

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
імені ТАРАСА ШЕВЧЕНКА**



ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра прикладних інформаційних систем

Звіт до лабораторної роботи №2

3 курсу

«Інженерія програмного забезпечення»

студента 2 курсу

групи ПП-22

спеціальності 122 «Комп'ютерні науки»

ОП «Прикладне програмування»

Шевлюк Вікторії Віталіївни

Перевірила:

Доц. Бойко Юлія Петрівна

Київ 2022

Тема: Побудова діаграм класів

Мета роботи: вивчення діаграм класів та їх застосування в процесі проектування.

Хід роботи:

Вправа 2.1. Створення структури моделі за вимогами архітектурного аналізу


- Діаграми класів, що реалізують варіант використання, і діаграми взаємодії, що відображають взаємодію об'єктів в процесі реалізації сценаріїв варіанти використання, поміщаються в кооперацію з ім'ям даного варіанту використання і стереотипом «use-case realization». Все кооперації поміщаються в пакет з ім'ям UseCase Realizations. Зв'язок між варіантом використання і його реалізацією зображується на спеціальній діаграмі трасування.

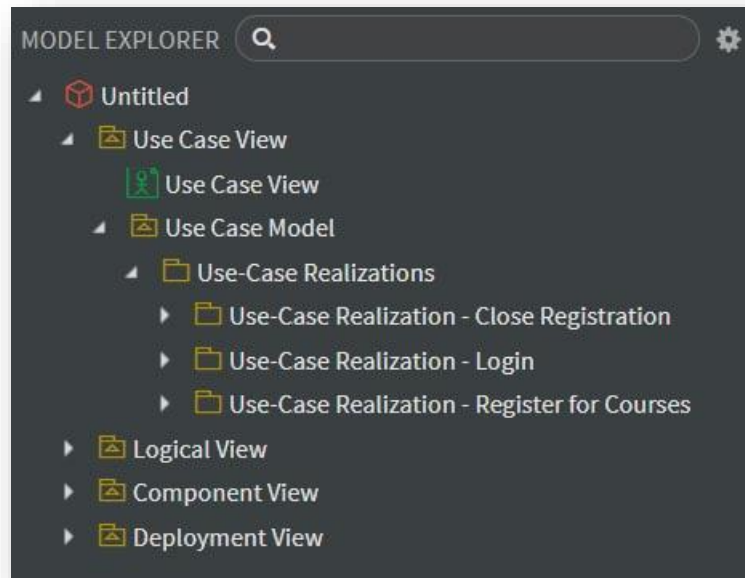
Ідентифікація основних абстракцій полягає в попередньому визначенні набору класів системи (класів аналізу) на основі опису предметної області та специфікації вимог до системи (зокрема, глосарію).

Так, для системи реєстрації ідентифіковано п'ять класів аналізу:

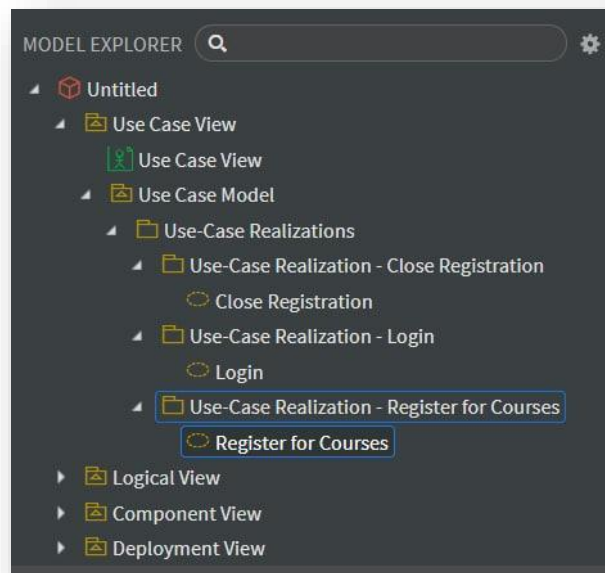
- Student (Студент);
- Professor (Професор);
- Schedule (Навчальний графік);
- Course (Курс);
- Course Offering (Пропонований курс).

Створення пакетів і діаграми трасування:

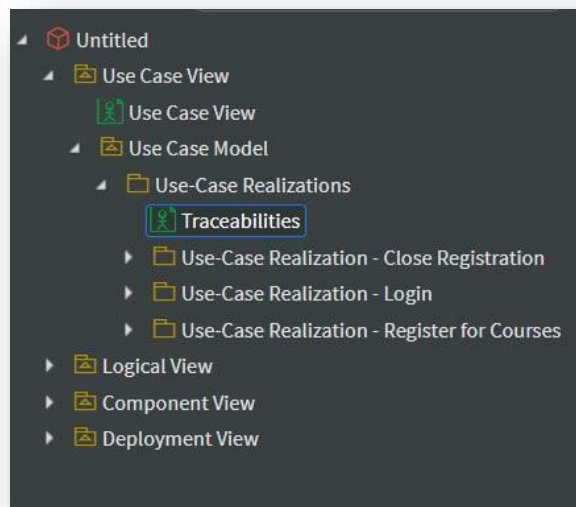
1. В пакеті  Use-Case Model створимо пакет Use-Case Realizations, потім всередині нього - 3 пакети: Use-Case Realization -Close Registration, Use-Case Realization – Login и Use-Case Realization - Register for Courses.




2. У кожному з трьох пакетів типу Use-Case Realization створимо кооперації (варіанти використання, Use Case) з іменами Close Registration, Login і Register for Courses відповідно.



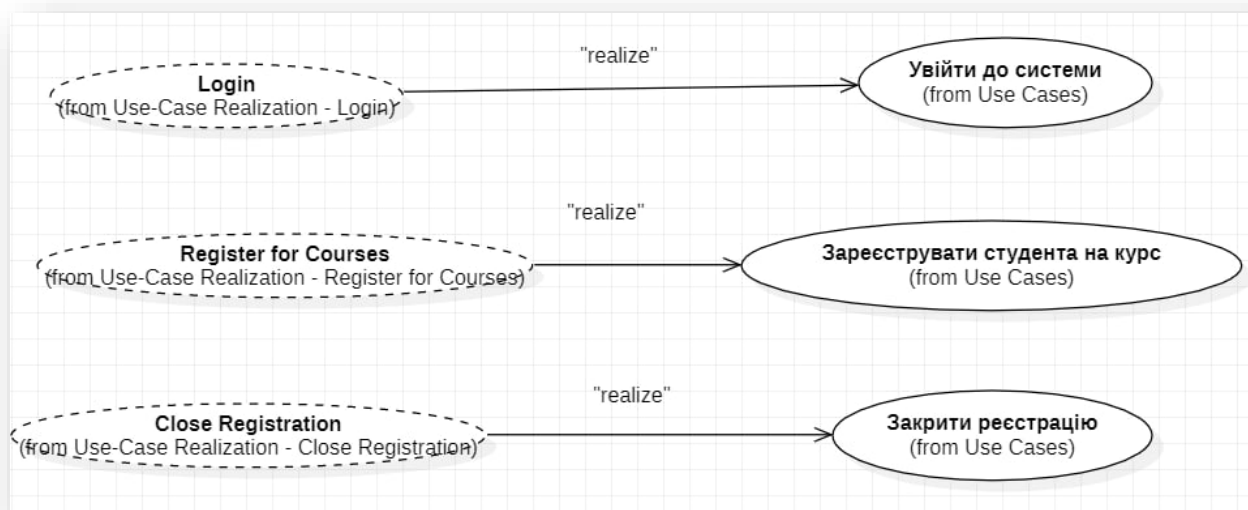
3. Створимо в пакеті Use-Case Realizations нову діаграму варіантів використання з назвою Traceabilities.



4. Перетягнемо у вікно варіанти «Увійти в систему», «Зареєструватися на курси», «Закрити реєстрацію» з пакета Use-Case Model.

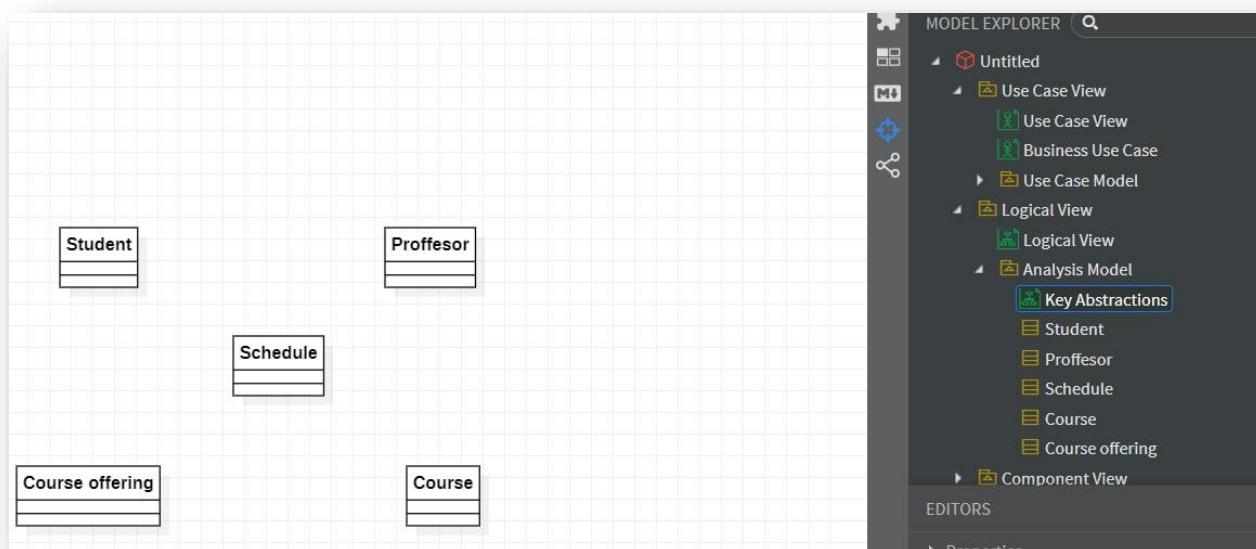
5. За допомогою кнопки  Unidirectional Association (Односпрямована асоціація) панелі інструментів намалюємо асоціацію між варіантами використання.

6. Задаємо асоціаціям через контекстне меню і пункт Open Specification стереотип, що дорівнює realize.





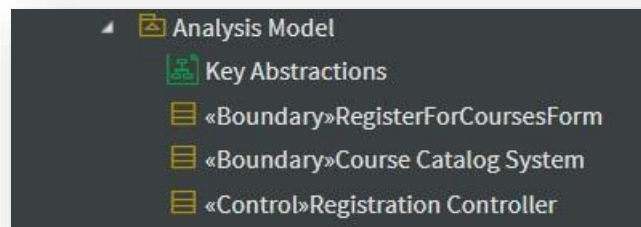
Вправа 2.2. Створення класів за вимогами архітектурного аналізу. Створення класів аналізу і відповідної діаграми Key Abstractions:

1. Клацнемо правою кнопкою миші на пакеті Analysis Model логічного представлення Logical View.
2. Оберемо в меню пункт New/ Class. Новий клас під назвою NewClass з'явиться в браузері.
3. Введемо ім'я Student.
4. Створимо аналогічним чином класи Professor, Schedule, Course і Course Offering.
5. Клацнемо правою кнопкою миші на пакеті Analysis Model.
6. У меню, оберемо пункт New/ Class Diagram.
7. Назвемо нову діаграму класів Key Abstractions.
8. Щоб Розташувати новостворені класи на діаграмі класів Key Abstractions, відкрийте її вікно подвійним клацанням і перетягнути класи на відкриту діаграму мишею.

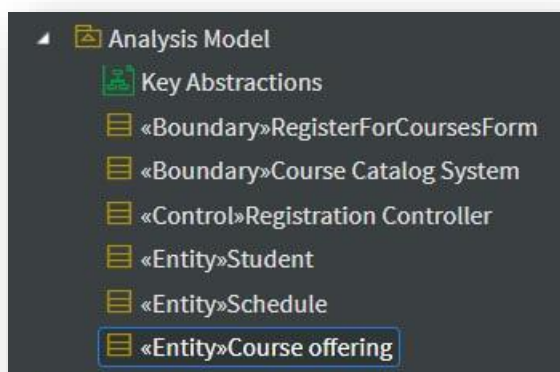




Вправа 2.3. Створення класів, що беруть участь в реалізації варіанту використання Register for Courses, і діаграми класів

1. Клацніть правою кнопкою миші на пакеті  Analysis Model логічного представлення  Logical View.
2. Виберіть в меню пункт New / Class. Новий клас під назвою NewClass з'явиться в браузері.
3. Виділіть його і введіть ім'я RegisterForCoursesForm.
4. Клацніть правою кнопкою миші на класі RegisterForCoursesForm.
5. У меню, виберіть пункт Open Specification.
6. У поле стереотипу виберіть Boundary і натисніть на кнопку ОК.
7. Створіть аналогічним чином класи Course Catalog System зі стереотипом Boundary і Registration Controller зі стереотипом Control.




8. Призначте класів Schedule, CourseOffering і Student стереотип Entity

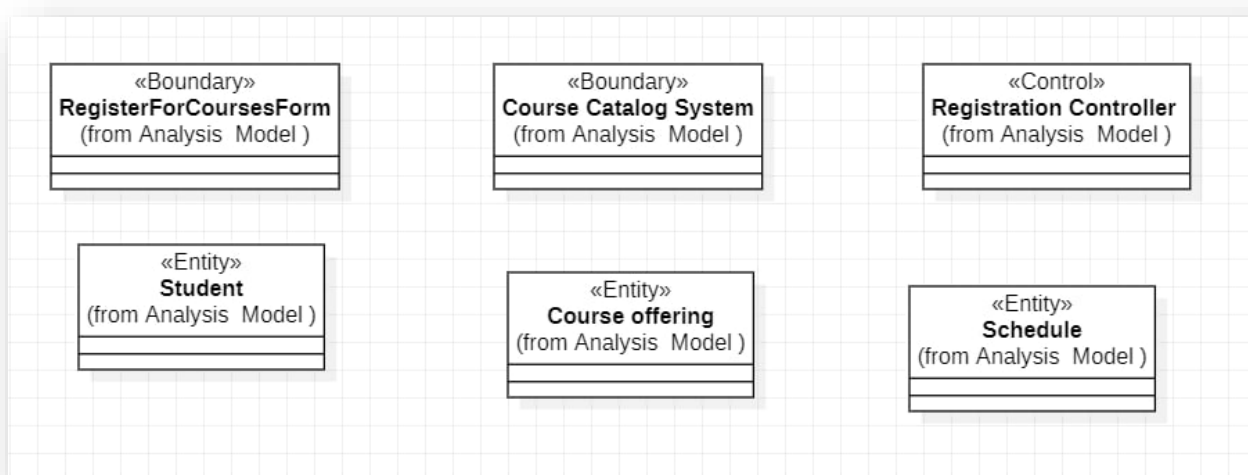


9. Клацніть правою кнопкою миші на кооперації Register for Courses в пакеті  Use-Case Realization - Register for Courses представлення  Use-Case View.

10. У меню, виберіть пункт *New/ Class Diagram*.

11. Назвіть нову діаграму класів  “VOPC (classes only)”.

12. Відкрийте вікно діаграми і перетягніть класи з  Analysis Model на діаграму



Висновок:

В ході даної лабораторної роботи я дослідила діаграми класів та їх застосування в процесі проектування, вивчила набір угод моделювання:

► Імена варіантів використання повинні бути короткими дієслівними фразами.

► Імена класів повинні бути іменниками, відповідними, по можливості, поняттям предметної області.

► Імена класів повинні починатися з великої літери.

► Імена атрибутів і операцій повинні починатися з малої літери.

► Складові імена повинні бути суцільними, без підкреслень, кожне окреме слово має починатися з великої літери.

► Всі класи і діаграми, які описують попередній системний

проект, поміщаються в пакет з ім'ям Analysis Model.

► Діаграми класів, що реалізують варіант використання, і діаграми взаємодії, що відображають взаємодію об'єктів в процесі реалізації сценаріїв варіанти використання, поміщаються в кооперацію з ім'ям даного варіанту використання і стереотипом «use-case realization». Все кооперації поміщаються в пакет з ім'ям UseCase Realizations. Зв'язок між варіантом використання і його реалізацією зображується на спеціальній діаграмі трасування

► Ідентифікація основних абстракцій полягає в попередньому визначенні набору класів системи (класів аналізу) на основі опису предметної області та специфікації вимог до системи (зокрема, глосарію).

Контрольні питання

1. Призначення діаграм класів. Для чого використовується діаграма класів на стадії аналізу і на стадії 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.

2. Назвіть основні компоненти діаграм класів та основні типи статичних зв'язків між класами.

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.

3. Що являє собою асоціація? У чому зміст множинності асоціацій? У чому відмінність атрибутів від асоціацій?

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

Множинність вказує, скільки об'єктів може брати участь у даному зв'язку.

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

4. Що являє собою операція класу? У чому зміст узагальнення?

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.

Зміст узагальнення полягає в тім, що інтерфейс підтипу повинен включати всі елементи інтерфейсу супертипу. Інша сторона узагальнення пов'язана із принципом підстановки.