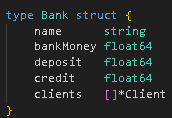
**ЛАБОРАТОРНА РОБОТА № 6**

**Паралельне програмування**

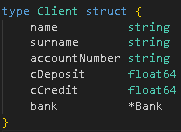
***Мета:*** засвоїти принцип проектування програми з використанням паралельного програмування; ви-вчити особливості застосування: go-рутин, каналів, синхронізації go-рутин

**Хід роботи:**

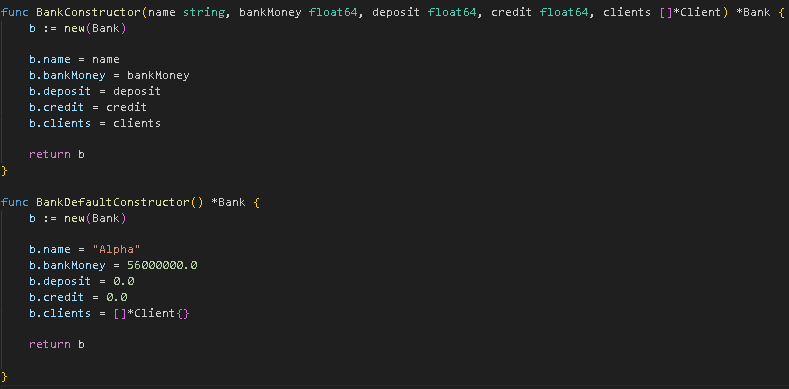
**Завдання 1**: Оголосив структуру Bank, яка представляє інформацію про наявність грошей, які зберігаються у банку. Має наступні поля

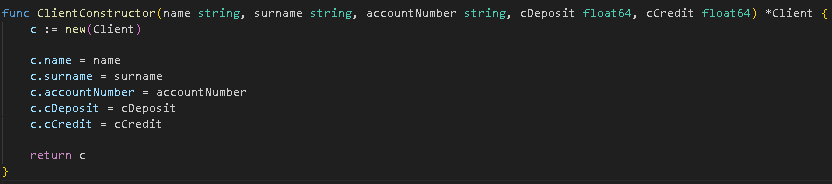


Оголосив структуру Client, яка містить наступні поля

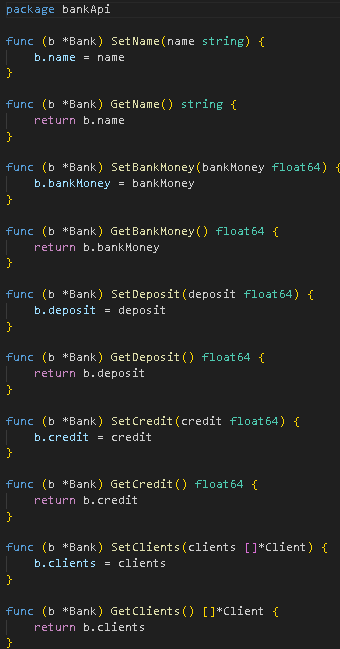
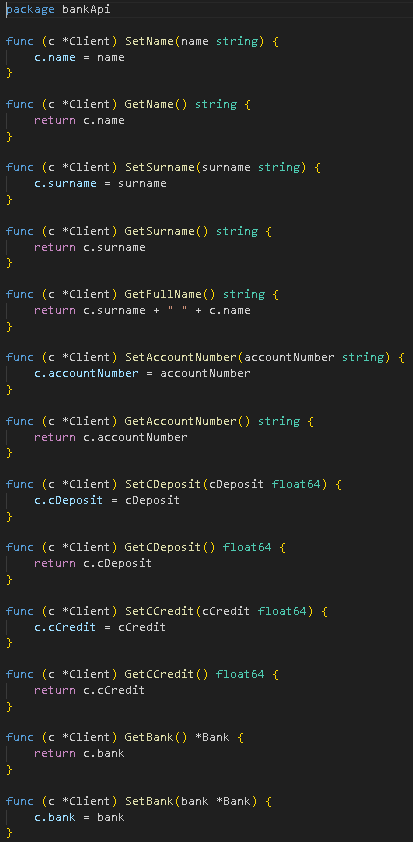


**Завдання 2**: Для кожної структури реалізував конструктори.





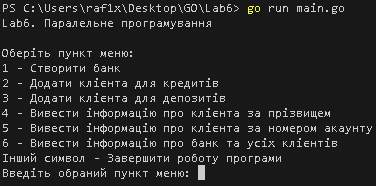
**Завдання 3**:. Реалізував set- та get- методи для кожної структури

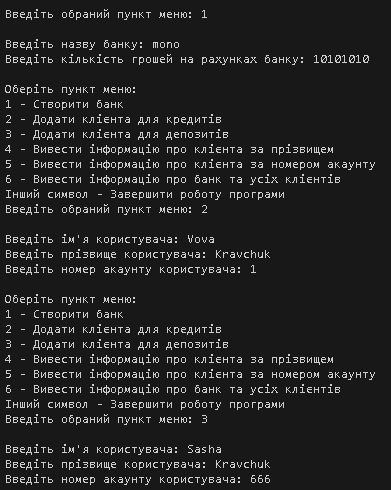
**Завдання 4**:. Реалізував необхідні методи для структур.

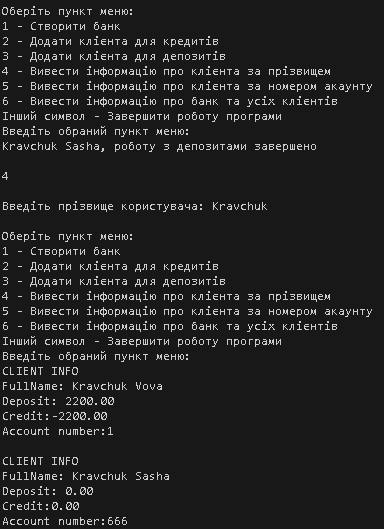
**Завдання 5**:. Виконав усі необхідні перевірки на можливість отримання грошей з рахунку

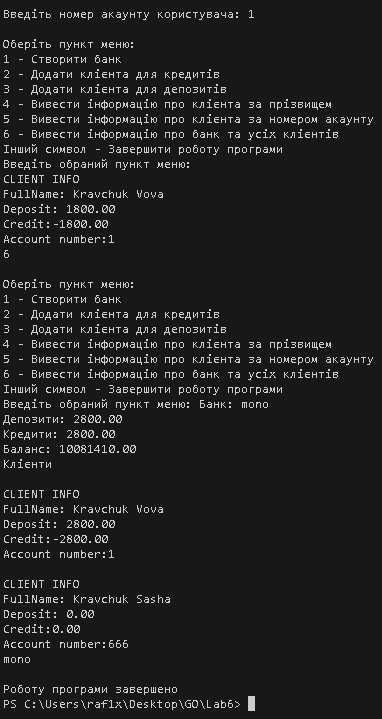
**Завдання 6**:. Створив консольне меню з наступними пунктами:



Результат виконання програми







***Висновки:*** в ході виконання лабораторної роботи засвоїв принцип проектування програми з використанням паралельного програмування; ви-вчив особливості застосування: go-рутин, каналів, синхронізації go-рутин

**Контрольні питання**

1. **Що таке процес?**

*Процесом називають поточну програму і всі її елементи: адресний простір, глобальні змінні, регістри, стек, відкриті файли і так далі.*

1. **Що таке потік?**

*Потік виконання – найменша одиниця обробки. Потік виконання знаходиться всередині процесу. Кілька потоків виконання можуть існувати в рамках одного і того ж процесу і спільно використовувати ресурси, тоді як процеси не поділяють цих ресурсів, а взаємодія відбувається за допомогою спеціальних механізмів*

1. **Які особливості потоків у мові Go?**

*У Go паралельне програмування засноване на моделі CSP (Communicating Sequential Processes – взаємодіючі послідовні процеси). У мові Go є дві особливості. Перша: go-підпрограми – легковагі потоки виконання. Друга: канали, що забезпечують надійний засіб одно- і двостороннього обміну даними між go підпрограмами*

*Особливості go-підпрограм:*

*- незалежно виконується функція, запущена за допомогою інструкції go;*

*- наявність власного стека;*

*- невеликі витрати по пам'яті ~ 2 Кб-4 Кб;*

*- Runtime Go займається розподілом горутін по процесам.*

1. **Як створити потік в мові Go?**

*Горутини можна уявляти як «легковисні потоки», щоб створити горутину потрі-бно просто поставити ключове слово go перед кодом виклику функції. Щоб проде-монструвати наскільки це просто, давайте створимо дві функції пошуку, викличемо їх з ключовим словом go і друкуватимемо повідомлення кожного разу, коли вони знайдуть «руду» у своїй копальні.*

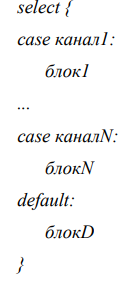
*Канали дозволяють горутин обмінюватися даними. Це своєрідна труба, через яку горутини можуть посилати та приймати інформацію від інших горутин. Читання та запис у канал здійснюється за допомогою оператора-стрілочки (<-), який вказує на-прямок руху даних.*

1. **Для чого призначені канали?**

*Канали – це механізм взаємодії між go-підпрограмами, які паралельно виконуються. При використанні каналів синхронізація кінців каналу (і відповідних їм go-підпрограм), які відправляю і приймають, виконується в момент взаємодії*

1. **Призначення керуючої конструкції select?**

*Інструкція select використовується для вибору каналу, який виконав транзакцію. Похожа на оператор switch. Має наступний вигляд:*

****