

Задание

Нужно написать консольное приложение, которое моделирует работу банка: создание счетов, операции, отчёты и начисления.

Требования к предметной области

Сущности

1. Клиент (Customer)

- Поля (все private): `id`, `fullName`
- Только через конструктор + геттеры
- `id` уникален (можно просто увеличивать счётчик в коде)

2. Счёт (Account) — базовый класс

- private поля:
`accountNumber`, `balance`, `owner (Customer)`
- Методы:
 - o `deposit(double amount)` — пополнение
 - o `withdraw(double amount)` — снятие (может быть переопределено)
 - o `transfer(Account to, double amount)` — перевод между счетами
 - o `getBalance()`, `getAccountNumber()`, `getOwner()`
- Валидация:
 - o суммы > 0
 - o нельзя снять больше, чем доступно (если тип счёта не допускает)
- В `deposit/withdraw/transfer` добавь проверки через `if` и возвращай `boolean` (успешно/нет)

3. Два типа счетов (наследование)

1. `DebitAccount` extends `Account`

- Снимать можно только если `balance >= amount`

2. `CreditAccount` extends `Account`

- Есть private `double creditLimit` (например 10_000)
 - Баланс может уходить в минус до `-creditLimit`
 - Переопредели `withdraw`
-

2) История операций (циклы + методы)

Класс **Transaction**:

- type (например enum: DEPOSIT/WITHDRAW/TRANSFER)
- amount
- fromAccountNumber (может быть null)
- toAccountNumber (может быть null)
- timestamp (можно LocalDateTime)
- success (boolean)
- message (строка с причиной ошибки или "OK")

Банк должен хранить список операций:

- `ArrayList<Transaction> transactions`
-

3) Класс Bank (главная логика)

Bank хранит:

- список клиентов
- список счетов
- список транзакций

Методы (обязательные):

- `Customer createCustomer(String fullName)`
 - `Account openDebitAccount(Customer owner)`
 - `Account openCreditAccount(Customer owner, double creditLimit)`
 - `Account findAccount(String accountNumber)` (или по номеру)
 - `boolean deposit(String accountNumber, double amount)`
 - `boolean withdraw(String accountNumber, double amount)`
 - `boolean transfer(String from, String to, double amount)`
 - `void printCustomerAccounts(int customerId)` — вывести все счета клиента
 - `void printTransactions()` — вывести все операции
 - `void printReport()` — общий отчёт:
 - сколько счетов каждого типа
 - суммарные балансы по типам
 - количество успешных/неуспешных операций
-

4) Консольное меню (условия + циклы)

1. Создать клиента
2. Открыть дебетовый счёт
3. Открыть кредитный счёт
4. Пополнить
5. Снять

6. Перевести
7. Показать счета клиента
8. Показать транзакции
9. Отчёт банка
10. Выход