### Реферат до семестрової роботи

### Семирог-Орлика Єгора

#### ****Тема:**** Реалізація функціоналу банківської системи мовою програмування C

#### ****Вступ****

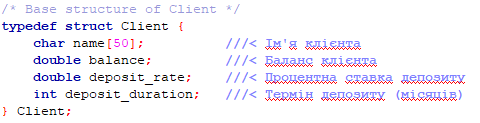
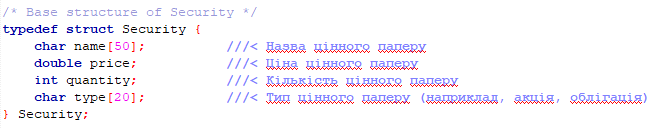
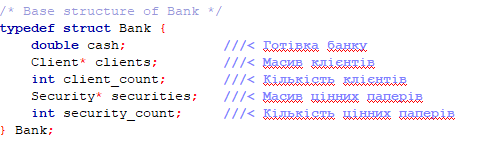
Сучасні інформаційні системи вимагають інтеграції різноманітних функціоналів, що підтримують ефективне управління фінансовими процесами. У даному рефераті розглядається приклад створення банківської системи мовою програмування C, яка дозволяє управляти клієнтами, депозитами, цінними паперами, а також зберігати й обробляти дані у форматі CSV.

#### ****Опис програми****

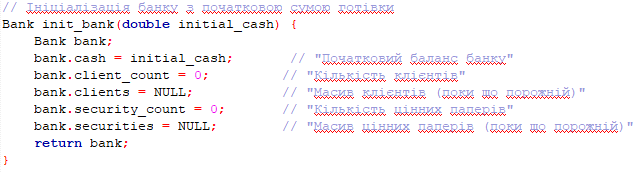
##### **Структура програми**

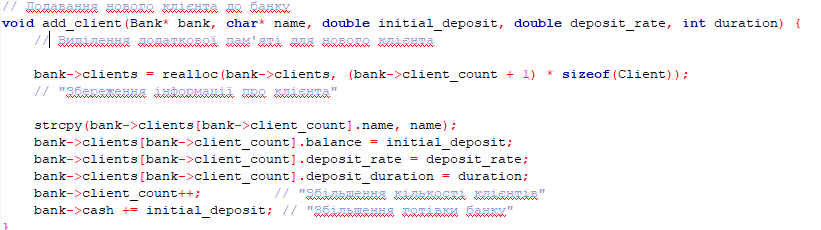
Програма складається з основного файлу main.c, заголовкового файлу Bank.h, а також файла реалізації функцій Bank.c. Вона організована у вигляді модульної системи для розділення логіки обробки даних і взаємодії з користувачем.

**Основні структури даних:**

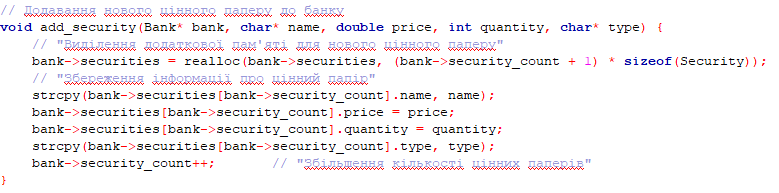
* **Client**: Зберігає дані про клієнта банку (ім’я, баланс, відсоткову ставку депозиту, термін депозиту).
* **Security**: Описує цінний папір (назва, ціна, кількість, тип).
* **Bank**: Головна структура для управління банком (готівка, масив клієнтів, масив цінних паперів). 

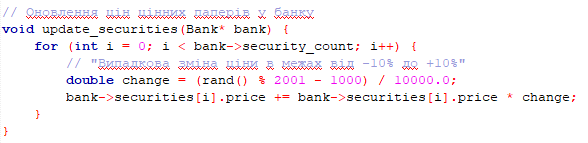
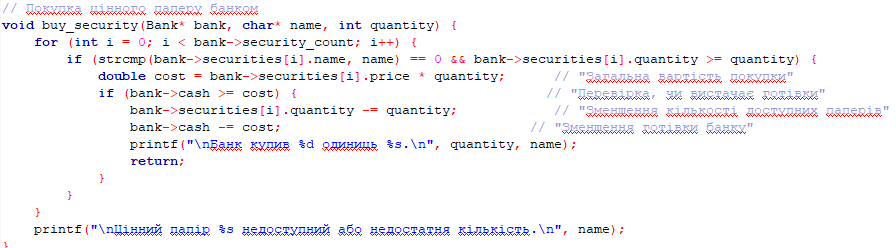
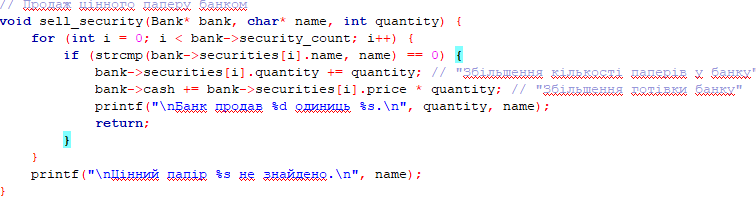
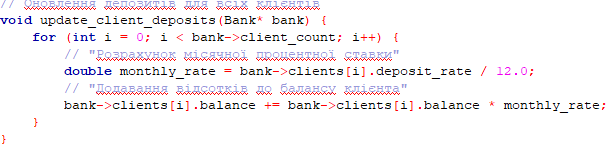
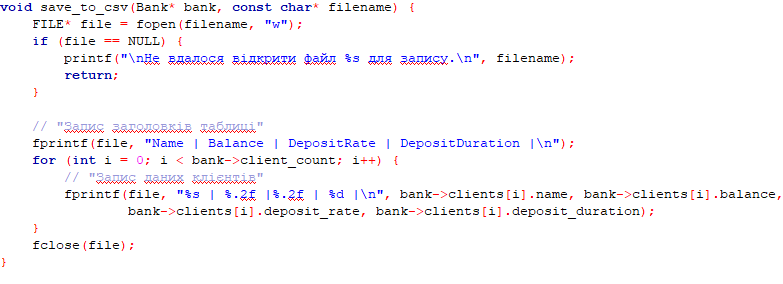
##### **Функціонал програми**

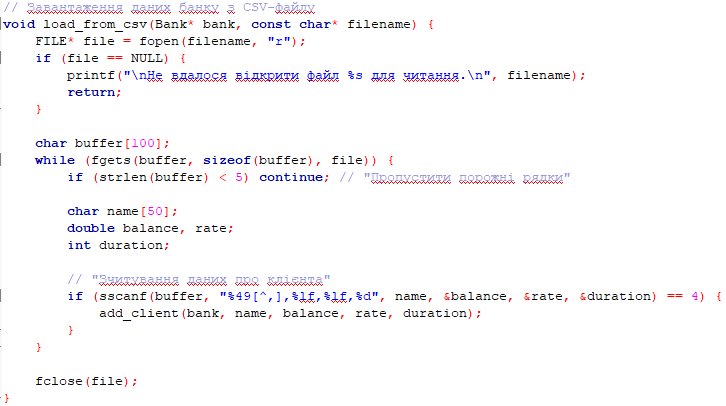
1. **Ініціалізація банку (init\_bank)**  
   Функція створює початкову структуру банку з заданою сумою готівки, нульовою кількістю клієнтів і цінних паперів.
2. **Додавання клієнта (add\_client)**  
   Ця функція дозволяє додати нового клієнта з початковим депозитом, відсотковою ставкою та строком депозиту. При цьому відбувається:

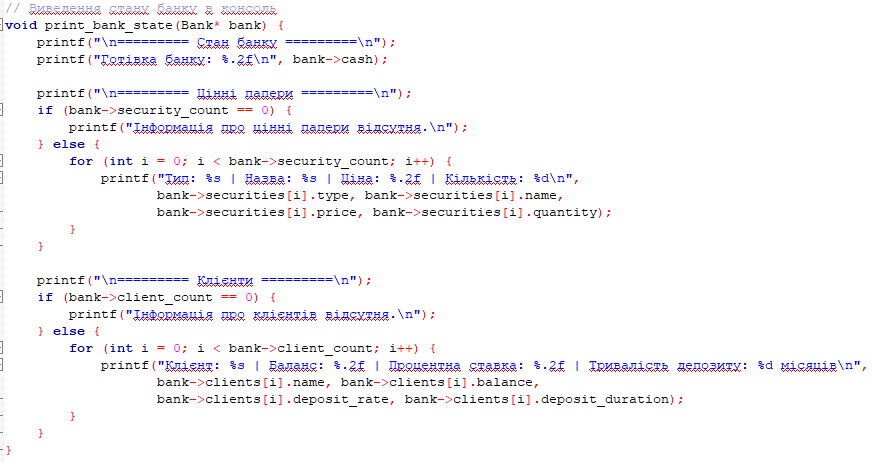
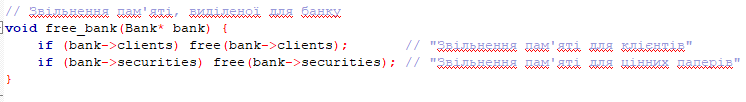
* Динамічне виділення пам’яті для нового запису клієнта.
* Збільшення загальної кількості клієнтів банку.
* Зарахування депозиту на баланс банку.

1. **Додавання цінного паперу (add\_security)**  
   Функція дозволяє додати новий вид цінного паперу з його назвою, ціною, кількістю та типом (наприклад, акція чи облігація).



1. **Оновлення цін цінних паперів (update\_securities)**  
   Ціни на цінні папери змінюються випадковим чином у межах від -10% до +10%. Це дозволяє моделювати динаміку ринку.
2. **Купівля цінного паперу (buy\_security)**  
   Банк може придбати певну кількість цінних паперів за умови наявності достатньої готівки. Якщо угода успішна, кількість паперів у банку зменшується, а баланс банку знижується.
3. **Продаж цінного паперу (sell\_security)**  
   Функція дозволяє банку продати певну кількість паперів. У результаті кількість паперів збільшується, а баланс банку поповнюється.
4. **Оновлення депозитів клієнтів (update\_client\_deposits)**  
   Ця функція розраховує щомісячний дохід від депозиту для кожного клієнта та додає його до балансу клієнта.
5. **Збереження даних у CSV-файл (save\_to\_csv)**  
   Функція записує інформацію про клієнтів банку у текстовий файл формату CSV. Це дає змогу зберігати дані для подальшого використання.
6. **Завантаження даних із CSV-файлу (load\_from\_csv)**  
   Функція дозволяє завантажити дані клієнтів із файлу CSV, динамічно додаючи їх до існуючої бази.



1. **Виведення стану банку (print\_bank\_state)**  
   Відображає поточний стан банку, включаючи баланс, список клієнтів і доступні цінні папери.
2. **Очищення пам’яті (free\_bank)**  
   Функція звільняє динамічно виділену пам’ять для масивів клієнтів і цінних паперів, запобігаючи витоку пам’яті.

##### **Інтерфейс користувача**

У головному файлі main.c реалізовано меню, яке дозволяє користувачу виконувати наступні операції:

* Додавати клієнтів та цінні папери.
* Оновлювати депозити та ціни.
* Здійснювати купівлю й продаж цінних паперів.
* Зберігати та завантажувати дані.

Меню побудоване у вигляді циклу do-while, що забезпечує повторне виконання операцій до вибору пункту "Вихід".

#### ****Висновок****

Розглянута програма демонструє базову реалізацію банківської системи, яка включає роботу з клієнтами, депозитами та цінними паперами. Її основною перевагою є використання модульного підходу, динамічного управління пам’яттю та підтримка збереження й завантаження даних із зовнішніх файлів. У майбутньому її можна розширити, додавши складніші фінансові операції або інтеграцію з мережею банківських установ.