

Практична робота № 7

СТВОРЕННЯ ІЄРАРХІЇ ОБ'ЄКТІВ ІС В МЕЖАХ ЗАДАНОЇ ПРЕДМЕТНОЇ ГАЛУЗІ. ПОБУДОВА ДІАГРАМ КЛАСІВ.

Мета роботи – вивчення діаграм класів та їх застосування в процесі проектування моделі інформаційної системи.

Завдання

Виділити основні класи об'єктів у системі, що проектується і побудувати діаграму класів, яка у загальному вигляді демонструє архітектуру системи. Побудувати одну-дві діаграми класів, що деталізують окремі підсистеми. Вказати для класів основні атрибути, операції, вид і напрямки асоціацій. Зазначені дії виконати для предметних галузей, що задані у додатку відповідно до індивідуального варіанту завдання студента.

Номер варіанта	Назва предметної галузі
1	Діяльність туристичної фірми
2	Робота інтернет-магазину
3	Автоматизація роботи біржі праці
4	Робота відеотеки
5	Автоматизація роботи бібліотекаря
6	Брокерська система продажу квартир
7	Автоматизована система роботи перукарні
8	Система обміну валют у пункті обміну
9	Система нарахування стипендії студентам
10	Система проходження практики в ІТ-фірмах
11	Організація оренди нерухомості
12	Робота працівника відділу кадрів по роботі із студентами
13	Система роботи ювелірного магазину
14	Система поселення студентів у гуртожиток
15	Система вступу до магістратури
16	Робота фотостудії
17	Робота хімчистки
18	Діяльність відділу реклами
19	Робота інтернет-магазину з продажу побутової техніки
20	Автоматизоване робоче місце юриста
21	Робота кадрового агентства
22	Автоматизоване робоче місце працівника кінотеатру

23	Система резервування і доставки замовлень у ресторані
24	Діяльність підприємства —Будинок музики
25	Аудиторська перевірка організації
26	Робота інтернет-провайдера по обслуговуванню клієнтів
27	Обслуговування клієнтів готелю
28	Діяльність букмекерської контори
29	Робота метеорологічної станції
30	Робота підприємства по розробці програмних продуктів

Контрольні питання

1. Призначення діаграм класів. Для чого використовується діаграма класів на стадії аналізу і на стадії проектування?
2. Назвіть основні компоненти діаграм класів та основні типи статичних зв'язків між класами.
3. Що являє собою асоціація? У чому зміст множинності асоціацій? У чому відмінність атрибутів від асоціацій?
4. Що являє собою операція класу? У чому зміст узагальнення?

Теоретичні відомості

Побудова діаграм класів (class diagrams) є центральною ланкою методології об'єктно-орієнтованого аналізу й проектування. Діаграма класів відображає класи та їх взаємовідносини, тим самим представляючи логічний аспект проекту. Кожна діаграма класів представляє певний ракурс структури класів. На стадії аналізу діаграми класів використовуються, щоб виділити загальні ролі й обов'язки сутностей, які забезпечують необхідну поведінку системи. На стадії проектування діаграми класів застосовують, щоб передати структуру класів, які формують архітектуру системи [1; 2].

Кожен клас має ім'я, яке повинно бути унікальним в проекті. Діаграма класів визначає типи об'єктів системи й різного роду статичні зв'язки, які існують між ними. Є два основних види статичних зв'язків: асоціації (наприклад, менеджер може вести кілька проектів); підтипи (працівник є різновидом особистості).

На діаграмах класів зображуються також атрибути класів, операції та обмеження, які накладаються на зв'язки між об'єктами [4-7]. Типова діаграма класів зображена на рис. 8.4.

Асоціації

Асоціації являють собою зв'язки між екземплярами класів (особистість працює в компанії, компанія має ряд офісів).

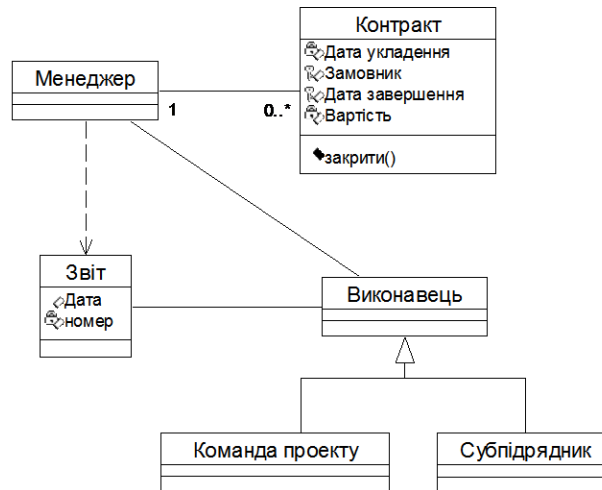


Рис. 8.4. Типова діаграма класів

Будь-яка асоціація володіє двома ролями; кожна роль являє собою напрямок асоціації. Таким чином, асоціація між «Виконавцем» і

«Звітом» містить дві ролі: одна від «Виконавця» до «Звіту»; інша – від «Звіту» до «Виконавця». Роль може бути явно поійменована за допомогою позначки (мітки). Якщо позначка відсутня, ролі привласнюється ім'я класу-мети. Таким чином, роль асоціації від «Виконавця» до «Звіту» може бути названа «Звітом».

Роль має множинність, яка вказує, скільки об'єктів може брати участь у даному зв'язку. На рис. 8.4 символ «0..*» над асоціацією між «Менеджером» й «Контрактом» вказує, що з одним «Менеджером» може бути зв'язано багато «Контрактів», а символ «1» показує, що будь-який «Контракт» управляється одним «Менеджером».

У загальному випадку множинність вказує нижню й верхню границі кількості об'єктів, які можуть брати участь у зв'язку. Для цього можуть використовуватися одиниця, діапазон або дискретна комбінація із чисел і діапазонів.

Для асоціації може бути зазначений напрямок навігації. Якщо навігація зазначена тільки в одному напрямку, то така асоціація називається односпрямованою (рис. 8.4, асоціація між «Менеджером» та «Звітом»). У двонаправленій асоціації навігація зазначена в обох напрямках. У мові UML відсутність стрілок в асоціації трактується наступним чином: напрямок навігації невідомий або асоціація є двонаправленою.

Атрибути

Атрибути багато в чому подібні до асоціацій. Різниця між ними полягає в тому, що атрибути припускають єдиний напрямок навігації – від типу до атрибута.

На рис.8.4 атрибути зазначені для класів «Контракт» і «Звіт». Залежно від ступеня деталізації діаграми, позначення атрибута може включати ім'я атрибута, тип і значення за замовчуванням. У синтаксисі UML це виглядає так: <ознака видимості> <ім'я> : <тип> = <значення за замовчуванням>, де ознака видимості може набувати одне з чотирьох значень:

- загальний (public) - атрибут доступний для всіх клієнтів класу;
- захищений (protected) - атрибут доступний тільки для підкласів і друзів класу;
- секретний (private) - доступний тільки для друзів класу;
- реалізація (implementation) - атрибут доступний тільки усередині пакета, що обрамляє.

Операції

Операції являють собою процеси, реалізовані класом. Найбільш очевидна відповідність існує між операціями й методами класу. Повний синтаксис UML для операцій виглядає так: <ознака видимості> <ім'я> (<список-параметрів>) : <тип виразу, що повертає значення> = <рядок-властивостей>, де:

- ознака видимості може набувати ті ж значення, що й для атрибутів;
- ім'я являє собою символічний рядок;
- список параметрів містить необов'язкові аргументи, синтаксис яких збігається із синтаксисом атрибутів;
- тип виразу, що повертає значення, є необов'язковою специфікацією й залежить від конкретної мови програмування;
- рядок властивостей показує значення властивостей, які застосовуються до даної операції. Прикладом операції на рис. 2.1 є операція закриття класу «Контракт».

Узагальнення

Типовий приклад узагальнення містить «Команду проекту» й «Субпідрядника» (див. рис.1.4). Вони мають деякі розходження, однак у них є також багато спільного. Однакові характеристики можна помістити в узагальнений клас «Виконавець» (супертип), при цьому класи «Команда проекту» й «Субпідрядник» виступають як підтипи.

Зміст узагальнення полягає в тім, що інтерфейс підтипу повинен включати всі елементи інтерфейсу супертипу. Інша сторона узагальнення пов'язана із принципом підстановки. Субпідрядника можна підставити в будь-який код, де потрібен «Виконавець», і при цьому все повинно нормально працювати. Це означає, що, розробивши код, який припускає використання «Виконавця», можна вільно вживати екземпляр будь-якого підтипу «Виконавця». Субпідрядник може реагувати на деякі команди відмінним від

іншого «Виконавця» чином (відповідно до принципу поліморфізму), але ця відмінність не повинна турбувати об'єкт, що викликає «Виконавця».

Узагальнення з погляду реалізації пов'язане з поняттям успадкування в мовах програмування. Підклас успадковує всі методи й поля суперкласу і може перевизначити наслідувані методи. Підтип можна також реалізувати, використовуючи механізм делегування.

Обмеження

При побудові діаграм класів основним завданням є відображення різних обмежень. На рис.8.4 показано, що «Контракт» може управлятися тільки одним «Менеджером».

За допомогою конструкцій асоціації, атрибута й узагальнення можна специфікувати найбільш важливі обмеження, але неможливо виразити їх усі.

В UML відсутній чіткий синтаксис опису обмежень, за винятком поміщення їх у фігурні дужки {}. Описання інструментів для рисування діаграм класів наведено в [4].

Приклади діаграм класів

На рис. 8.5 і 8.6 зображено дві діаграми класів, що реалізують однаковий фрагмент системи «Служба зайнятості в рамках ВНЗ»:

взаємодія користувача з БД, що містить персональні відомості.

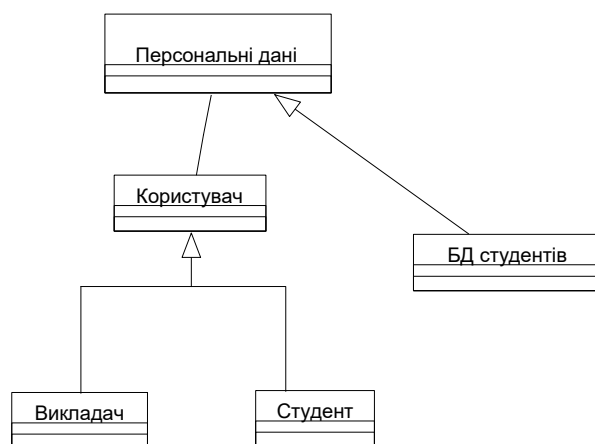


Рис. 8.5. Діаграма класів 1

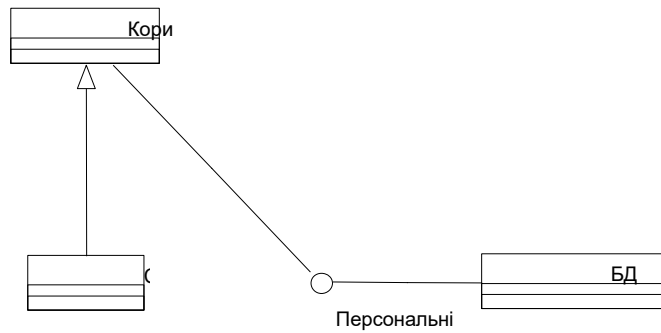


Рис. 8.6. Діаграма класів 2

На діаграмі класів 1 відображений клас «Викладач», який хоча і є нащадком класу «Користувач системи», але не бере участь у взаємодії із БД. Відповідно він ускладнює модель, не вносячи при цьому корисної інформації. На діаграмі 1 інтерфейс «Персональні дані» зображений у розгорнутій формі (як клас-стереотип), а на діаграмі класів 2 він показаний значком інтерфейсу. Остання форма відображення краща, коли не вказуються операції інтерфейсу.