org/w/index.php?title=CDDL&action=edit&redlink=1). Проект NetBeans IDE підтримувався і спонсорувався фірмою [Sun Microsystems](https://uk.wikipedia.org/wiki/Sun_Microsystems) і після придбання Sun — [Oracle](https://uk.wikipedia.org/wiki/Oracle_Corporation), проте розробка NetBeans ведеться незалежно співтовариством розробників (NetBeans Community) і компанією NetBeans.Org.

NetBeans IDE доступна для платформ [Microsoft Windows](https://uk.wikipedia.org/wiki/Microsoft_Windows), [GNU](https://uk.wikipedia.org/wiki/GNU)/[Linux](https://uk.wikipedia.org/wiki/Linux), [FreeBSD](https://uk.wikipedia.org/wiki/FreeBSD), і [Solaris](https://uk.wikipedia.org/wiki/Solaris_(%D0%BE%D0%BF%D0%B5%D1%80%D0%B0%D1%86%D1%96%D0%B9%D0%BD%D0%B0_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0)) (як [SPARC](https://uk.wikipedia.org/wiki/SPARC), так [x86](https://uk.wikipedia.org/wiki/X86)). Для інших платформ доступна можливість зібрати NetBeans самостійно із open source code.

За якістю і можливостям останні версії NetBeans IDE змагається з найкращим інтегрованими середовищами розробки для мови Java, підтримуючи [рефакторинг](https://uk.wikipedia.org/wiki/%D0%A0%D0%B5%D1%84%D0%B0%D0%BA%D1%82%D0%BE%D1%80%D0%B8%D0%BD%D0%B3), [профілювання](https://uk.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D1%84%D1%96%D0%BB%D1%8E%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F_(%D0%BA%D0%BE%D0%BC%D0%BF%27%D1%8E%D1%82%D0%B5%D1%80%D0%BD%D1%96_%D0%BD%D0%B0%D1%83%D0%BA%D0%B8)), виділення синтаксичних конструкцій кольором, автодоповнення мовних конструкцій на льоту, шаблони коду тощо.

H2 - відкрита кроссплатформенная СУБД, повністю написана на мові Java.

Незважаючи на малий розмір (трохи більше 1 МБ) H2 підтримує такі можливості:

Два режиму роботи (клієнт-сервер, вбудований)

Два режиму зберігання даних (файлова система, пам'ять)

Підтримка планів виконання запитів

Підтримка кластеризації і реплікації

шифрування даних

Зовнішні (пов'язані) таблиці

драйвер ODBC

повнотекстовий пошук

визначення доменів

Мультіверсіонний конкурентний доступ

підтримка послідовностей

Підтримка ключових слів LIMIT і OFFSET в запитах

тимчасові таблиці

обчислювані стовпці

Призначені для користувача агрегатні функції

Призначені для користувача процедури,

Стиснення CLOB / BLOB об'єктів

Робота з CSV файлами на читання і запис

Браузерна консоль управління

Запуск як сервіс Windows

Що ще не реалізовано:

Віконні функції (реалізована тільки функція ROW\_NUMBER () OVER ())

Многопоточная обробка запитів

Повна підтримка стандарту SQL 2003

Застосовується H2, наприклад, в проекті Grails як базова СУБД для розробки.

Java Сокети ([англ.](https://uk.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D1%96%D0%B9%D1%81%D1%8C%D0%BA%D0%B0_%D0%BC%D0%BE%D0%B2%D0%B0) socket - заглиблення, гніздо, роз'єм) — назва [програмного інтерфейсу](https://uk.wikipedia.org/wiki/API) для забезпечення обміну даними між [процесами](https://uk.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D1%86%D0%B5%D1%81_(%D1%96%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0)) . Процеси при такому обміні можуть виконуватися як на одній ЕОМ, так і на різних ЕОМ, пов'язаних між собою [мережею](https://uk.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%27%D1%8E%D1%82%D0%B5%D1%80%D0%BD%D0%B0_%D0%BC%D0%B5%D1%80%D0%B5%D0%B6%D0%B0). Сокет - абстрактний об'єкт, що представляє кінцеву точку з'єднання.

## 3.2 Опис програмних бібліотек

JavaMail - це Java API, призначене для отримання і відправки електронної пошти з використанням протоколів SMTP, POP3 та IMAP. JavaMail є частиною платформи Java EE, але також доступний в якості додаткового пакета для використання в додатках Java SE.

Існує також альтернативна реалізація JavaMail з відкритим вихідним кодом - GNU JavaMail - яка реалізує тільки специфікацію JavaMail версії 1.3; це єдина вільна реалізація, що підтримує протокол NNTP, дозволяючи читати і відсилати статті в новинні групи.

До складу JavaMail не входить поштовий сервер, проте для подібних завдань можуть використовуватися вільні Apache James і Java Email Server (POP3 і SMTP), або бібліотека SubEthaSMTP для створення SMTP сервера.

JDOM це вільна Java-реалізація DOM для XML, створена з урахуванням особливостей мови і платформи Java. JDOM інтегрується з Document Object Model (DOM) і Simple API for XML (SAX), підтримує XPath і XSLT. У JDOM використовуються зовнішні парсери для генерації документів.

## 3.3 Особливості створення програмних модулів з урахуванням мови програмування

В основі створення програмних модулів лежить ООП підхід. Java відноситься до мов об'єктно-орієнтованого типу, тому, повністю підходить для написання програмних модулів.

Об'є́ктно-орієнто́ване програмува́ння (ООП) — одна з [парадигм](https://uk.wikipedia.org/wiki/%D0%9F%D0%B0%D1%80%D0%B0%D0%B4%D0%B8%D0%B3%D0%BC%D0%B8_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F) [програмування](https://uk.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F), яка розглядає програму як множину «об'єктів», що взаємодіють між собою. Основу ООП складають три основні концепції: [інкапсуляція](https://uk.wikipedia.org/wiki/%D0%86%D0%BD%D0%BA%D0%B0%D0%BF%D1%81%D1%83%D0%BB%D1%8F%D1%86%D1%96%D1%8F), [успадкування](https://uk.wikipedia.org/wiki/%D0%A3%D1%81%D0%BF%D0%B0%D0%B4%D0%BA%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F)) та [поліморфізм](https://uk.wikipedia.org/wiki/%D0%9F%D0%BE%D0%BB%D1%96%D0%BC%D0%BE%D1%80%D1%84%D1%96%D0%B7%D0%BC_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F)). Одною з переваг ООП є краща [модульність](https://uk.wikipedia.org/wiki/%D0%9C%D0%BE%D0%B4%D1%83%D0%BB%D1%8C_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F)) програмного забезпечення (тисячу функцій процедурної мови, в ООП можна замінити кількома десятками класів із своїми методами).

## 3.4 Особливості створення структур даних

Для збереження і відновлення даних на стороні клієнту використовуються xml-файли.

Структура даних по компаніям (для розрахунку даних) повинна бути така:

<companies>

<company>

<name>Назва компанії</name>

<depo>Початковий депозит</depo>

<persent>Середній відсоток інвестування</persent>

<period>Період інвестування</period>

<intervals>

<i>

<plus>Сумма довкладень за i-й період</plus>

<minus>Сумма знімань коштів за i-й період </minus>

<reinvesting>Перенесення поточної сумми у наступний період інвестування</reinvesting>

</i>

</intervals>

</company>

</companies>

Усі маніпуляції виконуються із даним файлом через JDOM парсер.

Дані на стороні серверу зберігаються в БД. Доступ для елементів комірок доступний через активне підключення до БД із виконанням sql-подібних запитів.

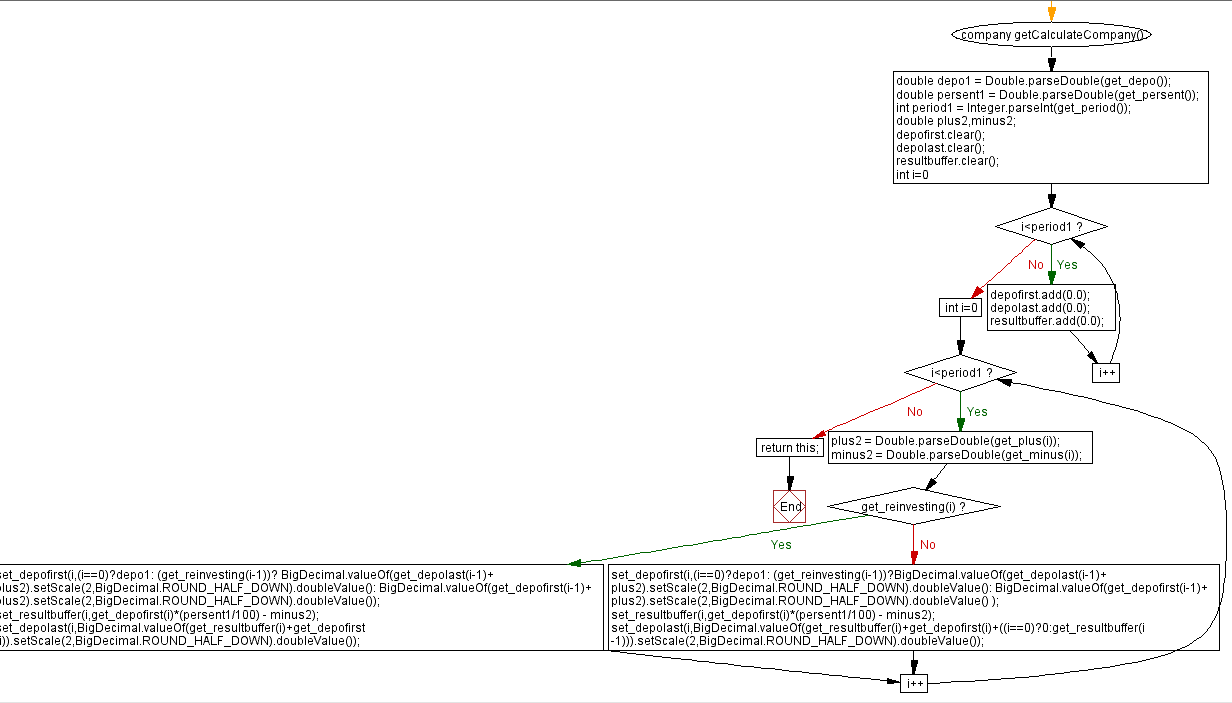
Приклад вибірки даних усіх користувачів із БД:

ResultSet result = st.executeQuery("SELECT \* FROM USER");  
while (result.next()) {  
 // Виконуються необхідні маніпуляції

}

## 3.5 Модульне тестування

1 Блок-схема алгоритму розрахунку даних (блок-схема автоматично створена в програмі на основі коду програми, надалі блок-схема буде перерисована):

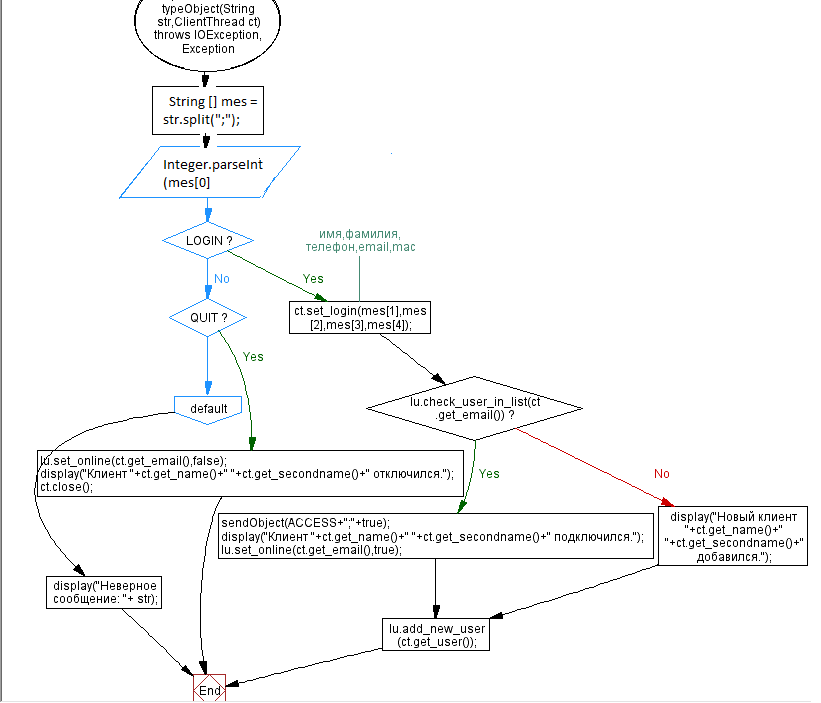


Метод чорного ящика:

Еквівалентне розбиття:

|  |  |  |
| --- | --- | --- |
| Вхідна умова | Правильний клас еквівалентності | Неправильний клас еквівалентності |
| Depo = 100; persent = 7; period = 10; reinvesting = true; | 191,72 – вірне округлення | 191,7151 – невірні дані без округлення чисел |
| Depo = 100; persent = 7; period = 10; reinvesting = true; plus[0]=100 | 420,97 – плюс додається до 0-го періоду | 380,56 – коли переноситься на наступну дату при реінвестуванні |
| Depo = 100; persent = 7; period = 10; reinvesting = true; plus[0]=100; minus[0]=100; | 191,72 – вірні дані зняття та до вкладення сум | 208,59 – невірна сума, бо дані до вкладені на початок, а зняті в кінці місяця, а потім сума перенесена на наступний період |

2 Блок-схема обміну даних між клієнтом та сервером. Серверна частина. (блок-схема автоматично створена в програмі, надалі блок-схема буде перерисована):



Метод білого ящика:

Покриття умов:

Подача сигналу на вхід:

Дані введення: 0;Peter;Buslyuk;34242423;skyliner270594@mail.ru;04-7D-7B-97-0C-9F

Перевірка коректності – усі дані вірні, обмежень немає

Дані введення: 0;Peter;Buslyuk;34242423;skyliner270594@mail.ru;04-7D-7B-97-0C-99F

Перевірка коректності – невірно, мак-адреса не вірна

Дані введення: 0;Peter;Buslyuk;34242423;skyliner270594mail.ru;04-7D-7B-97-0C-99F

Перевірка коректності – невірно, мак-адреса та е-мейл невірний

Дані введення: 3;Peter;Buslyuk;34242423;skyliner270594mail.ru;04-7D-7B-97-0C-99F

Перевірка коректності – невірно, такого типу вибору не існує

Перевірка на існування користувача:

Дані введення:skyliner270594@mail.ru – користувач існує

Skyliner11270594mail.ru – користувач не існує, створюється новий користувач

# 4 ТЕСТУВАННЯ ПРОГРАМНОГО ПРОДУКТУ



## Функціональне тестування

|  |  |  |  |
| --- | --- | --- | --- |
| № прецеденту | Дія | Очікуваний результат | Результат тестування (успішний(passed)/ невдачний(failed) |
| 1 | Заповнення профілю вірними даними. | Система клієнта зберігає дані клієнта. Дані збережені. | Passed |
| 1 | Система Клієнта відправляє дані авторизації на сервер. | Дані клієнта прийняті на сервері та оброблені. | Passed |
| 1 | Сервер шукає дані клієнта в БД та відсилає доступ. | Доступ наданий | Passed |
| 1 | Сервер не знайшов клієнта в БД. | Сервер створив нового клієнта та надіслав доступ. | Passed |
| 2 | Клієнт обирає збереження даних компаній. | Система Клієнта перевіряє дані та зберігає їх у XML файл | Passed |
| 2 | Перевірка даних не коректність при збереженні. | Система Клієнта видає повідомлення щодо невірних даних. | Failed |
| 3 | Збереження даних віддалено. | Система Клієнта перевіряє дані та відсилає їх на Сервер. | Failed |
| 3 | Сервер отримує дані та зберігає в БД. | Сервер зберіг дані. | Passed |
| 3 | Вказано невірні дані. | Система Клієнта пропонує ввести дані ще раз. | Passed |
| 4 | Розрахунок обраних компаній. | Система Клієнта розраховує дані та видає результат. | Passed |
| 5 | Оповіщення клієнтів на обрані емейл адреси. | Клієнти оповіщені. | Passed |

## 5 РОЗГОРТАННЯ ПРОГРАМНОГО ПРОДУКТУ

## 5.1 Інструкція з встановлення

Для роботи з програмою перед її інсталяцією необхідно встановити:

1) Jdk 7 версії або вище;

2) БД H2 та зробити бекап із файлу дампу, котрий знаходиться на github репозиторії – для серверної частини.

3) Запустити jar-файл із програмою.

## 5.2 Інструкція з використання

Початкова загрузка програми відбувається із того, що клієнту пропонується ввести дані щодо його подальшої роботи, а потім зберегти дані локально. Профіль клієнта знаходиться в пункті меню «Профіль». Після чого з’являється форма, зображена на Рисунку 1.

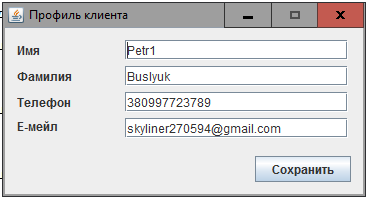


Рис. 1 – Форма клієнта «Профіль клієнта»

Клієнт повинен заповнити дані імені, фамілії, телефону та емейлу для інформування клієнта щодо інвестиційних новин серед нещодавно створених, або свої інвестиційних продуктів.

Наступним кроком необхідно додати нову компанію для розрахунку даних. Для цього обирається розділ меню «Компанії->Додати компанію», або натискається комбінацію клавіш Ctrl+N.

Далі необхідно ввести назву компанії, початковий депозит, середній відсоток та продовжність інвестування. Після цього створена компанія з’явиться в таблиці. На рисунку 2 зображено створені компанії, а виділеним рядком – щойно створена.

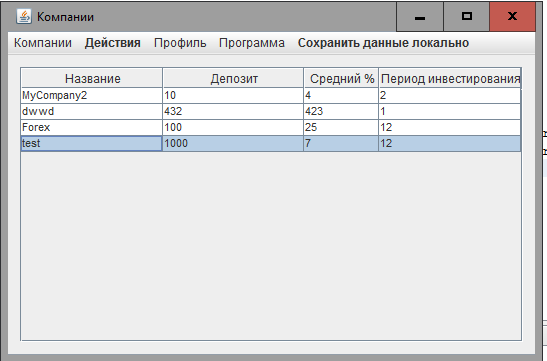


Рисунок 2 – Форма «Компанії»

Перейдемо до щойно створеної компанії двійним натисненням лівої кнопки миші на неї.

Висвітиться вікно, показане на рисунку 3.

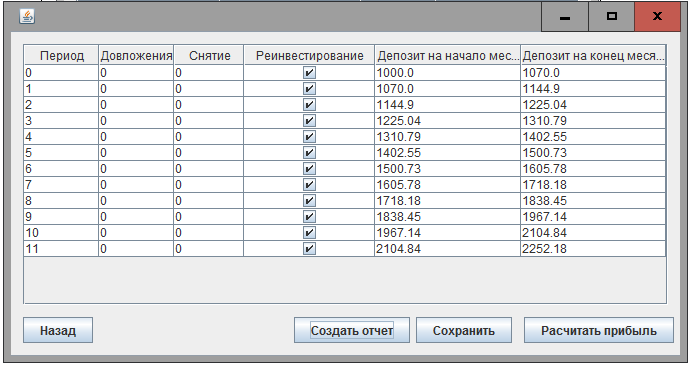


Рисунок 3 – Форма обраної компанії

Тут ми можемо зберегти дані, створити звіт та розрахувати дані.

Розглянемо створення звіту. Звіт можна створити 2 шляхами – через вибір декількох компаній, і безпосередньо в обраній в 1 екземлярі компанії.

Для створення звіту необхідно буде ввести назву звіту, для того, щоб ви змогли в майбутньому його відшукати на комп’ютері. За умовчанням він зберігається в папці з програмою. Автоматично він зберігається в pdf форматі - рисунок 4.



Рисунок 4 – Приклад тестового звіту.

Приклад звіту всередині можна проглянути на рисунку 5.

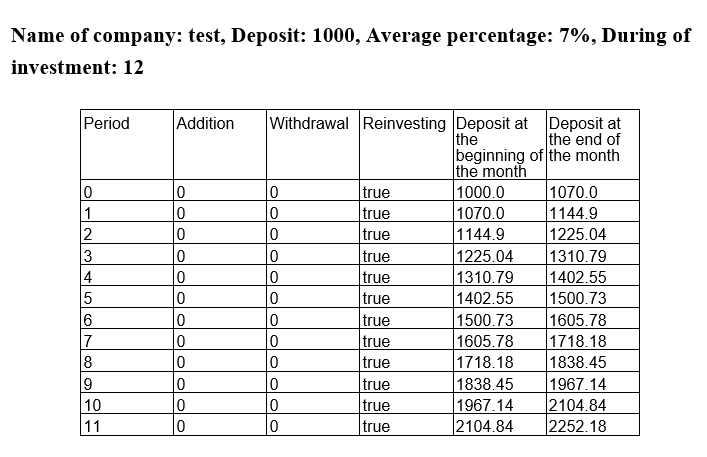


Рисунок 5 – Приклад звіту в pdf –файлу

Інструкція оператора (сервер):

Коли оператор входить до серверної частини програми, то він попадає на головну форму, зображену на рисунку 6.

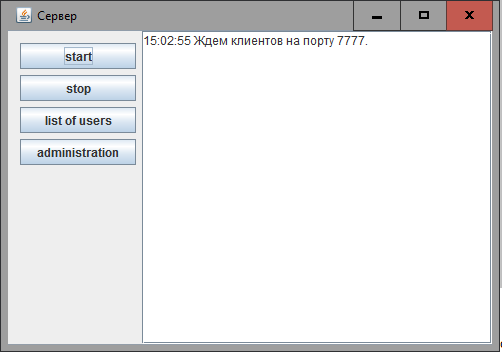


Рис. 6 – Головна форма серверу

Тут ми можемо бачити запуск та зупинення сервера, список клієнтів та адміністрування основних деталей.

При виборі списку клієнтів ми попадаємо на форму, де можемо проглянути список усіх користувачів, а також їх поточний статус, та персональні дані. Форма з клієнтами представлена на рисунку 7.

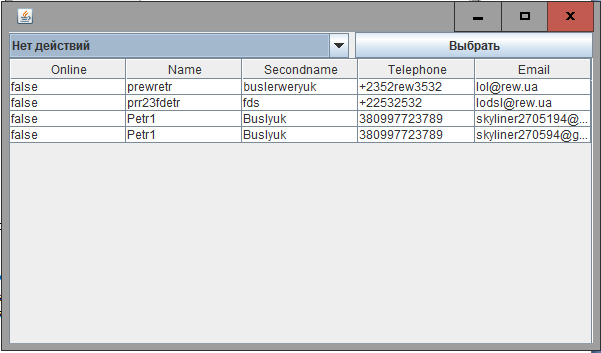
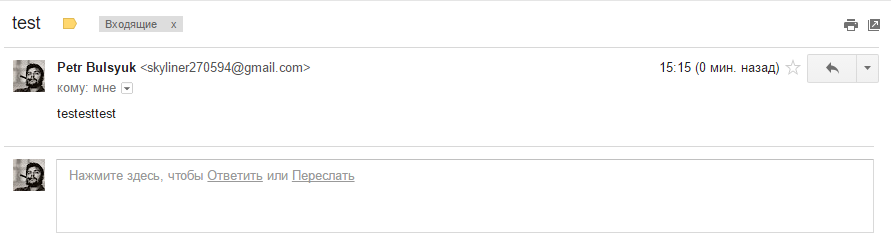


Рисунок 7 – Форма клієнтів

Ми можемо відіслати заданим користувачам повідомлення на емейл адресу. Для цього необхідно обрати необхідних клієнтів на у випадаючому списку обрати «Вислати повідомлення» та натиснути на кнопку «Обрати».

Система запросить ввести Вас заголовок повідомлення та саме повідомлення. Після цього проінформує про відправлення.

Результат можна проглянути на рисунку 8.

Рисунок 8 - Поштовий ящик із результатом відправлених даних на вказаного клієнта.