**Практична робота №5 Побудова діаграм класів.**

**Мета:** Ознайомитися з принципами побудови діаграм класів засобами мови UML та застосувати їх на практиці.

**Завдання:**

1. Опрацювати теоретичні відомості. Перевірити засвоєння матеріалу на контрольних запитаннях.
2. Розробити діаграми класів для застосунку, який реалізуємо в ході вивчення ООП:
   1. Створити клас Особа (*Person*) та два похідних від нього Студент (*Student*) та Викладач (*Teacher*). Клас *Student* окрім успадкованих від класу *Person* повинен мати додатково члені-дані: курс, назву групи, три оцінки з предметів математика, фізика, програмування. Клас *Teacher* окрім успадкованих від класу *Person* повинен мати додатково члені-дані: предмет, курси та групи, для якого цей предмет викладається. Селектори і модифікатори можете надати в узагальненому вигляді (без деталізації по членах-даних). Визначте зв’язки між класами.
   2. Для цього прикладу за результатами ЛР №2 повинно було бути визначено як мінімум три Актори: Адміністратор (введення та коригування інформації, формування звітів, підтримка інформаційної бази), Студент та Викладач. Зробіть об’єкти "Звіт по успішності по групах" та "Перелік студентів по предмету викладача". В залежності від визначення ролей встановіть асоціацію з відповідним класом на створеній в п.2.1. діаграмі.
3. Оформити звіт, до якого включити розроблені діаграми та письмові відповіді на запитання в кінці цієї ПР.

Файл зі звітом надавати з іменем у форматі

**ОPI<Номер групи><Номер лекції / практичної / лабораторної>[-<Номер завдання>][літера позначення типу роботи L – лекція, P –практична, R – лабораторна]<Прізвище англійською>**. Наприклад, **ОPI3104L**buts.doc. Відповіді на запитання повинні бути не довгими і змістовними. Не копіюйте фрагментів з різних інформаційних джерел, подумайте і викладіть свою точку зору. При наявності відповідей-"близнюків" відповідь буде зараховуватися першому за часом надсилання.

**Строк виконання цієї роботи ІПЗ-31 19.02.2021, ІПЗ-32 19.02.2021**

## Теоретичні відомості.

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.

Кожен клас повинен мати ім'я і, якщо ім'я занадто довге, його можна скоротити або збільшити сам значок на діаграмі. Ім'я кожного класу повинно бути унікальним в поточному 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. типова діаграма класів зображена на рис. 1.

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



Рис. 1. 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 пойменована 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 також має множинність, яка вказує, скільки об'єктів може брати участь у даному зв'язку. На рис. 1 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 однина, 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. 1, асоціація між «Менеджером» та «Звітом»). У двонапрямленій асоціації навігація зазначена в обох напрямках. У мові UML відсутність стрілок в асоціації трактується наступним чином: напрямок навігації невідомий або асоціація є двонапрямленою.

*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. 1 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 символьний 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.1 є операція закриття класу «Контракт».

*Узагальнення*

Типовий приклад узагальнення містить «Команду проекту» й «Субпідрядника» (див. рис. 1). 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» будуть виступати 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. 1 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.

**Контрольні запитання**.

1. Який основний вид діаграм у концептуальній моделі
2. Яке призначення логічної моделі?
3. Назвіть основний вид діаграм у фізичній моделі.
4. Що являє собою діаграма класів?
5. Яке призначення діаграми класів?
6. Дайте визначення поняттю «клас».
7. Поясніть базові відношення в UML. Що являє собою асоціація? У чому зміст множинності асоціацій? У чому відмінність атрибутів від асоціацій?
8. Що являє собою операція класу? У чому зміст узагальнення?

*Запитання для письмової відповіді.*

1. Які основні типи моделей, на Вашу думку, потрібно використовувати під час проектування? (див. Лекцію 5)? Чи вистачає Вам вивченої інформації по діаграмах для розроблення проекту? Відповідь поясніть.
2. Подумайте, чи доцільніше зробити окремий один клас для зв’язку викладача з групами. Відповідь поясніть.