**ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНТЕЛЕКТУАЛЬНИХ ТЕХНОЛОГІЙ І ЗВ’ЯЗКУ**

**ВІДОКРЕМЛЕНИЙ СТРУКТУРНИЙ ПІДРОЗДІЛ**

**«ФАХОВИЙ КОЛЕДЖ ЗВ’ЯЗКУ ТА ІНФОРМАТИЗАЦІЇ**

**ДЕРЖАВНОГО УНІВЕРСИТЕТУ ІНТЕЛЕКТУАЛЬНИХ ТЕХНОЛОГІЙ І ЗВ’ЯЗКУ»**

Циклова комісія інформаційних технологій

**Звіт з практичної роботи №4**

**З дисципліни: об'єктно-орієнтоване програмування**

Виконав: студенти 3 курсу, групи КН-31 спеціальності

122 Комп’ютерні науки

освітня програма

Обслуговування програмних систем і комплексів

Ракітенко Артем

Керівник Андрій Куляк

Одеса – 2025

**ЗМІСТ**

[1. ВСТУП 4](#_Toc211882491)

[2. ДІАГРАМА КЛАСІВ 5](#_Toc211882492)

[3. ПРИКЛАДИ КОДУ 6](#_Toc211882493)

[3.1 Базові абстрактні класи: 6](#_Toc211882494)

[3.2 Наслідники та перевизначення 7](#_Toc211882495)

[3.3 Protected поля в дії 8](#_Toc211882496)

[3.4 Package-private демонстрація 9](#_Toc211882497)

[4. СКРІНШОТИ ПРАЦІ 10](#_Toc211882498)

# **ВСТУП**

**Мета:** Засвоїти наслідування через розробку банківської системи

**Завдання:**

- Створити 3 ієрархії класів

- Використати protected поля

- Перевизначити методи

- Демонструвати package-private

**Система:** Банківська система з користувачами, валютами та операціями

# **ДІАГРАМА КЛАСІВ**

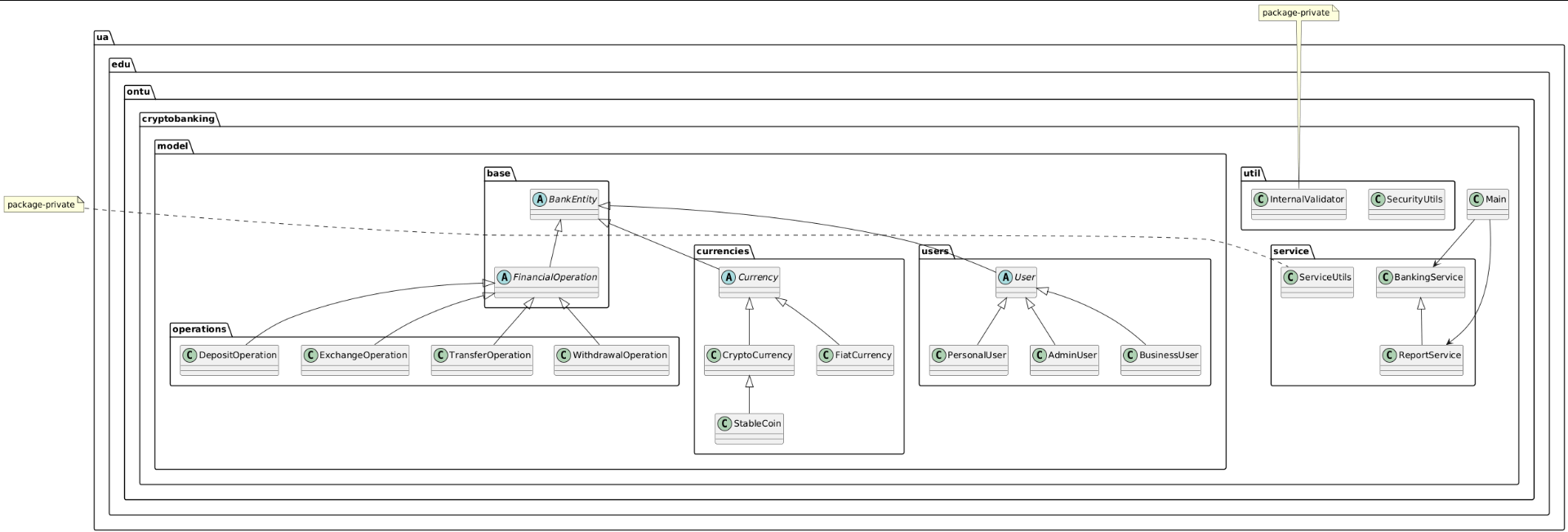


Рисунок 2.1 – Діаграма

3 основні ієрархії:

1. User ← Personal/Business/Admin

2. Currency ← Fiat/Crypto ← StableCoin

3. FinancialOperation ← Transfer/Deposit/Exchange/Withdrawal

# **ПРИКЛАДИ КОДУ**

## 3.1 Базові абстрактні класи:

**- BankEntity.java:**

public abstract class BankEntity {

protected String id;

protected String name;

protected Date createdDate;

public BankEntity(String id, String name) {

this.id = id;

this.name = name;

this.createdDate = new Date();

}

public abstract String getDescription();

}

**- User.java:**

public abstract class User extends BankEntity {

protected String email;

protected String phone;

protected double balance;

public User(String id, String name, String email) {

super(id, name);

this.email = email;

this.balance = 0.0;

}

public abstract String getAccountType();

public abstract void showSpecialFeatures();

public void deposit(double amount) {

if (amount > 0) {

balance += amount;

}

}

}

## 3.2 Наслідники та перевизначення

**- PersonalUser.java:**

public class PersonalUser extends User {

private String firstName;

private String lastName;

public PersonalUser(String id, String firstName, String lastName, String email) {

super(id, firstName + " " + lastName, email); // виклик super()

this.firstName = firstName;

this.lastName = lastName;

}

@Override

public String getAccountType() {

return "ФІЗИЧНА\_ОСОБА";

}

@Override

public void showSpecialFeatures() {

System.out.println("Ліміт зняття: 50,000 в день");

System.out.println("Безкоштовні перекази");

}

@Override

public boolean withdraw(double amount) {

if (amount > 50000) {

System.out.println("Перевищено ліміт для фізичної особи!");

return false;

}

return super.withdraw(amount);

}

}

## 3.3 Protected поля в дії

**- Currency.java:**

public abstract class Currency extends BankEntity {

protected String code;

protected String symbol;

protected double exchangeRate;

public Currency(String id, String name, String code, String symbol, double exchangeRate) {

super(id, name);

this.code = code;

this.symbol = symbol;

this.exchangeRate = exchangeRate;

}

public abstract String getSecurityInfo();

}

**- FiatCurrency.java:**

public class FiatCurrency extends Currency {

private String country;

public FiatCurrency(String id, String name, String code,

String symbol, double exchangeRate, String country) {

super(id, name, code, symbol, exchangeRate);

this.country = country;

}

@Override

public String getSecurityInfo() {

return "Валюта: " + code + ", курс: " + exchangeRate;

}

}

## 3.4 Package-private демонстрація

**- ServiceUtils.java:**

package ua.edu.ontu.cryptobanking.service;

class ServiceUtils {

static void performInternalServiceCheck() {

System.out.println("Внутрішня перевірка сервісу...");

}

static String generateServiceToken() {

return "TOKEN\_" + System.currentTimeMillis();

}

}

**- BankingService.java:**

package ua.edu.ontu.cryptobanking.service;

public class BankingService {

void processDailyFees() {

System.out.println("Обробка щоденних комісій...");

}

public void demonstratePackagePrivate() {

ServiceUtils.performInternalServiceCheck();

processDailyFees();

System.out.println("Package-private працює коректно!");

}

}

# **4. СКРІНШОТИ ПРАЦІ**

ЗНІМОК 1: Поліморфізм користувачів

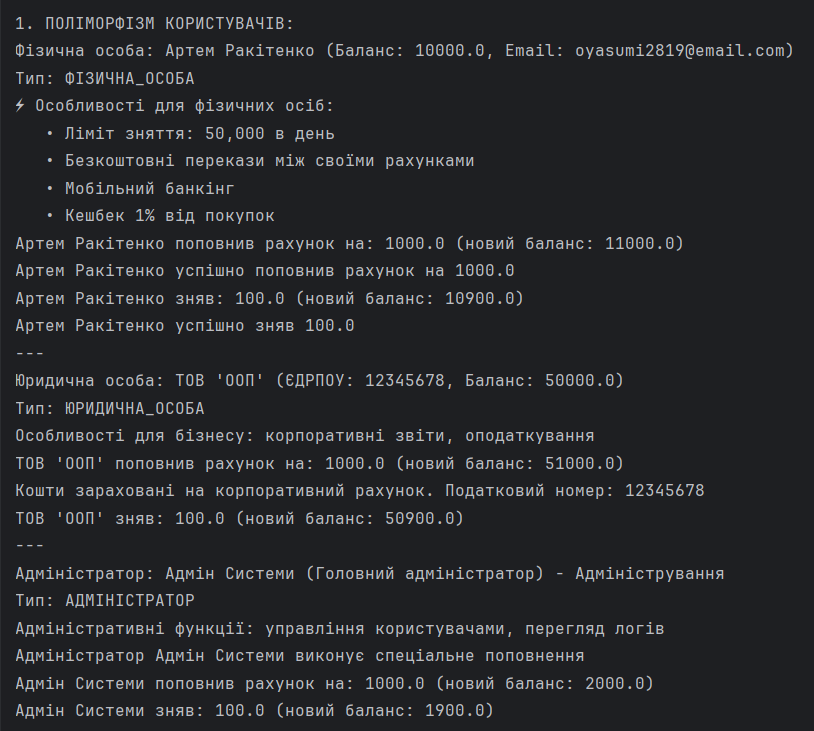


Рисунок 4.1 - Демонстрація поліморфізму.

Різні типи користувачів мають різну реалізацію методів getAccountType() та showSpecialFeatures()

ЗНІМОК 2: Поліморфізм валют

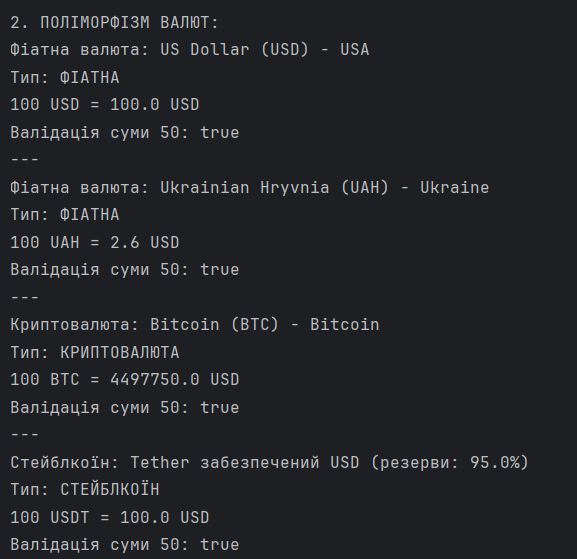


Рисунок 4.2 - Різна реалізація методів convertTo() та validateAmount() для різних типів валют.

ЗНІМОК 3: Операції через сервіс

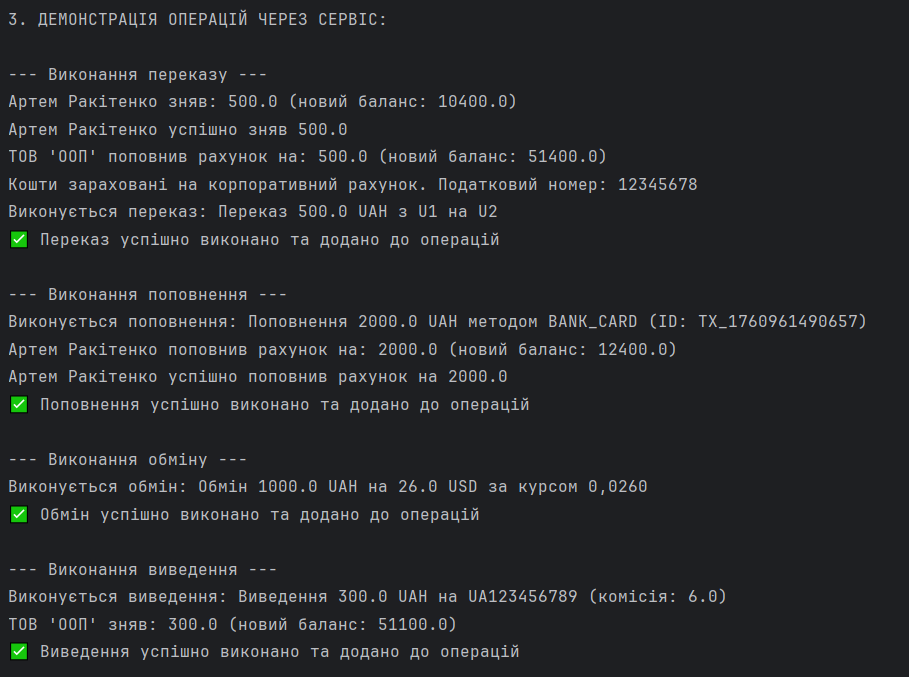


Рисунок 4.3 - Виконання різних типів банківських операцій через BankingService.

ЗНІМОК 4: Перевизначені методи

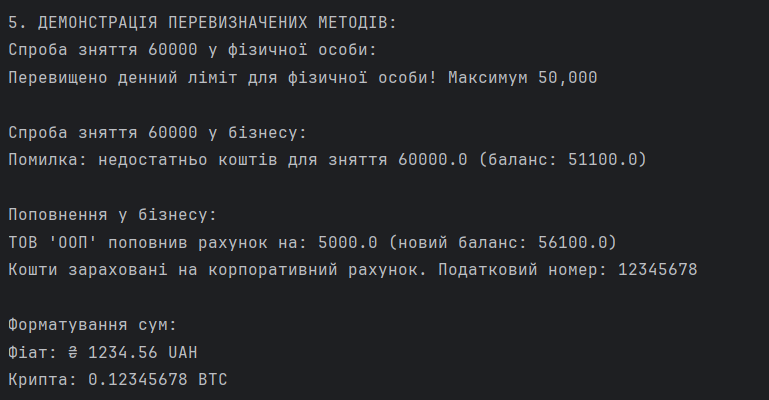


Рисунок 4.4 - Перевизначення методу withdraw() у PersonalUser та formatAmount() у валютах.

ЗНІМОК 5: Package-private доступ

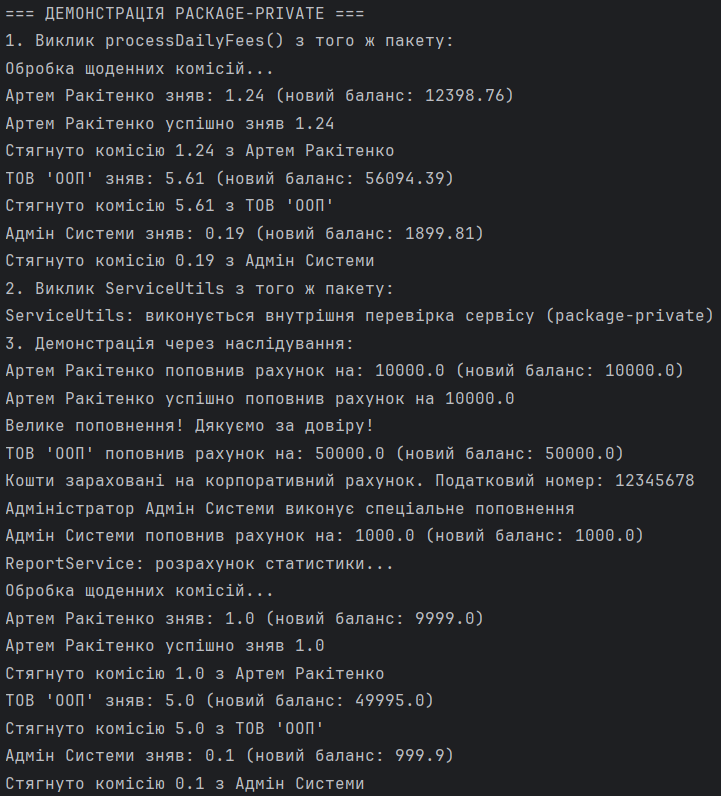


Рисунок 4.5 - Демонстрація package-private методів processDailyFees() та класу ServiceUtils.

ЗНІМОК 6: Абстрактні методи

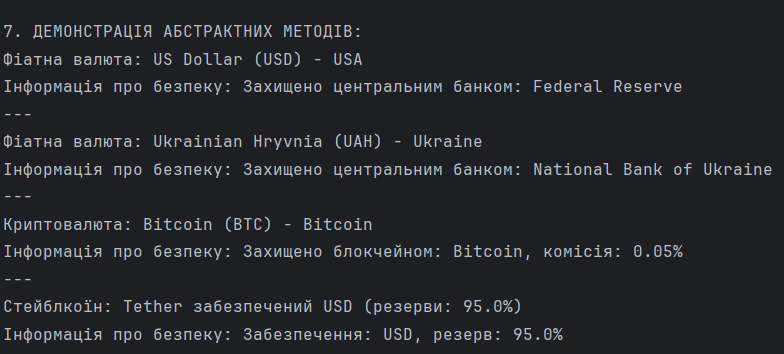


Рисунок 4.6 - Різна реалізація абстрактного методу getSecurityInfo() у ієрархії Currency.

ЗНІМОК 7: Фінальні результати

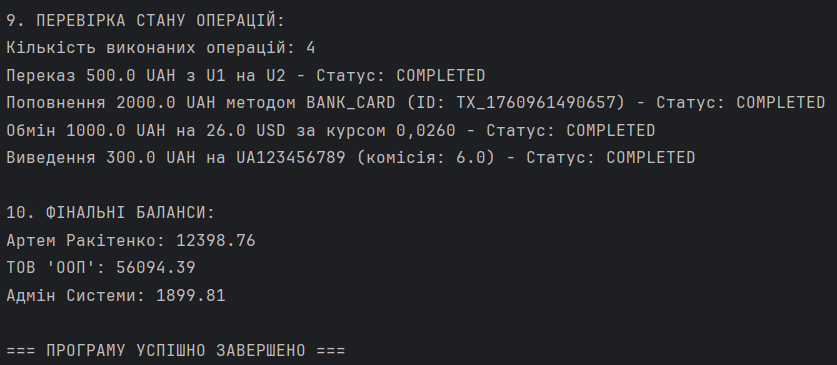


Рисунок 4.7 - Кінцеві результати роботи системи - всі операції успішно виконані.