**Лабораторна робота No7**

**Кохан Денис**

**Тема: Основи мови UML (Unіfіеd Modеlіng Lаnguаgе).**

**Діаграма класів**

**Мета: Отримати навички побудови діаграми класів,**

**створення пакетів та групування класів в пакети.**

**Завдання: створити діаграму класів для сценарію «Додати нове спостереження» прецеденту «Робота з валютними операціями».**

**1. Визначення класів-сутностей**

**CurrencyOperation (Валютна операція)**

**Атрибути:**

* *date* : Date – дата операції
* *operationType* : String – тип операції (купівля або продаж)
* *amount* : double – сума операції

**Методи (операції):**

* *calculateTotal()* – обрахунок загальної суми операції або інша логіка обчислень

**Trader (Трейдер)**

**Атрибути:**

* *name* : String – ім’я трейдера
* *traderID* : String – унікальний ідентифікатор трейдера
* *contactInfo* : String – контактна інформація

**Методи:**

*executeOperation()* – виконання операції; може містити перевірку валідності даних чи логіку виклику операцій

**CurrencyPair (Валютна пара)**

**Атрибути:**

* *name* : String – назва валютної пари (наприклад, USD/EUR)
* *exchangeRate* : double – поточний курс

**Методи:**

*updateRate(newRate: double)* – оновлення курсу відповідно до зміни ринкових даних

Оскільки **CurrencyOperation** може містити багато валютних пар, а кожна пара може бути використана у багатьох операціях, введено додатковий клас:

**OperationItem (Склад валютної операції)**

**Атрибути:**

* *quantity* : double – кількість одиниць валюти
* *rateAtOperation* : double – курс на момент операції

**Методи:**

*calculateItemTotal()* – обрахунок вартості для конкретного рядка операції

**2. Зв’язки між класами**

**Асоціація між *Trader* та *CurrencyOperation***

**Опис:** Один трейдер може здійснити багато валютних операцій, а кожна операція здійснюється лише одним трейдером.

**Кратність:**

* Trader – 1
* CurrencyOperation – 1..n

**Композиція між *CurrencyOperation* та *OperationItem***

**Опис:** Склад операції (OperationItem) є невід'ємною частиною валютної операції. Без операції не може існувати її склад.

**Кратність:**

* CurrencyOperation – 1
* OperationItem – 1..n

**Агрегація між *OperationItem* та *CurrencyPair***

**Опис:** Валютна пара є незалежною сутністю, яка, проте, використовується в кожному рядку операції. Одна валютна пара може бути асоційована з багатьма рядками операцій.

**Кратність:**

* CurrencyPair – 1
* OperationItem – 1 (кожен рядок містить лише одну валютну пару, але одна пара може повторюватися у кількох рядках)

**3. Граничні (Boundary) та керуючі (Control) класи**

**Граничні класи:**

* **TradeOptions (Параметри операцій з валютами)**

**Призначення:** Забезпечує механізм роботи з валютними операціями.

**Методи:**

* *displayOptions()* – відображення доступних параметрів роботи з операціями
* **AddNewCurrencyObservation (Додавання нового спостереження за валютою)**

**Призначення:** Служить для додавання нових валютних операцій.

**Методи:**

* *addObservation()* – ініціалізація або додавання нової операції

Відносини між цими класами є агрегацією:

**Кратність:** 1 до 1 (екземпляр класу *AddNewCurrencyObservation* є частиною класу *TradeOptions*)

**Керуючий клас:**

* **TradeManager (Керування операціями)**

**Призначення:** Реалізовує логіку управління валютними операціями, викликаючи методи відповідних сутностей та граничних класів.

**Методи:**

* *manageTrades()* – загальна логіка контролю та управління операціями

**4. Групування в пакети**

**Пакет "Сутності":**  
Містить класи:

* Trader
* CurrencyOperation
* OperationItem
* CurrencyPair

**Пакет "Границі":**  
Містить класи:

* TradeOptions
* AddNewCurrencyObservation

**Пакет "Контроль":**  
Містить клас:

* TradeManager

**5. Головна діаграма класів**

