

# ЛАБОРАТОРНА РОБОТА №11

## Тема: Можливості діаграми класів

**Мета:** Вивчити призначення діаграми класів. Опанувати методику побудови діаграм класів.

### 1 Завдання для самостійної підготовки

У процесі підготовки до заняття студент в обов'язковому порядку повинен виконати наступні завдання:

- вивчити конспект лекцій;
- опрацювати теоретичну частину лабораторної роботи;
- занести у зошит для лабораторних робіт номер, тему, мету лабораторної роботи.

### 2 Теоретична частина

#### *Призначення діаграми класів*

**Class diagram** (діаграма класів) основна діаграма для створення коду додатка. За допомогою діаграми класів створюється внутрішня структура системи, описується спадкування й взаємне положення класів друг щодо друга. Тут описується логічне представлення системи. Саме логічне, тому що класи - це лише заготовки, на основі яких потім будуть визначені фізичні об'єкти.

**Клас** в мові UML служить для позначення множини об'єктів, які мають однакову структуру, поведінку і відносини з об'єктами інших класів. Графічно клас зображується у вигляді прямокутника, який додатково може бути розділений горизонтальними лініями на розділи або секції. У цих розділах можуть зазначатися ім'я класу, атрибути (змінні) та операції (методи).

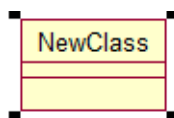


Рисунок 11.1 – Графічне зображення класу

**Ім'я класу** має бути унікальним в межах пакету, який описується деякою сукупністю діаграм класів або однією діаграмою.

**Атрибути класу** або властивості записуються у другій зверху секції прямокутника класу. В UML кожному атрибуту класу відповідає окремий рядок тексту, який складається з квантора видимості атрибута і , можливо, початкового значення:

<квантор видимості><ім'я атрибута>[кратність]: <тип атрибута>=<початкове значення>{рядок властивості}

*Квантор видимості* може приймати одне із трьох можливих значень і відображається за допомогою відповідних спеціальних символів:

- «+» позначає атрибут з областю видимості типу загальнодоступний (*public*). Атрибут з цією областю видимості доступний з будь якого іншого класу пакету, в якому визначена діаграма;
- «#» - атрибут із зоною видимості типу захищений (*protected*). Недоступний для всіх класів, за винятком підкласів даного класу;
- «-» - атрибут із зоною видимості типу закритий «private». Недоступний для всіх класів без винятків;
- «~» - видимість для елементів того ж простору імен (пакета) (*package*).

Квантор видимості може бути опущений. У цьому випадку його відсутність просто означає, що видимість атрибута не вказується.

*Ім'я атрибута* – рядок тексту, який використовується в якості ідентифікатора відповідного атрибута і тому повинний бути унікальним в межах даного класу.

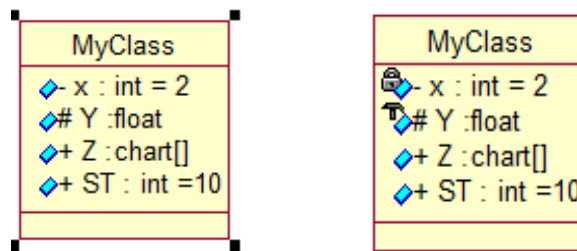


Рисунок 11.2 – Атрибути класу

*Методи класу* – записуються в третій зверху секції прямокутника. Кожному методу класу відповідає окремий рядок, який складається з квантору видимості операції, імені операції, виразу значення, що типу повертаються операцією і, можливо, рядок властивість операції:

<квантор видимості><ім'я методу>(список параметрів): <вираз типу значення, що повертається >{рядок властивості}

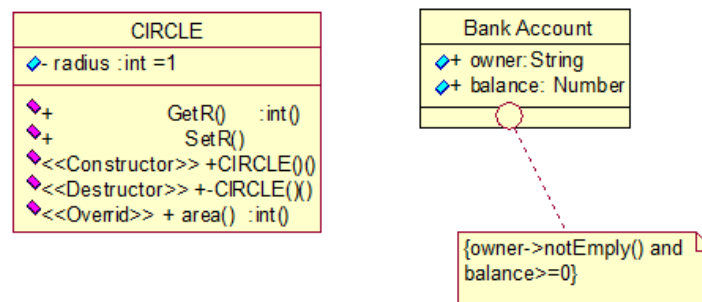


Рисунок 11.3 – Методи та примітки класу

## **Відношення між класами**

Крім внутрішньої будови або структури класів на відповідній діаграмі вказуються відносини між класами. При цьому сукупність типів таких відносин фіксована в мові UML і зумовлена семантикою цих типів відносин.

Базовими відносинами в мові UML є:

- **залежності;**
- **асоціації;**
- **узагальнення.**

**Відношення залежності** графічно зображуються **пунктирною лінією** між відповідними елементами **зі стрілкою**, направленою від класу-клієнту залежності до незалежного класу або класу-джерела.

Для відносин залежності зумовлені ключові слова, які позначають деякі спеціальні його види:

- «*access*» - для позначення доступності відкритих атрибутів і операцій класу-джерела для класів-клієнтів;
- «*bind*» - клас-клієнт може використовувати деякий шаблон для своєї подальшої параметризації;
- «*derive*» - атрибути класу-клієнту можуть бути обчислені по атрибутах класу-джерела;
- «*import*» - відкриті атрибути і операції класу-джерела стають частиною класу-клієнту, так, ніби вони були оголошені безпосередньо в ньому;
- «*refine*» - вказує, що клас-клієнт служить уточненням клас-джерела в силу причин історичного характеру, коли з'являється додаткова інформація в ході роботи над проектом.

**Відношення асоціації** відповідає наявності деяких відносин між класами. Дане відношення позначається суцільно. Лінією з додатковими спеціальними символами, які характеризують окремі властивості конкретної асоціації. В якості додаткових спеціальних символів можуть використовувати ім'я, асоціації, а також імена і кратність класів-ролей асоціації.

**Відношення узагальнення** є звичайним таксономічним відношенням між більш загальним елементом (предком) і більш конкретним або спеціальним елементом (нащадком). Дане відношення може використовуватися для представлення взаємозв'язків між пакетами, класами, варіантами використання та іншими елементами мови UML.

На діаграмах відношення узагальнення позначається **суцільною лінією з трикутною стрілкою на одному із кінців**. Стрілка вказує на загальний клас (клас-предок або супер-клас), а її відсутність – на спеціальний клас (клас-нащадок або підклас).



Browse, за допомогою пункту Browse у головному меню або за допомогою кнопки Class diagram.

#### Рядок інструментів

При активізації діаграми рядок інструментів здобуває наступний вид (рис. 11.6).



Рисунок 11.6 - Рядок інструментів для діаграми класів



- **Class** (клас). Даний інструмент дозволяє створити новий клас у діаграмі й моделі. Клас - це установки структури й шаблону поведінки для деякої безлічі реальних об'єктів, які надалі будуть визначені в програмі на основі даного шаблону. Клас - це деяка абстракція реального миру. Коли ця абстракція ухвалює конкретне втілення, вона називається *об'єктом*.



- **Interface** (інтерфейс). Значок Interface дозволяє створити об'єкт Interface, який указує на видимі ззовні операції класу або компонента. Звичайно інтерфейс створюється тільки для деяких строго певних класів або компонентів і призначений скоріше для логічного відображення системи, але може бути присутнім як на діаграмі класів, так і на діаграмі компонентів.



- **Unidirectional Association** (односпрямований зв'язок). Значок Unidirectional Association дозволяє створити односпрямований зв'язок класу із класом або класу з інтерфейсом. Це загальний і самий слабкий вид зв'язку.



- **Association Class** (асоціація класу). Значок Association Class дозволяє зв'язати класи асоціативним зв'язком. Ця властивість зберігається в класі й для того, щоб його встановити, необхідно створити клас і зв'язати клас реляцією з іншим за допомогою цього значка.



- **Package** (пакет). Значок Package дозволяє створити елемент Package, який використовується для угруповання елементів. Може бути використаний для фізичної або логічного угруповання. У нашій випадку пакет зручніше за все використовувати для фізичного угруповання коду. Але для невеликої системи, якої є тепличне господарство, ми не будемо використовувати пакети.

### **3 Порядок виконання лабораторної частини**

#### **3.1 Оснащення робочого місця**

- методичні вказівки до виконання лабораторної роботи;
- конспект лекцій з дисципліни;
- комп'ютер четвертого покоління і вище з операційною системою Windows;
- програмне середовище Rational Rose.

#### **3.2 Вимоги безпеки при проведенні заняття**

При проведенні лабораторного заняття слід дотримуватися наступних вимог техніки безпеки:

- у комп'ютерному класі знаходитися лише у присутності викладача або лаборанта;
- не вмикати і не вимикати штекер з розетки самостійно;
- під час лабораторного заняття відкривати тільки вікна тих комп'ютерних програм, які стосуються теми поточної роботи.

#### **3.3 Програма виконання лабораторної частини**

##### **3.3.2 Самостійна робота**

Згідно варіанту (лабораторна робота 1) розробити діаграму класів для обраного бізнес-процесу. Виконати її опис.

### **4 Рекомендації щодо оформлення звіту**

Звіт з лабораторної роботи виконується в окремому зошиті у клітинку.

Звіт повинен містити:

- найменування і мету роботи;
- постановка завдання
- діаграма класів бізнес-процесу та її опис;
- висновки по роботі.

### **5 Контрольні запитання**

- 1 Що являє собою діаграма класів?
- 2 Яке призначення діаграми класів?
- 3 Дайте визначення поняттю «клас».
- 4 Зобразіть та поясніть графічне позначення класу в UML.
- 5 Поясніть базові відношення в UML.
- 6 Наведіть графічне зображення відношень в UML.