

Навчальний проект

1. Система реєстрації для ВНЗ. Постановка задачі	2
2. Складання глосарію проекту	3
3. Опис додаткових специфікацій	4
4. Створення моделі варіантів використання	5
5. Аналіз системи	15
5.1. Архітектурний аналіз.....	15
5.2. Аналіз варіантів використання	18
6. Проектування системи	34
6.1. Проектування архітектури	34
6.2. Моделювання розподіленої конфігурації системи.....	38
6.3. Проектування класів	42
6.4. Проектування баз даних.....	48
7. Реалізація системи	52
7.1. Створення компонентів	52
7.2. Генерація коду	53

1. Система реєстрації для ВНЗ. Постановка задачі

Перед керівником інформаційної служби університету поставлено завдання розробки нової клієнт-серверної системи реєстрації студентів замість старої системи на мейнфреймі. Нова система має дозволяти студентам реєструватися на курси та переглядати свої таблиці успішності з персональних комп'ютерів, підключених до локальної мережі університету. Професори повинні мати доступ до онлайн-системи, щоб вказати курси, які вони читатимуть, і проставити оцінки за курси.

Через нестачу коштів університет не в змозі замінити одразу всю систему. Залишається функціонувати у колишньому вигляді база даних, що містить всю інформацію про курси (каталог курсів). Ця база даних підтримується реляційною СУБД. Нова система буде працювати з існуючою БД в режимі доступу без оновлення.

На початку кожного семестру студенти можуть запросити каталог курсів, що містить список курсів, які пропонуються в цьому семестрі. Інформація про кожен курс має включати ім'я професора, найменування кафедри та вимоги до попереднього рівня підготовки (прослуханих курсів).

Нова система має дозволяти студентам вибирати 4 курси у майбутньому семестрі. Додатково кожен студент може вказати 2 альтернативні курси на той випадок, якщо якийсь із вибраних ним курсів виявиться вже заповненим або скасованим. На кожен курс може записатися не більше 10 та не менше 3 студентів (якщо менше 3, то курс буде скасовано). У кожному семестрі існує період, коли студенти можуть змінити свої плани. У цей час студенти повинні мати доступ до системи, щоб додати або видалити вибрані курси. Після того, як процес реєстрації деякого студента завершено, система реєстрації направляє інформацію до розрахункової системи, щоб студент міг внести плату за семестр. Якщо курс виявиться заповненим у процесі реєстрації, студент має бути повідомлено про це до остаточного формування його особистого навчального плану.

Наприкінці семестру студенти повинні мати доступ до системи для перегляду своїх електронних таблиць успішності. Оскільки ця інформація є конфіденційною, система повинна забезпечувати її захист від несанкціонованого доступу.

Професори повинні мати доступ до онлайн-системи, щоб вказати курси, які вони читатимуть, і переглянути список студентів, які записалися на їх курси. Крім того, професори повинні мати можливість проставити оцінки за курси.

2. Складання глосарію проекту

Глосарій призначений для опису термінології предметної галузі. Він може бути використаний як неформальний словник даних системи.

Курс	Навчальний курс, що пропонує університет
Конкретний курс (Course Offering)	Конкретне читання цього курсу у конкретному семестрі (один і той самий курс може вестись у кількох паралельних сесіях). Включає точні дні тижня та час.
Каталог курсів	Повний каталог усіх курсів, які пропонує університет.
Розрахункова система	Система обробки інформації щодо оплати за курси.
Оцінка	Оцінка, одержана студентом за конкретний курс.
Професор	Викладач університету.
Табель успішності (Report Card)	Усі оцінки за всі курси, отримані студентом у цьому семестрі.
Список курсів (Roster)	Список усіх студентів, які записалися на курс.
Студент	Особистість навчання в університеті.
Навчальний графік (Schedule)	Курси, вибрані студентом у поточному семестрі.

3. Опис додаткових специфікацій

Призначення додаткових специфікацій - визначити вимоги до системи реєстрації курсів, які відображені у моделі варіантів використання. Разом вони утворюють повний набір вимог до системи.

Додаткові специфікації визначають нефункціональні вимоги до системи, такі як надійність, зручність використання, продуктивність, супроводжуваність, а також ряд функціональних вимог, що є загальними для декількох варіантів використання.

Функціональні можливості

Система повинна забезпечувати розрахований на багато користувачів режим роботи.

Якщо конкретний курс виявляється заповненим у той час, коли студент формує свій навчальний графік, що включає цей курс, то система повинна сповістити його про це.

Зручність використання

Інтерфейс користувача повинен бути сумісним з Windows 95/98.

Надійність

Система має бути у працездатному стані 24 години на день 7 днів на тиждень, час простою – не більше 10%.

Продуктивність

Система повинна підтримувати до 2000 користувачів, що одночасно працюють з центральною базою даних, і до 500 користувачів, що одночасно працюють з локальними серверами.

Безпека

Система не повинна дозволяти студентам змінювати будь-які навчальні графіки, крім власних, а також не повинна дозволяти професорам модифікувати конкретні курси, обрані іншими професорами.

Тільки професори мають право ставити студентам оцінки.

Тільки реєстратор може змінювати будь-яку інформацію про студентів.

Проектні обмеження

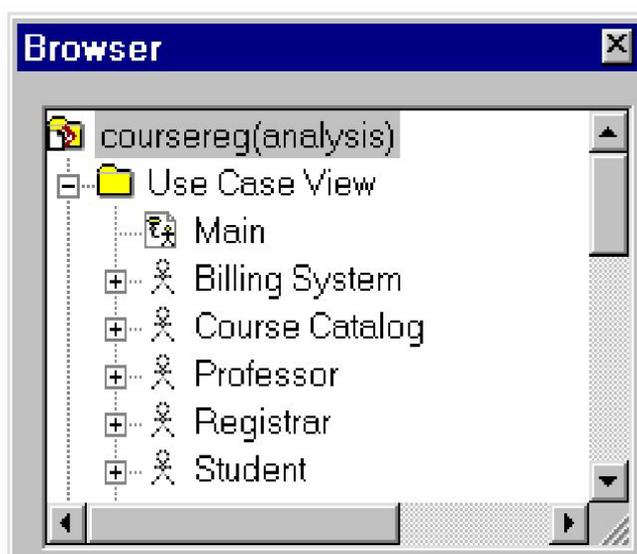
Система має бути інтегрована з існуючою системою каталогу курсів, що функціонує на основі реляційної СУБД.

4. Створення моделі варіантів використання

Діючі особи:

- Student (Студент) – записується на курси;
- Professor (Професор) – вибирає курси для викладання;
- Registrar (Реєстратор) - формує навчальний план та каталог курсів, веде всі дані про курси, професорів та студентів;
- Billing System (Розрахункова система) – отримує від даної системи інформацію щодо оплати за курси;
- Course Catalog (Каталог курсів) - передає в систему інформацію з каталогу курсів, які пропонують університет.

Крок1. Створення дійових осіб у середовищі Rational Rose



Щоб помістити дійову особу у браузер:

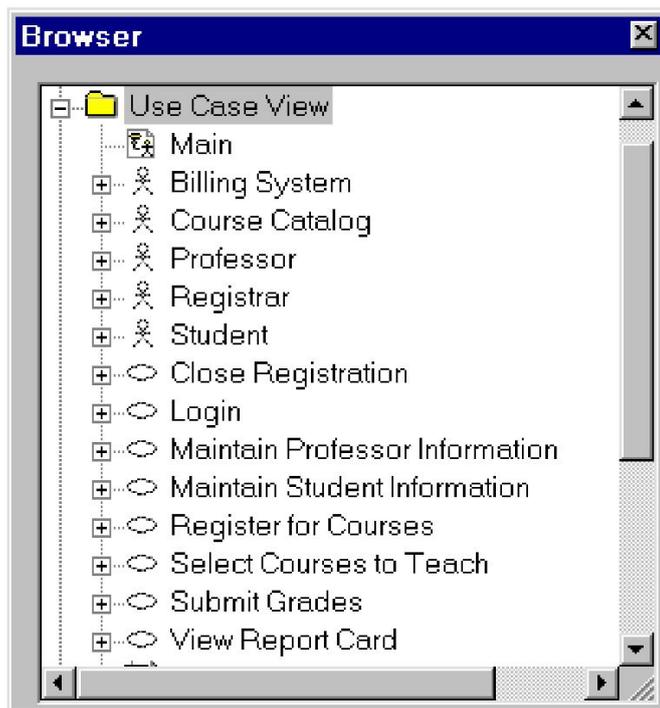
1. Клацніть правою кнопкою миші на пакеті представлення варіантів використання у браузері.
2. Виберіть у меню меню New > Actor
3. У браузері з'явиться нова дійова особа під назвою NewClass. Зліва від його імені ви побачите піктограму дійової особи UML.
4. Виділивши нову дійову особу, введіть її ім'я.
5. Після створення дійових осіб збережіть модель під назвою coursereg(analysis) за допомогою пункту меню File > Save.

Варіанти використання:

Виходячи з потреб дійових осіб, виділяються такі варіанти використання:

- Login (Увійти до системи);
- Register for Courses (Зареєструватися на курси);
- View Report Card (Переглянути таблиць успішності);
- Select Courses to Teach (Вибрати курси для викладання);
- Submit Grades (Проставити оцінки);
- Maintain Professor Information (Вісти інформацію про професорів);
- Maintain Student Information (Вісти інформацію про студентів);
- Close Registration (Закрити реєстрацію).

Крок2. Створення варіантів використання у середовищі Rational Rose

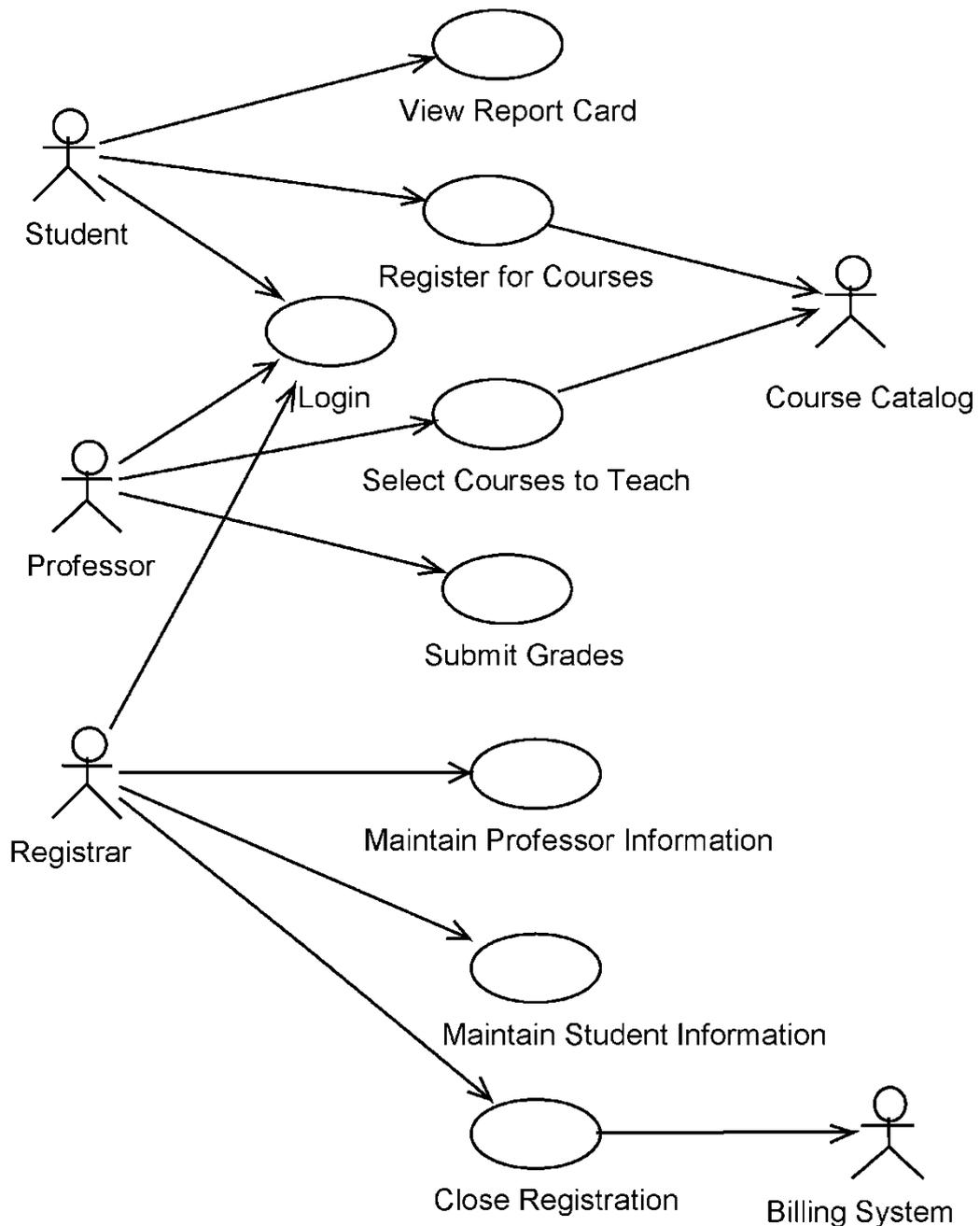


Щоб помістити варіант використання у браузер:

1. Клацніть правою кнопкою миші на пакеті представлення варіантів використання у браузері.
2. Виберіть у меню меню New > Use Case
3. Новий варіант використання під назвою NewUseCase з'явиться у браузері. Зліва від нього буде видно піктограму варіанта використання UML.
4. Виділивши новий варіант використання, введіть назву.

Діаграма варіантів використання:

Створіть діаграму варіантів використання системи реєстрації. Необхідні при цьому дії докладно перелічені далі. Готова діаграма варіантів використання має виглядати як на рис. 1с. 3.1.



Мал. 1. Діаграма варіантів використання системи реєстрації.

У середовищі Rose діаграми варіантів використання створюються у поданні варіантів використання. Головна діаграма (Main) пропонується за замовчуванням. Для моделювання системи можна розробити стільки додаткових діаграм, скільки необхідно.

Щоб отримати доступ до головної діаграми варіантів використання:

1. Поруч із поданням варіантів використання у браузері клацніть на значку «+», це призведе до відкриття цього подання.
2. Двічі клацніть на головній діаграмі Main, щоб відкрити її. Рядок заголовка зміниться, увімкнувши фразу [Use Case Diagram: Use Case view/Main].

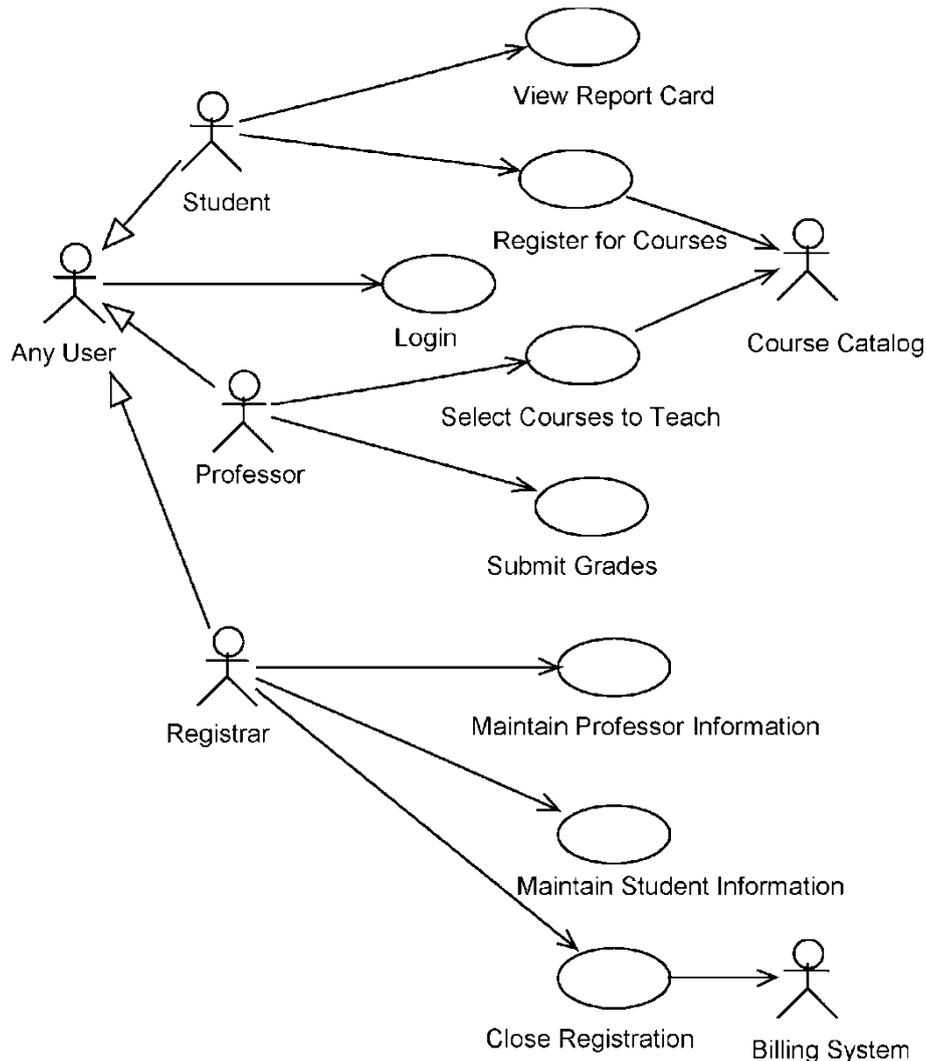
Для створення нової діаграми варіантів використання:

1. Клацніть правою кнопкою миші на пакеті представлення варіантів використання у браузері.
2. Виберіть зі спливаючого меню New > Use Case Diagram.
3. Виділивши нову діаграму, введіть її ім'я.
4. Двічі клацніть назву цієї діаграми в браузері, щоб відкрити її.

Крок3. Побудова діаграми варіантів використання

1. Відкрийте діаграму варіантів використання Main.
2. Щоб помістити дійову особу або варіант використання на діаграму, перетягніть її з браузера на діаграму варіантів використання.
3. За допомогою кнопки Unidirectional Association (Односпрямована асоціація) панелі інструментів намалюйте асоціації між дійовими особами та варіантами використання.

Наявність загального варіанта використання Login для трьох дійових осіб дозволяє узагальнити їхню поведінку та ввести нову дійову особу Any User. Модифікована діаграма варіантів використання показано на рис. 2.



Мал. 2. Модифікована діаграма варіантів використання

Крок 4. Додавання описів до варіантів використання

1. Виділіть у браузері варіант використання "Register for Courses".
2. У вікні документації введіть наступний опис до цього варіанта використання: «Цей спосіб використання можливостей для школярів у школі в поточному semester» (Цей варіант використання дає студенту можливість зареєструватися на курси в поточному семестрі).
3. Створіть за допомогою MS Word три текстові файли з описами варіантів використання Login (Увійти в систему), Register for Courses (Зареєструватися на курси) та Close Registration (Закрити реєстрацію).

Варіант використання Login:

Короткий опис

Цей варіант використання визначає вхід користувача до системи реєстрації курсів.

Основний потік подій

Цей варіант використання починає виконуватися, коли користувач хоче увійти до системи реєстрації курсів.

1. Система запитує ім'я користувача та пароль.
2. Користувач вводить ім'я та пароль.
3. Система перевіряє ім'я та пароль, після чого відкривається доступ до системи.

Альтернативні потоки

Неправильне ім'я/пароль

Якщо під час виконання Основного потоку виявиться, що користувач ввів неправильне ім'я та пароль, система виводить повідомлення про помилку. Користувач може повернутися до початку Основного потоку або відмовитись від входу в систему, при цьому виконання варіанту використання завершується.

Передумови

Немає.

Постуслів'я

Якщо варіант використання виконано успішно, користувач входить до системи. В іншому випадку стан системи не змінюється.

Варіант використання Register for Courses:

Короткий опис

Даний варіант використання дозволяє студенту зареєструватися на конкретні курси у поточному семестрі. Студент може змінити свій вибір (оновити або видалити курси), якщо зміна виконується у встановлений час на початку семестру. Система каталогу курсів надає список усіх конкретних курсів цього семестру.

Основний потік подій

Цей варіант використання починає виконуватися, коли студент хоче зареєструватися на конкретні курси або змінити свій графік курсів.

1. Система запитує потрібну дію (створити графік, оновити графік, видалити графік).
2. Коли студент вказує на дію, виконується один із підлеглих потоків (створити, оновити, видалити або прийняти графік).

Створити графік

1. Система виконує пошук у каталозі курсів доступних конкретних курсів та виводить їх список.
2. Студент обирає зі списку 4 основні курси та 2 альтернативні курси.
3. Після вибору система створює графік студента.

4. Виконується підлеглий потік "Прийняти графік".

Оновити графік

1. Система виводить поточний графік студента.
2. Система виконує пошук у каталозі курсів доступних конкретних курсів та виводить їх список.
3. Студент може оновити свій вибір курсів, видаляючи чи додаючи конкретні курси.
4. Після вибору система оновлює графік.
5. Виконується підлеглий потік "Прийняти графік".

Видалити графік

1. Система виводить поточний графік студента.
2. Система просить у студента підтвердження видалення графіка.
3. Студент підтверджує видалення.
4. Система видаляє графік. Якщо графік включає конкретні курси, на які записався студент, він має бути вилучений зі списків цих курсів.

Прийняти графік

Для кожного обраного, але ще «зафіксованого» конкретного курсу у графіку система перевіряє виконання студентом попередніх вимог (проходження певних курсів), факт відкриття конкретного курсу та відсутність конфліктів графіка. Потім система додає студента до списку обраного курсу. Курс фіксується у графіку та графік зберігається в системі.

Альтернативні потоки

Зберегти графік

У будь-який момент студент може замість ухвалення графіка зберегти його. В цьому випадку крок "Прийняти графік" замінюється на наступний:

1. "Незафіксовані" конкретні курси позначаються у графіку як "вибрані".
2. Графік зберігається у системі.

Не виконані попередні вимоги, курс заповнено або мають місце конфлікти графіка

Якщо під час виконання підпорядкованого потоку «Прийняти графік» система виявить, що студент не виконав необхідні попередні вимоги, чи обраний ним конкретний курс заповнено, чи мають місце конфлікти графіка, видається повідомлення про помилку. Студент може або вибрати інший конкретний курс та продовжити виконання варіанту використання, або зберегти графік, або скасувати операцію, після чого основний потік розпочнеться з початку.

Графік не знайдено

Якщо під час виконання підпорядкованих потоків «Оновити графік» або «Видалити графік» система не може знайти графік студента, видається повідомлення про помилку. Після того, як студент підтвердить це повідомлення, основний потік розпочнеться з початку.

Система каталогу курсів недоступна

Якщо виявиться, що неможливо встановити зв'язок із системою каталогу курсів, буде видано повідомлення про помилку. Після того, як студент підтвердить це повідомлення, варіант використання буде завершено.

Реєстрація на курси закінчено

Якщо на початку виконання варіанта використання виявиться, що реєстрацію на поточний семестр закінчено, буде видано повідомлення і варіант використання завершиться.

Видалення скасовано

Якщо під час виконання підлеглого потоку «Видалити графік» студент вирішить не видаляти його, видалення скасовується, і основний потік розпочнеться з початку.

Передумови

Перед початком виконання цього варіанта використання студент повинен увійти до системи.

Постуслів'я

Якщо параметр використання завершиться успішно, графік студента буде створено, оновлено або видалено. В іншому випадку стан системи не зміниться.

Варіант використання Close Registration:

Короткий опис

Цей варіант використання дозволяє реєстратору закривати процес реєстрації. Конкретні курси, на які не записалося достатньої кількості студентів (менше трьох), скасовуються. У розрахункову систему передається інформація про кожного студента за кожним конкретним курсом, щоб студенти могли внести оплату за курси.

Основний потік подій

Даний варіант використання починає виконуватися, коли реєстратор вимагає припинення реєстрації.

1. Система перевіряє стан процесу реєстрації. Якщо реєстрація ще виконується, видається повідомлення та варіант використання завершується.

2. Для кожного конкретного курсу система перевіряє, чи веде його якийсь професор, і чи записалося на нього не менше трьох студентів. Якщо ці умови виконуються, система фіксує конкретний курс у кожному графіку, який включає цей курс.
3. Для кожного студентського графіка перевіряється наявність у ньому максимальної кількості основних курсів; якщо їх недостатньо, система намагається доповнити альтернативними курсами зі списку графіка. Вибирається перший доступний альтернативний курс. Якщо таких курсів немає, ніякого доповнення не відбувається.
4. Система закриває усі конкретні курси. Якщо в якому-небудь конкретному курсі виявляється менше трьох студентів (з урахуванням додавань, зроблених у п.3), система скасовує його і виключає з кожного графіка, що містить його.
5. Система розраховує плату за навчання для кожного студента у поточному семестрі та надсилає інформацію до розрахункової системи. Розрахункова система надсилає студентам рахунки для оплати з копією їх остаточних графіків.

Альтернативні потоки

Конкретний курс ніхто не веде

Якщо під час виконання основного потоку виявляється, що певний конкретний не ведеться жодним професором, цей курс скасовується. Система виключає даний курс з кожного графіка, що містить його.

Розрахункова система недоступна

Якщо неможливо встановити зв'язок з розрахунковою системою, через деякий час система знову спробує зв'язатися з нею. Спроби повторюватимуться доти, доки зв'язок не встановиться.

Передумови

Перед початком виконання цього варіанта використання реєстратор повинен увійти до системи.

Постуслів'я

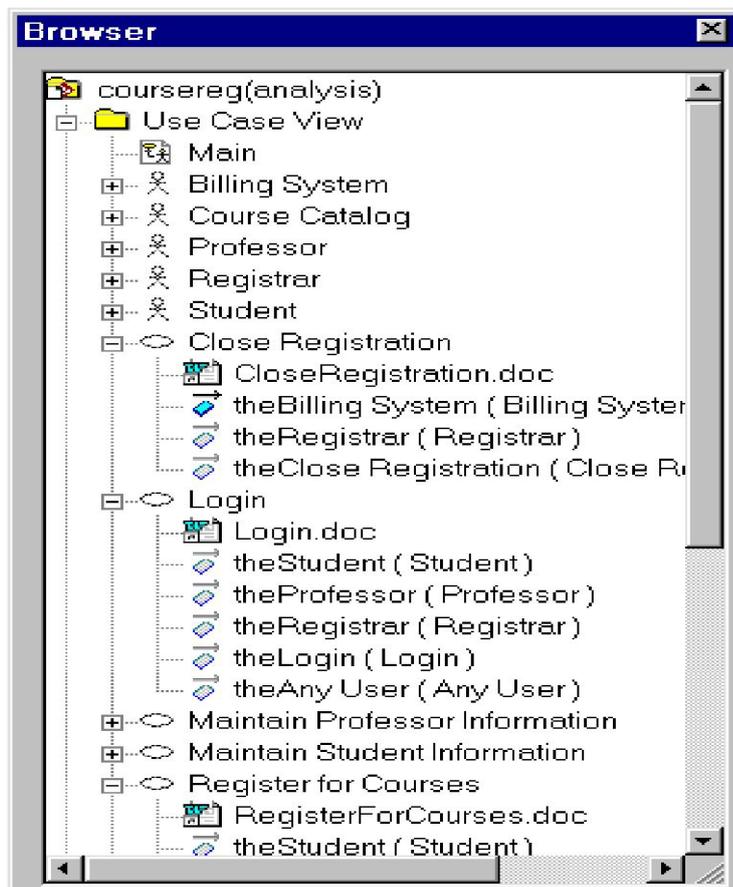
Якщо параметр використання завершиться успішно, реєстрація закривається. В іншому випадку стан системи не зміниться.

Крок5. Прикріплення файлу до варіанта використання

1. Клацніть правою кнопкою миші на варіанті використання.
2. У меню виберіть пункт Open Specification
3. Перейдіть на вкладку файлів.
4. Клацніть правою кнопкою миші на білому полі і з меню виберіть пункт Insert File.

5. Вкажіть створений раніше файл і натисніть кнопку Open, щоб прикріпити файл до варіанта використання

В результаті подання варіантів використання в браузері набуде наступного вигляду:



Видалення варіантів використання та дійових осіб

Існує два способи видалити елемент моделі – з однієї діаграми або з усієї моделі.

Щоб видалити елемент моделі з діаграми:

1. Виділіть елемент на діаграмі.
2. Натисніть клавішу Delete.
3. Зверніть увагу, що хоча елемент і видалений з діаграми, він залишився в браузері та на інших діаграмах системи.

Щоб видалити елемент із моделі:

1. Виділіть елемент на діаграмі.
2. Виберіть пункт Edit > Delete from Model або натисніть клавіші CTRL + D.

5. Аналіз системи

5.1. Архітектурний аналіз

Ухвалення угод з моделювання Включає:

- Використовувані діаграми та елементи моделі;
- правила їх застосування;
- Угоди щодо іменування елементів;
- Організація моделі (пакети).

Приклад угод моделювання:

- Імена варіантів використання мають бути короткими дієслівними фразами.
- Для кожного варіанта використання має бути створений пакет Use-Case Realization, що включає:
 - принаймні одну реалізацію варіанта використання;
 - діаграму "View Of Participating Classes" (VOPC).
- Імена класів мають бути іменниками, відповідними, по можливості, поняттям предметної галузі.
- Імена класів повинні починатися з великої літери.
- Імена атрибутів та операцій повинні починатися з малої літери.
- Складові імена мають бути суцільними, без підкреслень, кожне окреме слово має починатися з великої літери.

Реалізація варіанта використання (Use-Case Realization):

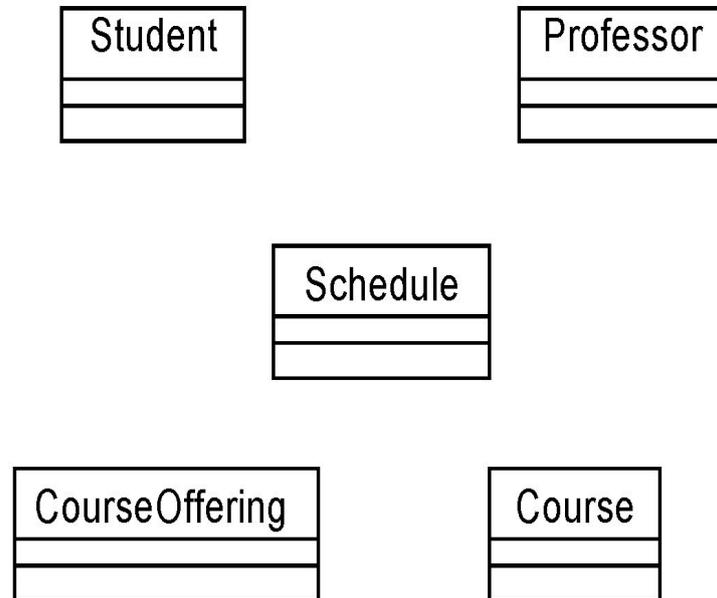
Описує реалізацію конкретного варіанта використання в термінах взаємодіючих об'єктів і представляється за допомогою набору діаграм (діаграм класів, що реалізують варіант використання, та діаграм взаємодії (діаграм послідовності та кооперативних діаграм), що відображають взаємодію об'єктів у процесі реалізації варіанта використання).



Мал. 3. Реалізація варіанта використання

Ідентифікація ключових абстракцій

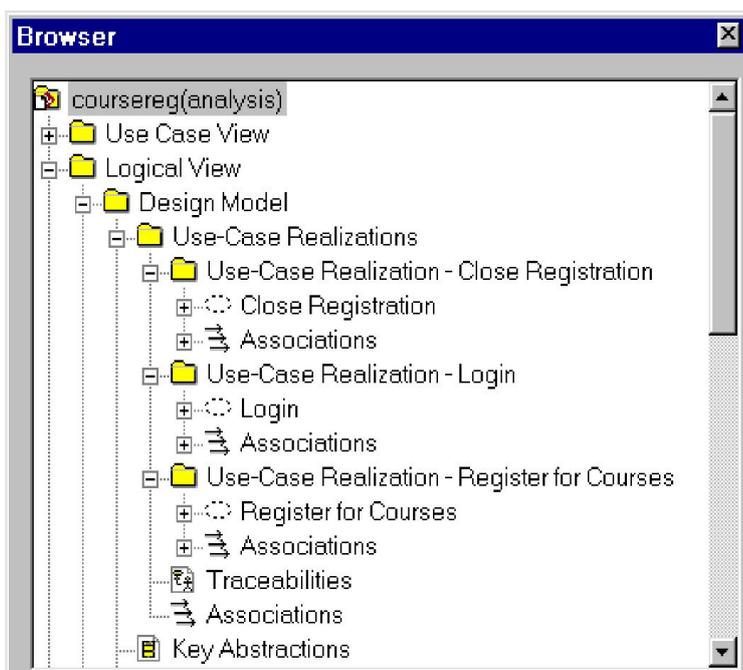
Полягає у попередньому визначенні класів системи (класів аналізу). Джерела – знання предметної галузі, вимоги до системи, глосарій. Класи аналізу системи реєстрації показано на рис. 4:



Мал. 4. Класи аналізу системи реєстрації

Крок6. Створення структури моделі та класів аналізу в відповідно до вимог архітектурного аналізу

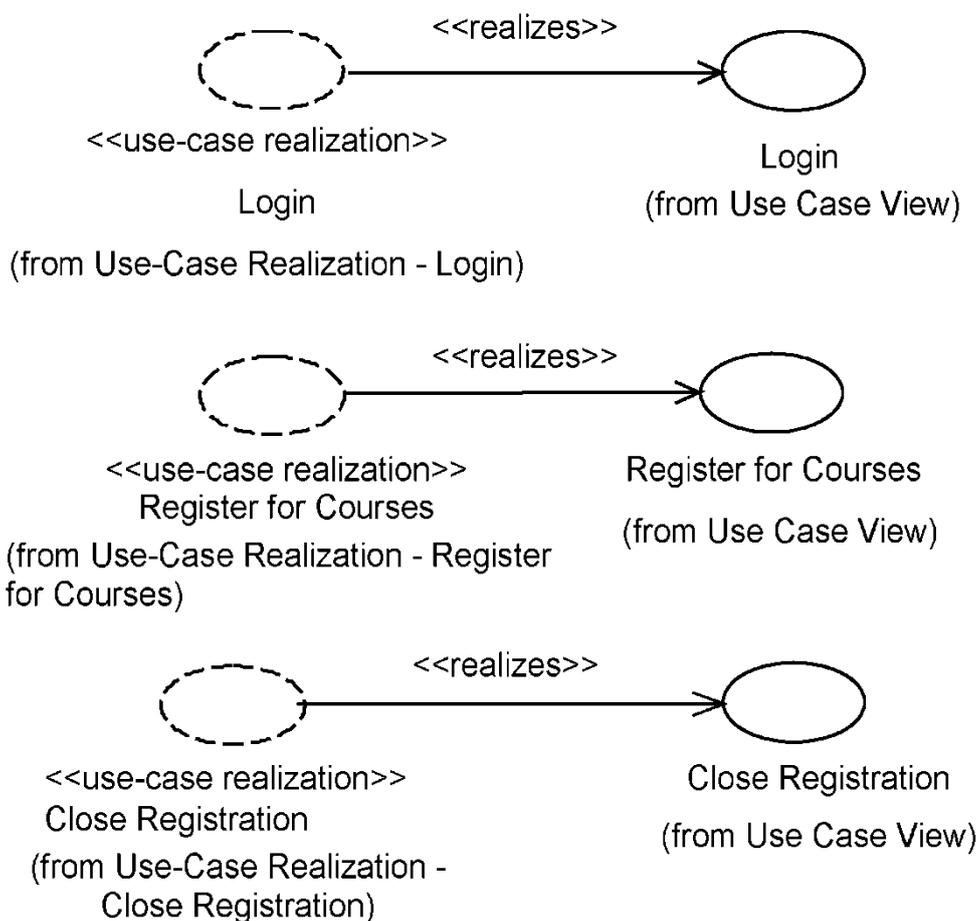
Структура логічного представлення браузера повинна мати такий вигляд:



Створення пакетів та діаграми Traceabilities:

1. Клацніть правою кнопкою миші на логічному уявленні браузера.
2. У меню виберіть New > Package

3. Назвіть новий пакет Design Model.
4. Створіть аналогічно пакети Use-Case Realizations, Use-Case Realization - Close Registration, Use-Case Realization - Login і Use-Case Realization - Register for Courses.
5. У кожному з пакетів типу Use-Case Realization створіть відповідні кооперації Close Registration, Login і Register for Courses (кожна кооперація є варіантом використання зі стереотипом «use-case realization», який задається у специфікації варіанта використання).
6. Створіть у пакеті Use-Case Realizations нову діаграму варіантів використання з назвою Traceabilities та побудуйте її відповідно до рис. 5.



Мал. 5. Діаграма Traceabilities

Створення класів аналізу та відповідної діаграми Key Abstractions:

1. Клацніть правою кнопкою миші на пакеті Design Model.
2. Виберіть у меню меню New > Class. Новий клас під назвою NewClass з'явиться у браузері.
3. Виділіть його та введіть ім'я Student.
4. Створіть аналогічно класи Professor, Schedule, Course і CourseOffering.
5. Клацніть правою кнопкою миші на пакеті Design Model.
6. У меню виберіть New > Class Diagram.

7. Назвіть нову діаграму класів Key Abstractions.
8. Щоб розташувати новостворені класи на діаграмі класів, відкрийте її та перетягніть класи на відкриту діаграму мишею. Діаграма класів має виглядати як на рис. 4.

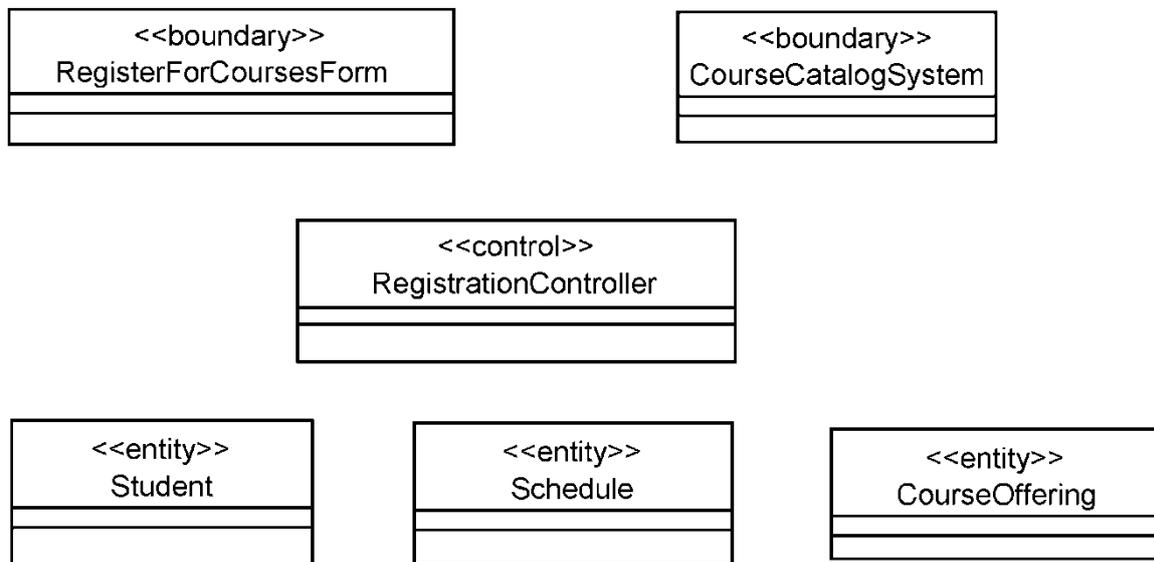
5.2. Аналіз варіантів використання

Ідентифікація класів, що беруть участь у реалізації потоків подій варіанта використання

У потоках подій варіанти використання виявляються класи трьох типів:

1. **Граничні класи (Boundary)**- служать посередниками при взаємодії зовнішніх об'єктів із системою. Як правило, для кожної пари «діюча особа – варіант використання» визначається один граничний клас. Типи граничних класів: інтерфейс користувача (обмін інформацією з користувачем, без деталей інтерфейсу - кнопок, списків, вікон), системний інтерфейс і апаратний інтерфейс (протоколи, що використовуються, без деталей їх реалізації).
2. **Класи-сутності (Entity)**- являють собою ключові абстракції (поняття) системи, що розробляється. Джерела виявлення класів-сутностей: ключові абстракції, створені у процесі архітектурного аналізу, глосарій, опис потоків подій варіантів використання.
3. Керуючі класи (Control) - забезпечують координацію поведінки об'єктів у системі. Можуть бути відсутніми в деяких варіантах використання, що обмежуються простими маніпуляціями зі збереженими даними. Як правило, для кожного варіанта використання визначається один клас, що управляє. Приклади класів, що управляють: менеджер транзакцій, координатор ресурсів, обробник помилок.

Приклад набору класів, що у реалізації варіанти використання Register for Courses, наведено на рис. 3.6.



Мал. 6. Класи, що беруть участь у реалізації варіанта використання Register for Courses

Крок 7. Створення класів, що беруть участь у реалізації варіанта використання Register for Courses, та діаграми класів «View Of Participating Classes» (VOPC)

1. Клацніть правою кнопкою миші на пакеті Design Model.
2. Виберіть у меню меню New > Class. Новий клас під назвою NewClass з'явиться у браузері.
3. Виділіть його та введіть ім'я RegisterForCoursesForm.
4. Клацніть правою кнопкою миші на класі RegisterForCoursesForm.
5. У меню виберіть пункт Open Specification.
6. У полі стереотипу виберіть пункт Boundary і натисніть кнопку ОК.
7. Створіть аналогічно класи CourseCatalogSystem зі стереотипом Boundary та RegistrationController зі стереотипом Control.
8. Призначте класам Schedule, CourseOffering та Student стереотип Entity.
9. Клацніть правою кнопкою миші на кооперації Register for Courses у пакеті Use-Case Realization - Register for Courses.
10. У меню виберіть New > Class Diagram.
11. Назвіть нову діаграму класів VOPC (classes only).
12. Відкрийте її та перетягніть класи на відкриту діаграму відповідно до рис. 6.

Розподіл поведінки, що реалізується варіантом використання, між класами

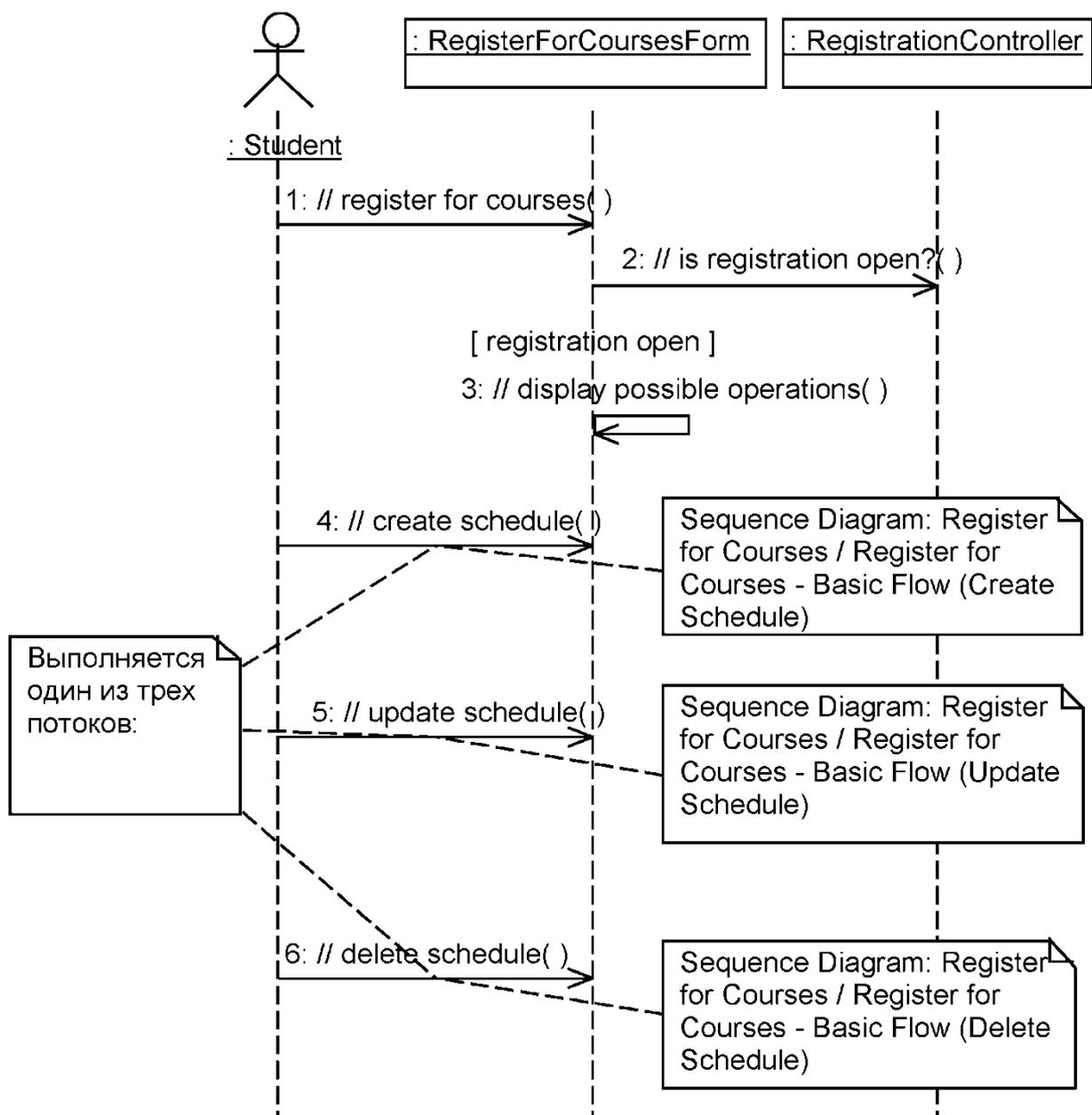
Реалізується за допомогою діаграм взаємодії (діаграм послідовності та кооперативних діаграм). Насамперед будується діаграма (одна чи більше), що описує основний потік подій та його підлеглі потоки. До кожного альтернативного потоку подій будується окрема діаграма. Приклади:

- обробка помилок;
- контроль часу виконання;
- обробка неправильних даних, що вводяться.

Недоцільно описувати тривіальні потоки подій (наприклад, у потоці бере участь лише один об'єкт).

Крок8. Створення діаграм взаємодії

Створимо діаграми послідовності та кооперативні діаграми для основного потоку подій варіанта використання Register for Courses. Готові діаграми послідовності повинні мати вигляд як на рис. 7 – 11.



Мал. 7. Діаграма послідовності Register for Courses - Basic Flow

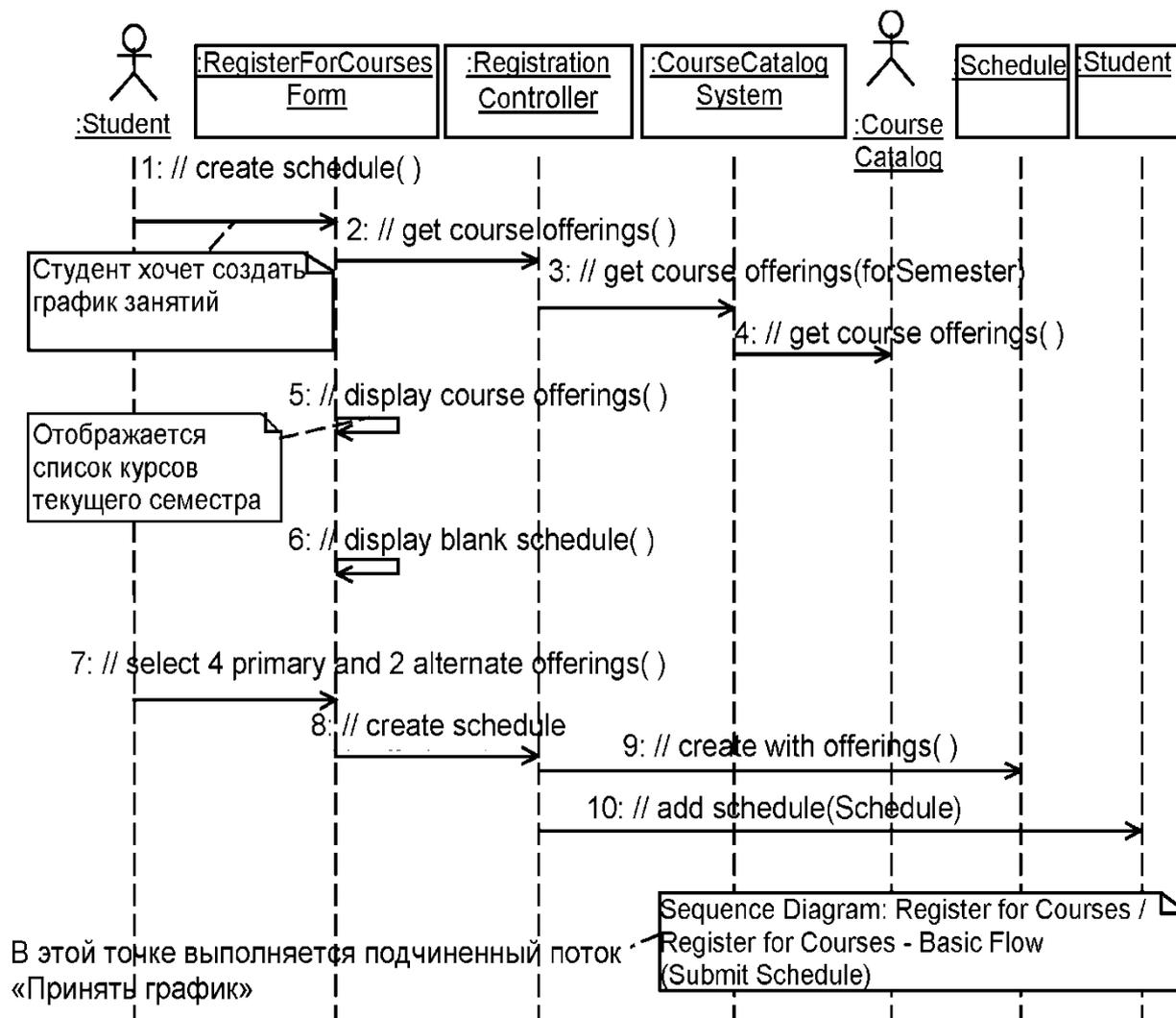


Рис.8. Діаграма послідовності Register for Courses - Basic Flow (Create Schedule)

Налаштування

1. У меню моделі виберіть Tools > Options.
2. Перейдіть на вкладку діаграм.
3. Контрольні перемикачі Sequence Numbering, Collaboration Numbering повинні бути позначені, а Focus of Control – ні.
4. Натисніть кнопку ОК, щоб вийти з вікна параметрів.

Створення діаграми послідовності

1. Клацніть правою кнопкою миші на кооперації Register for Courses у пакеті Use-Case Realization - Register for Courses.
2. У меню виберіть New > Sequence Diagram.
3. Назвіть нову діаграму Register for Courses – Basic Flow.
4. Двічі клацніть на ній, щоб відкрити її.

Додавання на діаграму дійової особи, об'єктів та повідомлень

1. Перетягніть дійову особу Student із браузера на діаграму.
2. Перетягніть класи RegisterForCoursesForm і RegistrationController з браузера на діаграму.
3. Натисніть кнопку Object Message (Повідомлення об'єкта) на панелі інструментів.
4. Проведіть мишею від лінії життя дійової особи Student до лінії життя об'єкта RegisterForCoursesForm.
5. Виділивши повідомлення, введіть ім'я: // register for courses.
6. Повторіть дії 3 - 5, щоб помістити на діаграму решту повідомлень, як показано на рис. 7 (для рефлексивного повідомлення з використовується кнопка Message to Self).

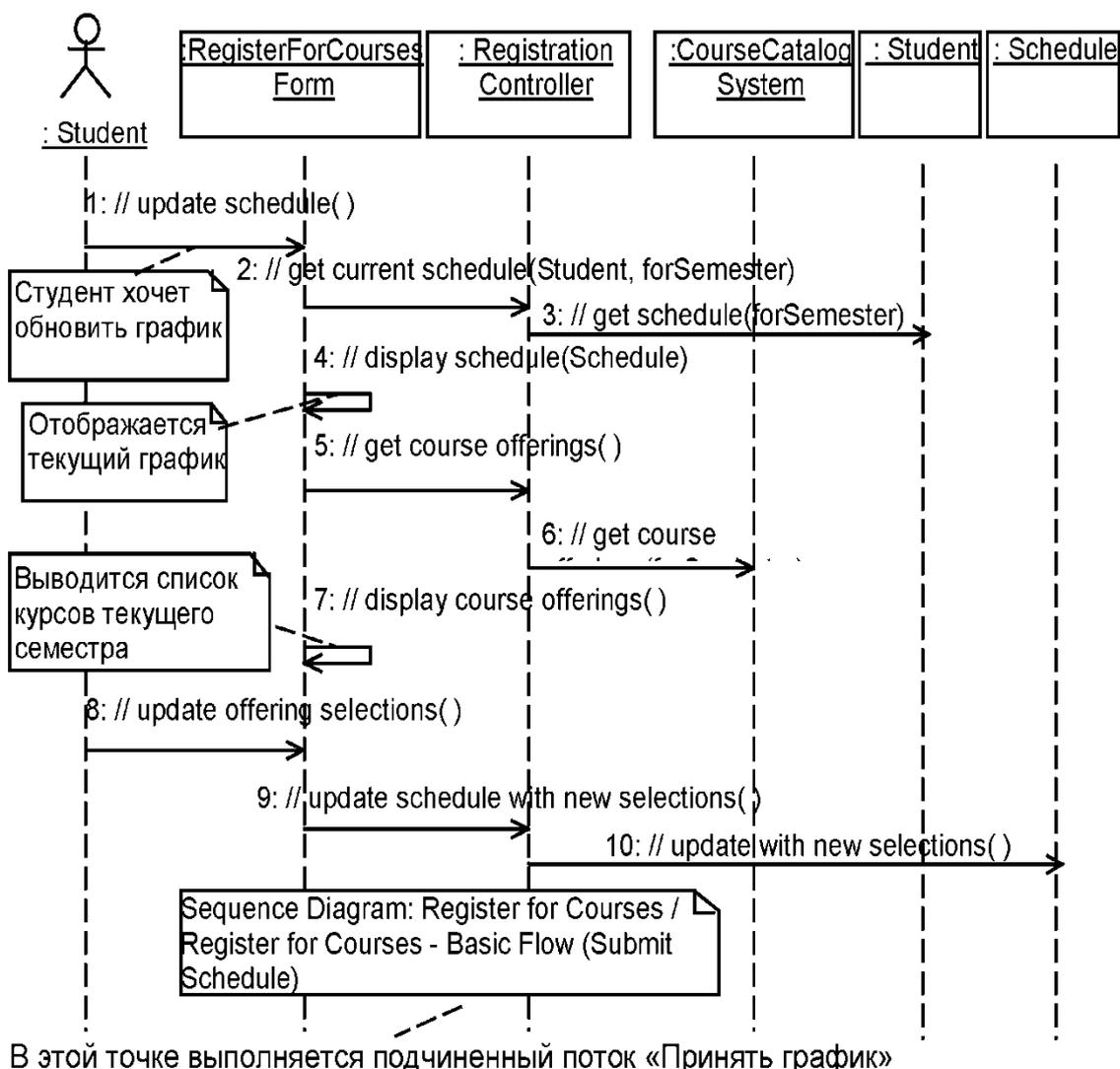
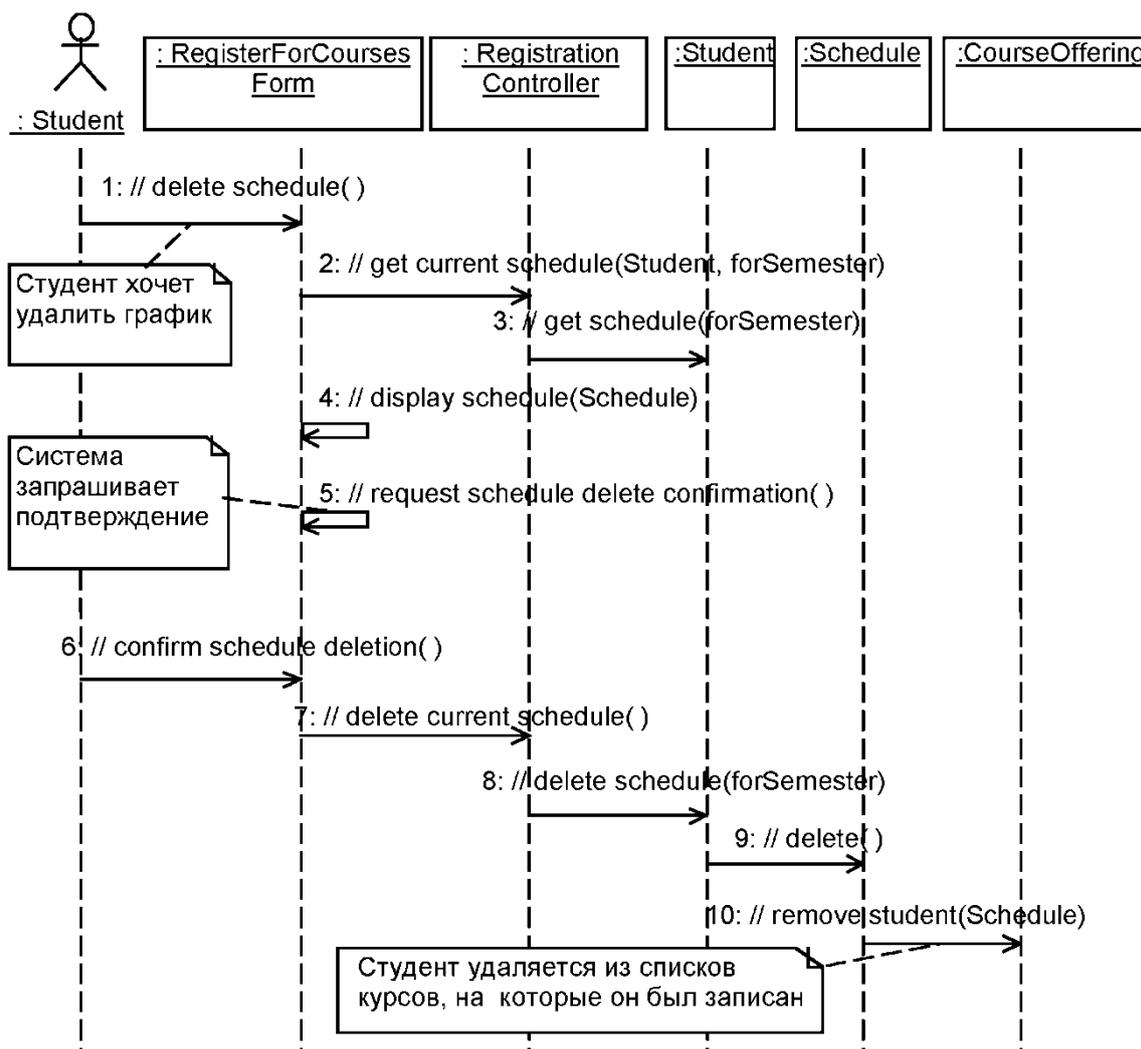


Рис..9. Діаграма послідовності Register for Courses - Basic Flow (Update Schedule)

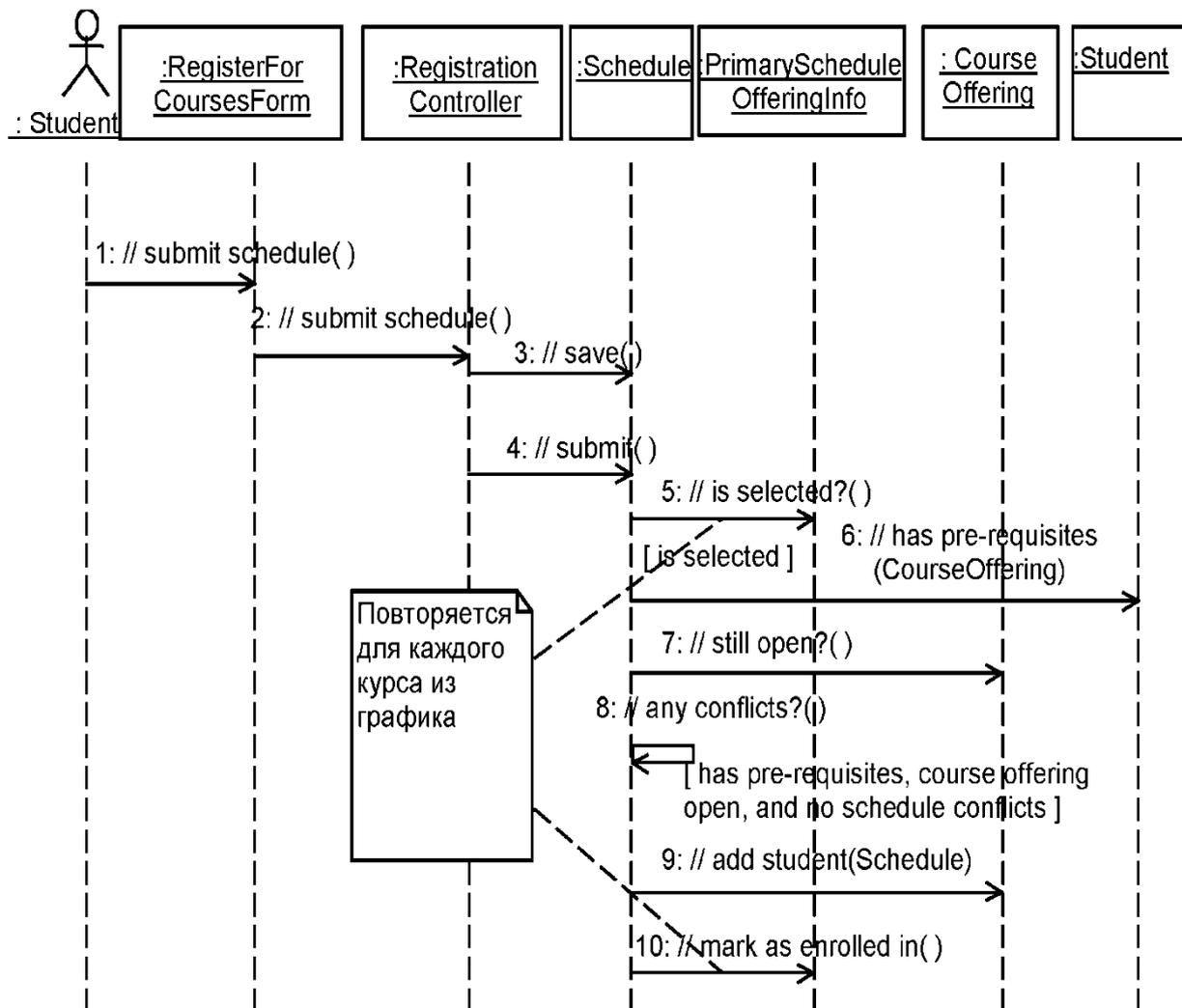


Мал. 10. Діаграма послідовності Register for Courses - Basic Flow (Delete Schedule)

Співвідношення повідомлень з операціями

1. Клацніть правою кнопкою на повідомленні 1 // Register for courses.
2. У меню виберіть <new operation>. Відобразиться вікно специфікації операції.
3. У полі імені залиште ім'я повідомлення // register for courses.
4. Натисніть кнопку ОК, щоб закрити вікно специфікації операції та повернутися на діаграму.
5. Повторіть дії 1 - 4, доки не співвіднесете з операціями всі інші повідомлення.

Виконайте аналогічні дії для створення діаграм послідовності, показаних на рис. 8 – 11. Зверніть увагу, що на діаграмі рис. 11 з'явився об'єкт нового класу PrimarySheduleOfferingInfo (класу асоціацій, що описує зв'язок між класами Shedule та OfferingInfo), який потрібно попередньо створити.



Мал. 11. Діаграма послідовності Register for Courses - Basic Flow (Submit Schedule)

Створення приміток

Щоб помістити на діаграму примітку:

1. Натисніть кнопку Note на панелі інструментів.
2. Клацніть мишею там діаграми, куди збираєтеся помістити примітку.
3. Виділивши нову примітку, введіть текст.
4. Щоб прикріпити примітку до елемента діаграми, на панелі інструментів натисніть кнопку Anchor Notes To Item (Прикріпити примітки до елемента).
5. Натиснувши ліву кнопку миші, проведіть покажчик від примітки до елемента діаграми, з яким воно буде пов'язане. Між приміткою та елементом виникне штрихова лінія.
6. Щоб створити примітку-посилання на іншу діаграму (як це зроблено на діаграмі мал. 7 та інших), створіть порожню примітку (без тексту) та перетягніть на неї з браузера потрібну діаграму.

Крім приміток, на діаграму можна помістити також текстову область. З її допомогою можна, наприклад, додати до діаграми заголовок.

Щоб помістити текстову область на діаграму:

1. На панелі керування натисніть кнопку Text Box.
2. Клацніть в діаграмі, щоб помістити туди текстову область.
3. Виділивши цю область, введіть текст.

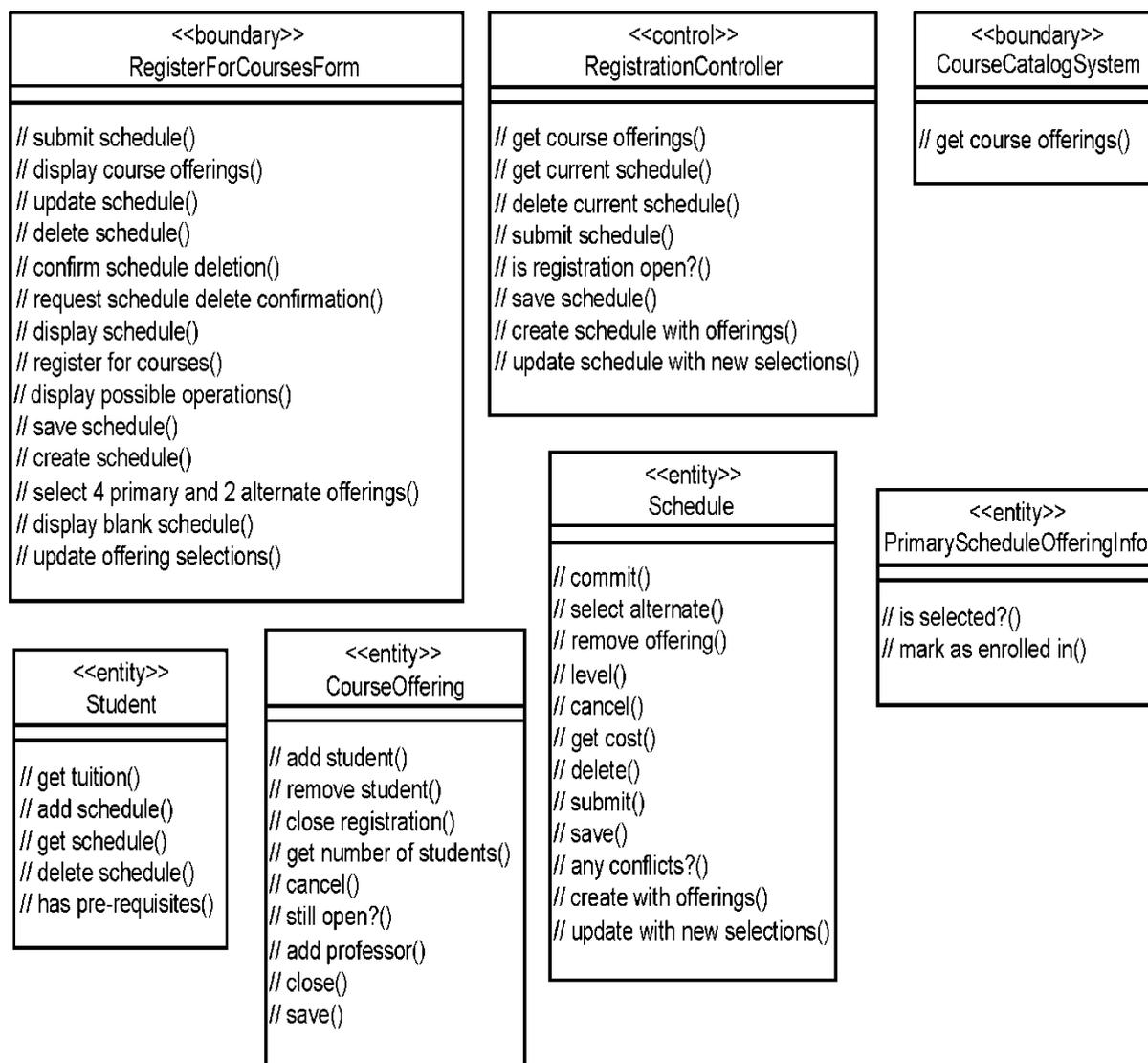
Створення кооперативної діаграми

Для створення кооперативної діаграми достатньо відкрити діаграму послідовності та натиснути клавішу F5.

Визначення обов'язків (responsibilities), атрибутів та асоціацій класів

Обов'язок (responsibility) - дія, яку об'єкт зобов'язаний виконувати на запит інших об'єктів. Обов'язок перетворюється на одну або більше операцій класу на етапі проектування. Обов'язки визначаються, з повідомлень на діаграмах взаємодії, і документуються у класах як операцій «аналізу», які з'являються там автоматично у процесі побудови діаграм взаємодії (співвіднесення повідомлень з операціями).

Так, діаграма класів VOPC (classes only) (рис. 6) після побудови діаграм взаємодії у вправі 8 має набути вигляду, зображеного на рис. 12.



Мал. 12. Діаграма класів VOPC (classes only) із операціями «аналізу»

Атрибути класів аналізу визначаються, виходячи зі знань про предметну область, вимог до системи та глосарію.

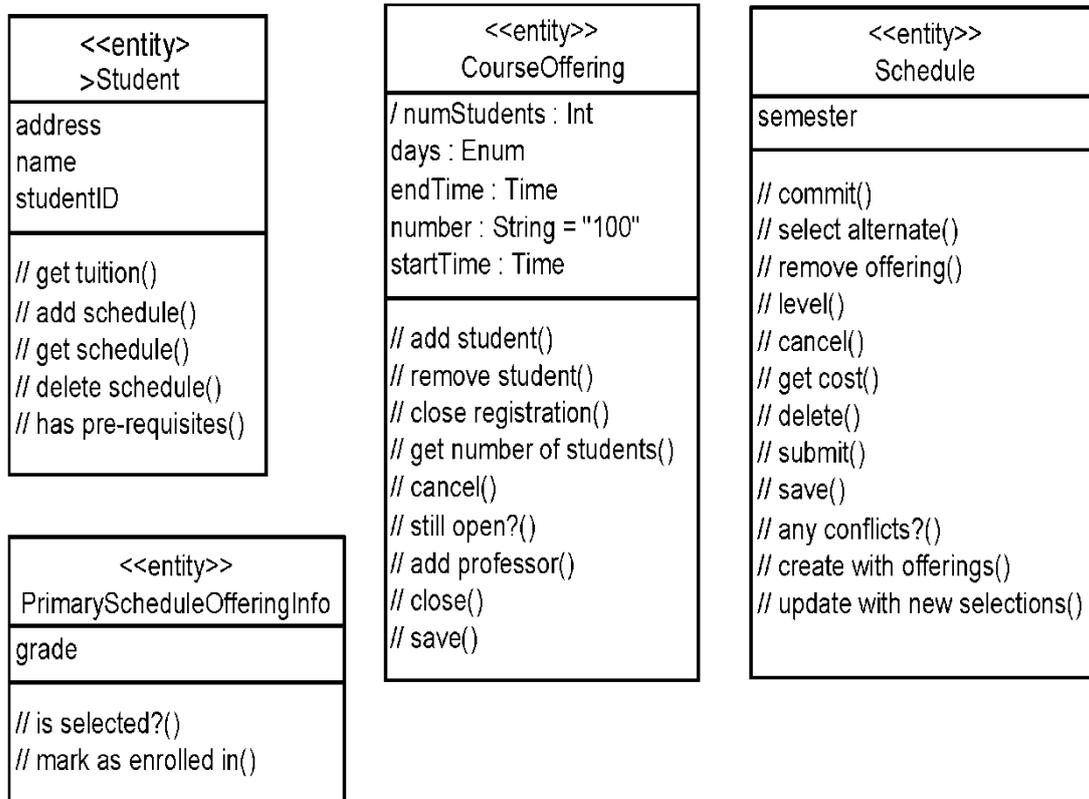
Крок9. Додавання атрибутів до класів Налаштування

1. У меню моделі виберіть Tools > Options.
2. Перейдіть на вкладку Diagram.
3. Переконайтеся, що перемикач Show All Attributes позначений.
4. Переконайтеся, що перемикачі Suppress Attributes та Suppress Operations не позначені.

Додавання атрибутів

1. Клацніть правою кнопкою миші на класі Student.
2. У меню виберіть New Attribute.
3. Введіть новий атрибут address
4. Натисніть клавішу Enter.

5. Повторіть кроки 1 - 4, додавши атрибути name та studentID.
6. Додайте атрибути до класів CourseOffering, Shedule та PrimaryScheduleOfferingInfo, як показано на рис. 13.



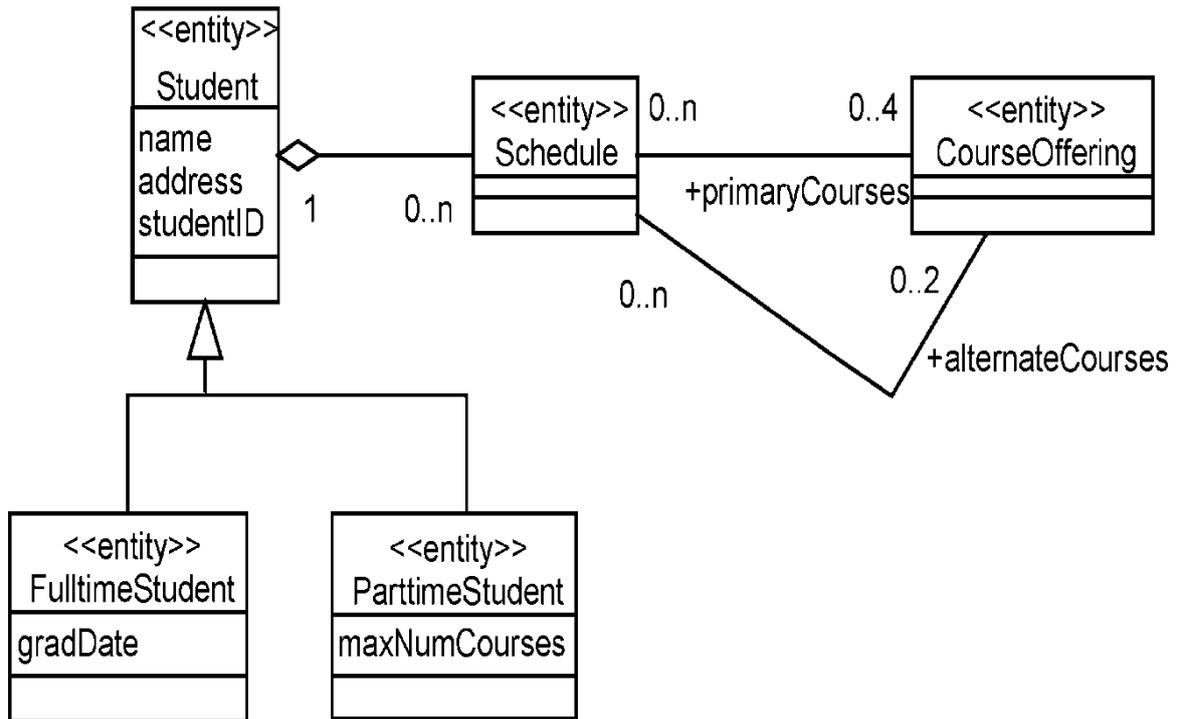
Мал. 13. Класи з операціями «аналізу» та атрибутами

Зв'язки між класами (асоціації) визначаються з урахуванням діаграм взаємодії. Якщо два об'єкти взаємодіють (обмінюються повідомленнями), між ними має існувати зв'язок (шлях взаємодії). Для асоціацій задаються множинність і, можливо, напрямок навігації. Можуть використовуватися численні асоціації, агрегації та класи асоціацій.

Крок10. Додавання зв'язків

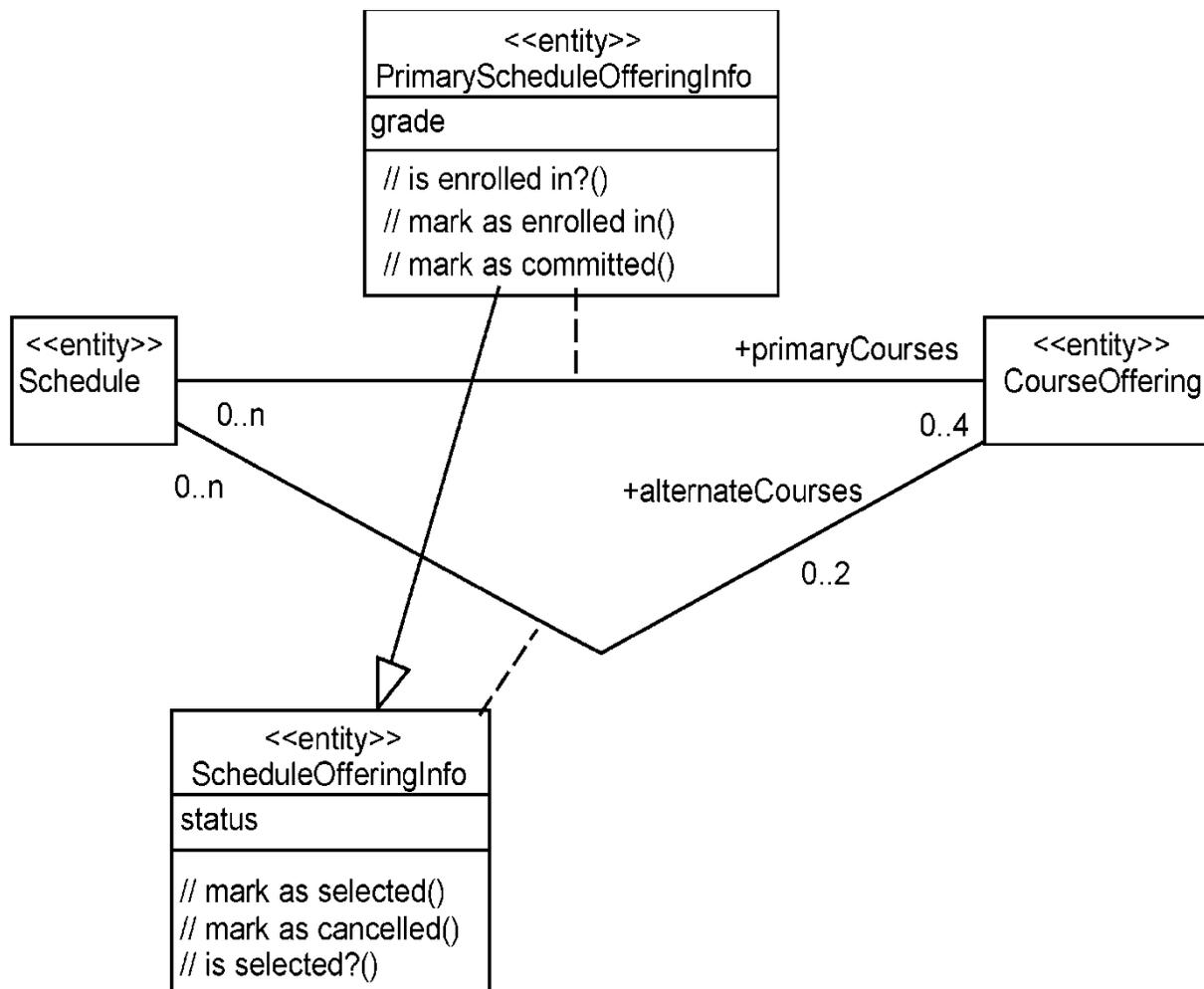
Додамо зв'язки з класами, що беруть участь у варіанті використання Register for Courses. Для відображення зв'язків між

Класами побудуємо три нові діаграми класів у кооперації Register for Courses пакета Use-Case Realization - Register for Courses (рис. 14 - 16).



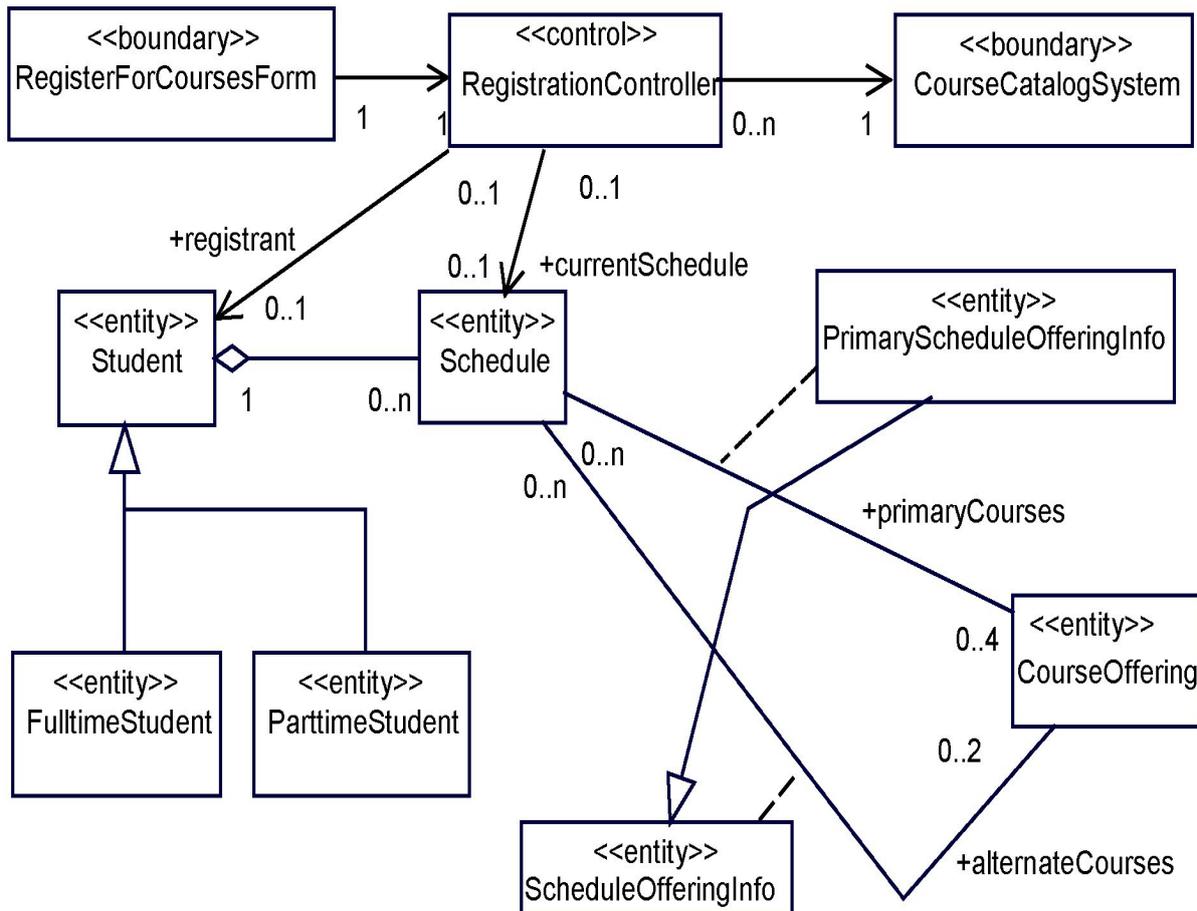
Мал. 14. Діаграма Entity Classes (класи-сутності)

Додано два нових класи - підкласи FulltimeStudent (студент очного відділення) та ParttimeStudent (студент вечірнього відділення).



Мал. 15. Діаграма CourseOfferingInfo

На цій діаграмі показані класи асоціацій, що описують зв'язки між класами Schedule та CourseOffering та доданий суперклас ScheduleOfferingInfo. Дані та операції, що містяться в цьому класі (status - курс включений до графіка або скасовано), відносяться як до основних, так і до альтернативних курсів, у той час як оцінка (grade) та остаточне включення курсу до графіка можуть мати місце тільки для основних курсів.



Мал. 16. Повна діаграма класів VOPC (без атрибутів та операцій)

Створення асоціацій

Асоціації створюють безпосередньо на діаграмі класів. Панель інструментів діаграми класів містить кнопки для створення як одно-, так і двонаправлених асоціацій.

Щоб на діаграмі класів створити асоціацію:

1. Натисніть кнопку Association на панелі інструментів.
2. Проведіть мишею лінію асоціації від одного класу до іншого.

Щоб задати можливості навігації по асоціації:

1. Клацніть правою кнопкою миші на зв'язку з кінця, на якому хочете показати стрілку.
2. У меню виберіть пункт Navigable.

Щоб створити рефлексивну асоціацію:

1. На панелі інструментів діаграми натисніть кнопку Association.
2. Проведіть лінію асоціації від класу до якогось місця поза класом.
3. Відпустіть кнопку миші.
4. Проведіть лінію асоціації назад до класу.

Створення агрегацій

1. Натисніть кнопку Aggregation на панелі інструментів.

2. Проведіть лінію агрегації від класу-частини до цілого.

Щоб помістити на діаграму класів рефлексивну агрегацію:

1. На панелі інструментів діаграми натисніть кнопку Aggregation.
2. Проведіть лінію агрегації від класу до якогось місця поза класом.
3. Відпустіть кнопку миші.
4. Проведіть лінію агрегації до класу.

Створення узагальнень

При створенні узагальнення може знадобитися перенести деякі атрибути або операції з одного класу до іншого. Якщо, наприклад, знадобиться перенести їх із підкласу в суперклас Employee, у браузері для цього досить просто перетягнути атрибути або операції з одного класу до іншого. Не забудьте видалити іншу копію атрибута з другого підкласу, якщо він є.

Щоб помістити узагальнення на діаграму класів:

1. Натисніть кнопку Generalization на панелі інструментів.
2. Проведіть лінію узагальнення від підкласу до суперкласу.

Специфікації зв'язків

Специфікації зв'язків стосуються імен асоціацій, рольових імена, множинності та класів асоціацій.

Щоб задати множинність зв'язку:

1. Клацніть правою кнопкою миші на одному кінці зв'язку.
2. У меню виберіть пункт Multiplicity.
3. Вкажіть потрібну множину.
4. Повторіть те саме для іншого кінця зв'язку.

Щоб задати ім'я зв'язку:

1. Виділіть потрібний зв'язок.
2. Введіть її ім'я.

Щоб задати зв'язку рольове ім'я:

1. Клацніть правою кнопкою миші на асоціації з потрібного кінця.
2. У меню виберіть пункт role Name.
3. Введіть рольове ім'я.

Щоб задати елемент зв'язку (клас асоціацій):

1. Відкрийте вікно специфікації необхідного зв'язку.
2. Перейдіть на вкладку Detail.

3. Задайте елемент зв'язку в полі Link Element.

Завдання для самостійної роботи

Виконати аналіз варіанта використання Close Registration та побудувати відповідні діаграми взаємодії та класів.

6. Проектування системи

6.1. Проектування архітектури

Цілі проектування архітектури системи:

- аналіз взаємодій між класами аналізу, виявлення підсистем та інтерфейсів;
- уточнення архітектури з урахуванням можливостей повторного використання;
- ідентифікація архітектурних рішень та механізмів, необхідних для проектування системи.

Вводяться глобальні пакети:

- базисні (foundation) класи (списки, черги тощо);
- обробники помилок (error handling classes);
- математичні бібліотеки;
- утиліти;
- бібліотеки інших постачальників Визначаються проектні класи (design classes):
- клас аналізу відображається у проектний клас, якщо він простий або представляє єдину логічну абстракцію;
- складний клас аналізу може бути розбитий на кілька класів, перетворений на пакет або в підсистему.

Приклади можливих підсистем:

- класи, що забезпечують складний комплекс послуг (наприклад, забезпечення безпеки та захист);
- граничні класи, що реалізують складний інтерфейс користувача або інтерфейс із зовнішніми системами;
- різні продукти: комунікаційне програмне забезпечення (middleware, підтримка COM/CORBA), доступ до баз даних, типи та структури даних (стеки, списки, черги), загальні утиліти (математичні бібліотеки), різні прикладні продукти.

Ухвалення рішення про перетворення класу в підсистему визначається досвідом та знаннями архітектора проекту.

Угоди щодо проектування інтерфейсів:

- Ім'я інтерфейсу: коротке (одне-два слова), що відбиває його роль системі.
- Опис інтерфейсу: має відображати його обов'язки (розмір невеликий абзац).
- Опис операцій: ім'я, що відображає результат операції, ключові алгоритми, значення, що повертається, параметри з типами.

- Документування інтерфейсу: характер використання операцій та порядок їх виконання (показується за допомогою діаграм послідовності), тестові плани та сценарії тощо. Вся ця інформація поєднується в спеціальний пакет зі стереотипом «subsystem», який містить елементи, що утворюють підсистему, діаграми послідовності та/або кооперативні діаграми, що описують взаємодію елементів при реалізації операцій інтерфейсу та інші діаграми.
- Клас «subsystem проху» безпосередньо реалізує інтерфейс і керує реалізацією його операцій.
- Всі інтерфейси повинні бути повністю визначені в процесі проектування архітектури, оскільки вони будуть служити як точки синхронізації при паралельній розробці.

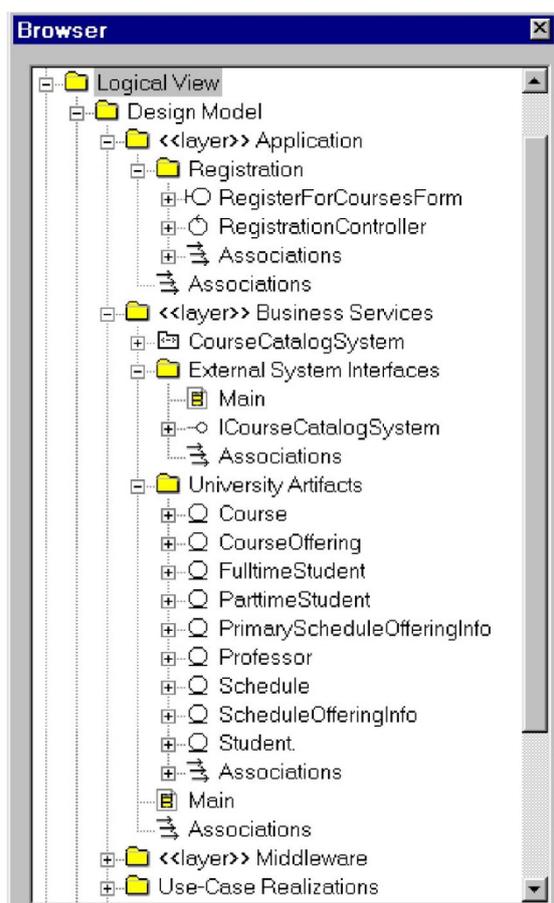
Виділення архітектурних рівнів:

Application Layer - містить елементи прикладного рівня (інтерфейс користувача);

Business Services Layer - містить елементи, що реалізують бізнес-логіку додатків (найстійкіша частина системи);

Middleware Layer – забезпечує сервіси, незалежні від платформи.

Приклад виділення архітектурних рівнів та об'єднання елементів моделі пакети для системи реєстрації наведено на рис. 17.



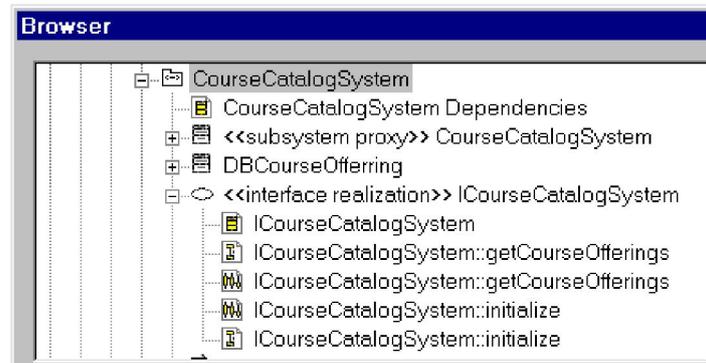
Мал. 17. Структура логічного представлення моделі на етапі проектування

Щоб помістити клас у пакет, досить просто перетягнути його в браузері на потрібний пакет. Дане уявлення відображає такі рішення, прийняті архітектором:

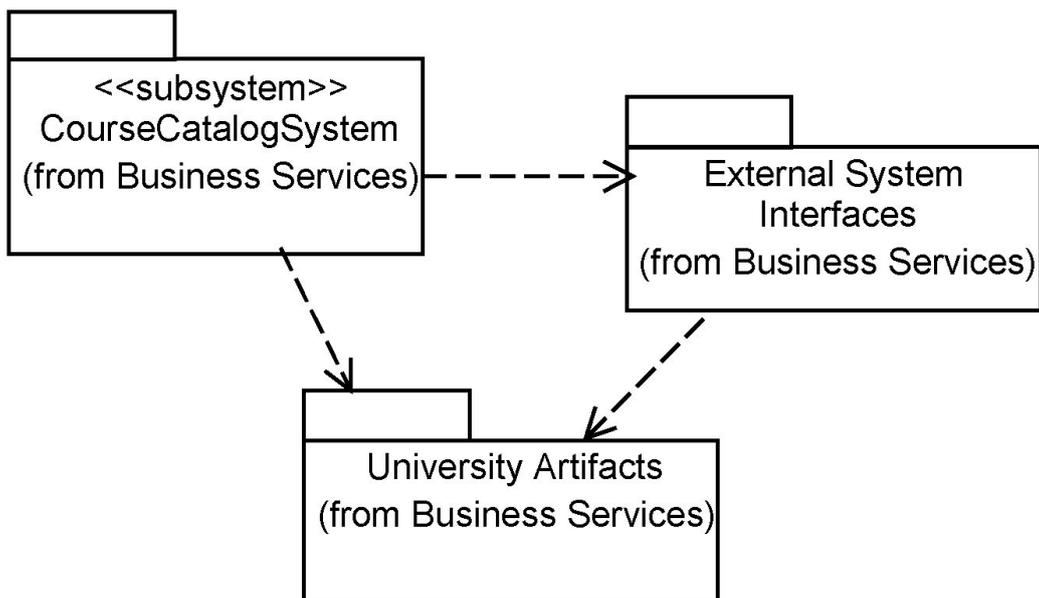
- Виділено три архітектурні рівні (створені три пакети зі стереотипом <<layer>>);
- У пакеті Application створено пакет Registration, куди включені граничні та керуючі класи;
- Граничний клас CourseCatalogSystem перетворено на підсистему (пакет CourseCatalogSystem зі стереотипом <<subsystem>>)
- У пакет Business Services, крім підсистеми CourseCatalogSystem, включено ще два пакети: пакет External System Interfaces включає інтерфейс із підсистемою CourseCatalogSystem (клас ICourseCatalogSystem зі стереотипом <<Interface>>), а пакет University Artifacts - всі класи-сутності.

Структура та діаграми пакета (підсистеми) CourseCatalogSystem показано на рис. 18 –

22.



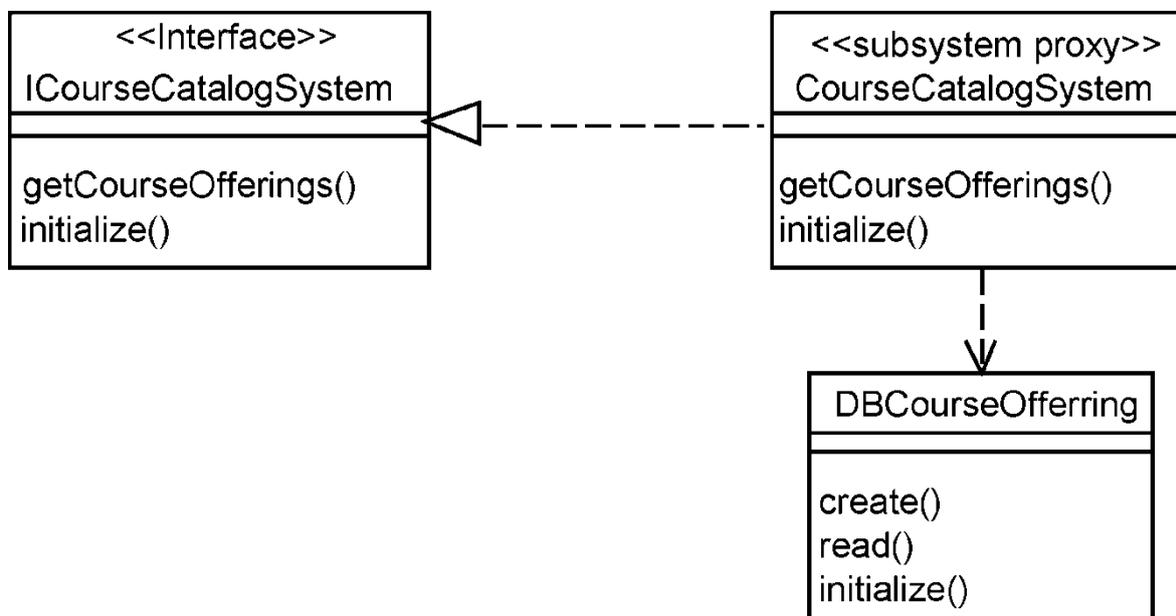
Мал. 18. Структура пакету CourseCatalogSystem



Мал. 19. Залежність між підсистемою та іншими пакетами (діаграма класів CourseCatalogSystem Dependencies)

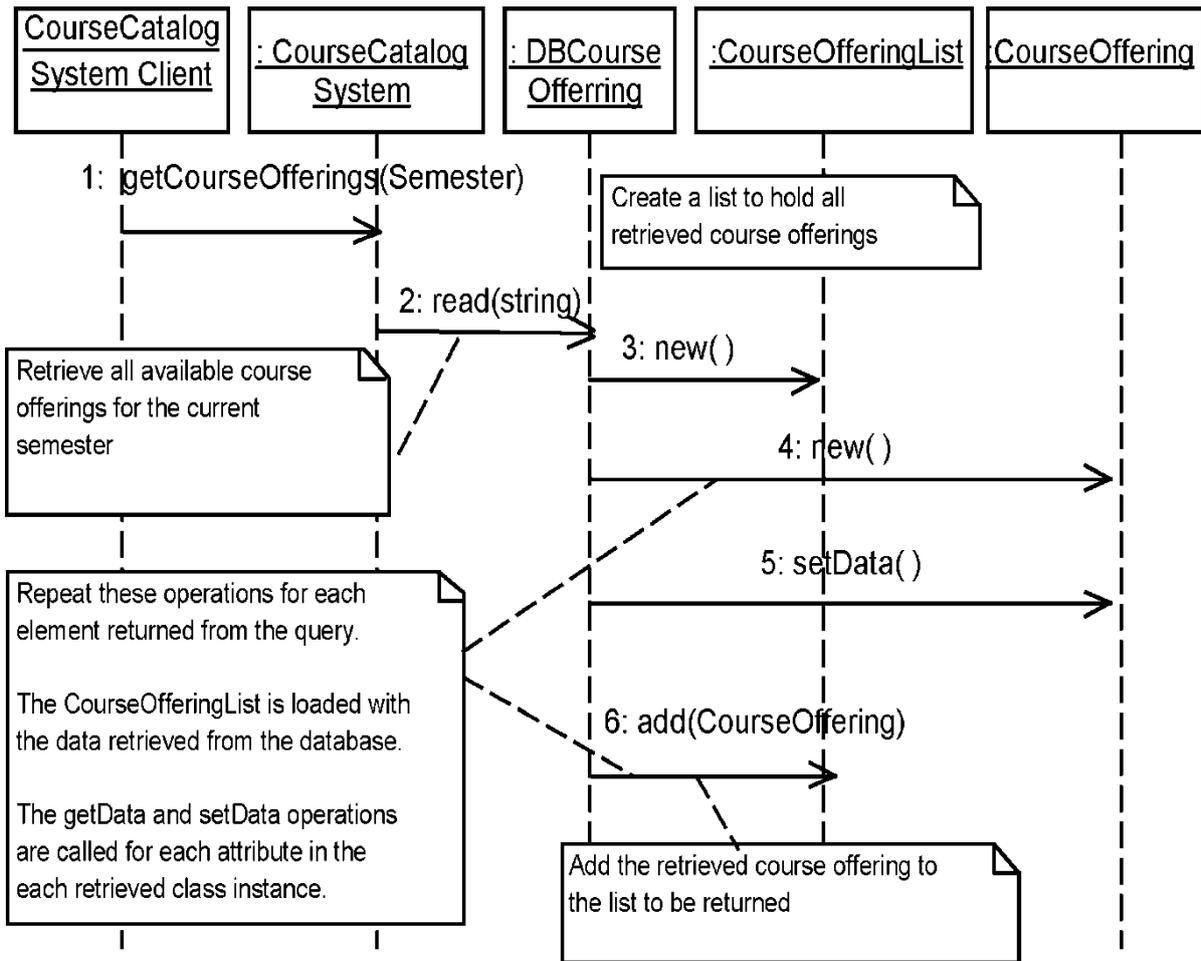
Щоб розмістити залежність між пакетами на діаграму класів:

1. Натисніть кнопку Dependency панелі інструментів.
2. Проведіть лінію залежності від залежного пакета до того, від якого він залежить.

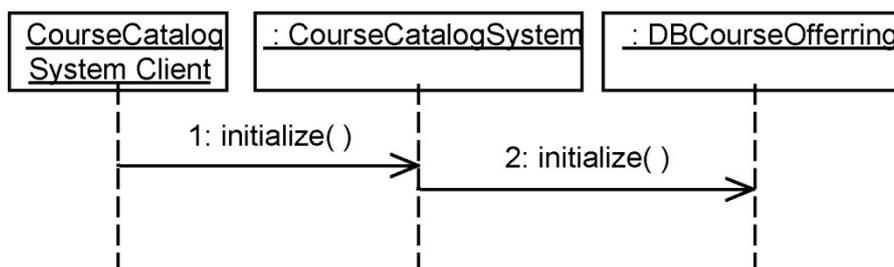


Мал. 20. Класи, що реалізують інтерфейс підсистеми (діаграма класів ICourseCatalogSystem)

Клас DBCourseOffering відповідає за взаємодію з БД каталогу курсів.



Мал. 21. Діаграма послідовності ICourseCatalogSystem::getCourse Offerings, що описує взаємодію елементів при реалізації операції інтерфейсу getCourseOfferings



Мал. 22. Діаграма послідовності ICourseCatalogSystem: initialize, що описує взаємодію елементів при реалізації операції інтерфейсу initialize

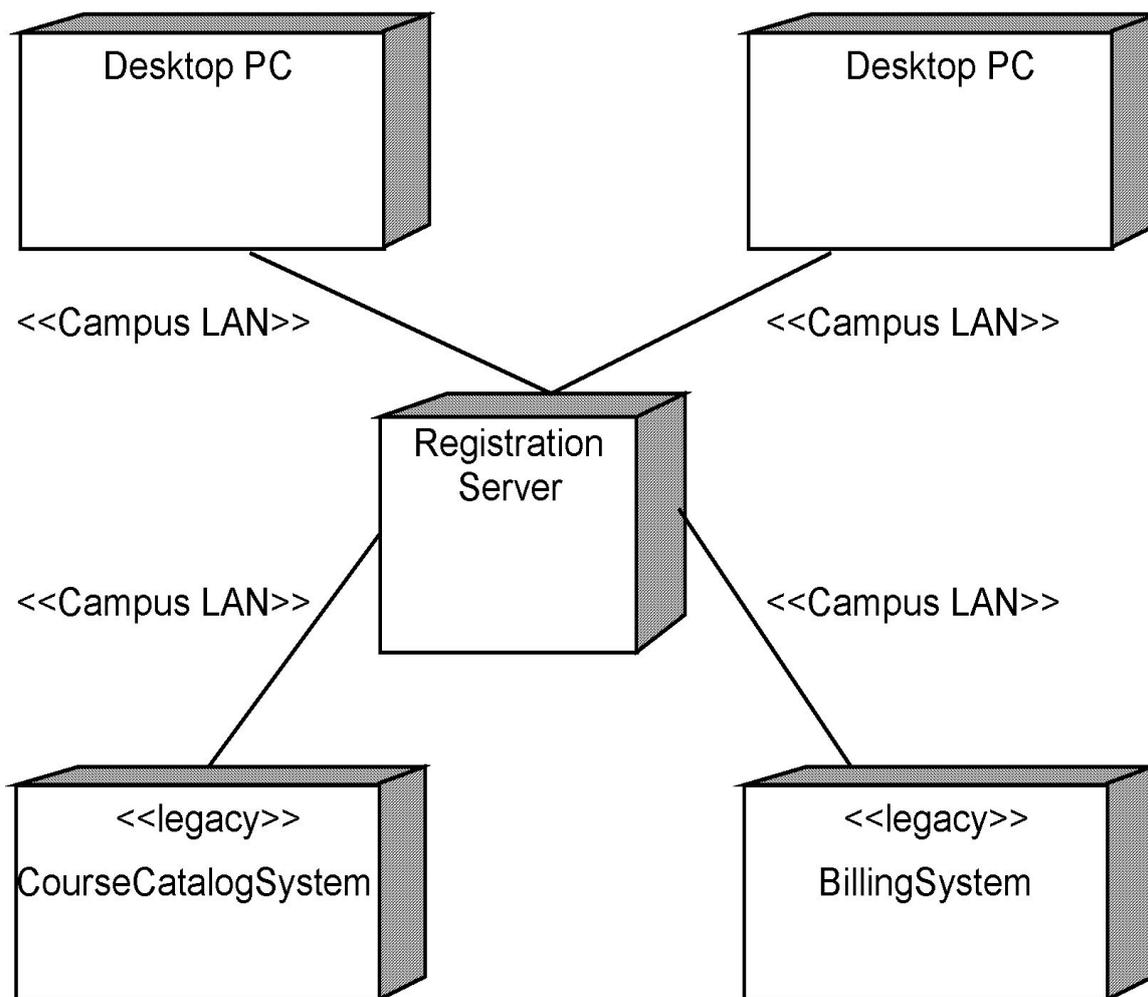
6.2. Моделювання розподіленої конфігурації системи

Розподілена конфігурація системи моделюється за допомогою діаграми розміщення.

Її основні елементи:

- вузол (node) - обчислювальний ресурс (процесор або інший пристрій (дисківа пам'ять, контролери різних пристроїв і т. д.)) Для вузла можна задати процеси, що виконуються на ньому;
- з'єднання (connection) – канал взаємодії вузлів (мережа).

Приклад: мережна конфігурація системи реєстрації (без процесів).

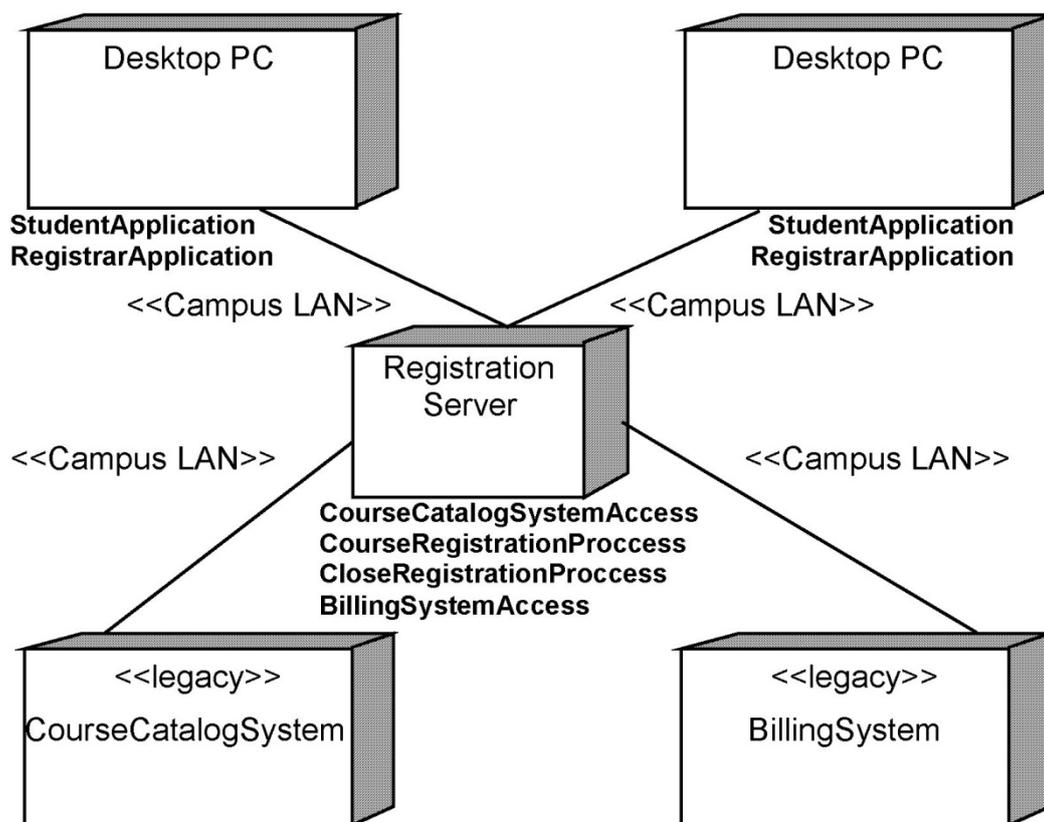


Мал. 23. Мережева конфігурація системи реєстрації

Розподіл процесів по вузлах мережі провадиться з урахуванням наступних факторів:

- використувані зразки розподілу (триланкова клієнт-серверна конфігурація, «товстий клієнт», «тонкий клієнт», рівноправні вузли (peer-to-peer) і т.д.);
- час відгуку;
- мінімізація мережного трафіку;
- потужність вузла;
- надійність обладнання та комунікацій.

Приклад розподілу процесів вузлами наведено на рис. 24.



Мал. 24. Мережева конфігурація системи реєстрації з розподілом процесів по вузлам

Крок 11. Створення діаграми розміщення системи реєстрації

Щоб відкрити діаграму розміщення, треба двічі клацнути мишею на поданні Deployment View (подання розміщення) у браузері.

Щоб помістити на діаграму процесор:

1. На панелі інструментів діаграми натисніть кнопку Processor.
2. Клацніть на діаграмі розміщення там, куди хочете його помістити.
3. Введіть назву процесора.

У специфікаціях процесора можна ввести інформацію про його стереотип, характеристики та планування. Стереотипи використовуються для класифікації процесорів (наприклад, комп'ютерів під керуванням UNIX або ПК).

Характеристики процесора – це його фізичний опис. Воно може, зокрема, включати швидкість процесора та обсяг пам'яті.

Поле планування (scheduling) процесора містить опис того, як здійснюється планування його процесів:

- **Preemptive (з пріоритетом).** Високопріоритетні процеси мають перевагу перед низькопріоритетними.

- **Non preemptive (без пріоритету).** У процесів немає пріоритету. Поточний процес виконується до завершення, після чого починається наступний.
- **Cyclic (циклічний).** Управління передається між процесами за колом. Кожному процесу дається певний час на його виконання, потім керування переходить до наступного процесу.
- **Executive (виконавчий).** Існує певний обчислювальний алгоритм, який управляє плануванням процесів.
- **Manual (вручну).** Процеси плануються користувачем.

Щоб призначити процесору стереотип:

1. Відкрийте вікно специфікації процесора.
2. Перейдіть на вкладку General.
3. Введіть стереотип у полі Stereotype.

Щоб ввести характеристики та планування процесора:

1. Відкрийте вікно специфікації процесора.
2. Перейдіть на вкладку Detail.
3. Введіть параметри в полі параметрів.
4. Вкажіть один із типів планування.

Щоб показати планування на діаграмі:

1. Клацніть правою кнопкою миші на процесорі.
2. У меню виберіть пункт Show Scheduling.

Щоб додати зв'язок на діаграму:

1. На панелі інструментів натисніть кнопку Connection.
2. Клацніть на вузлі діаграми.
3. Проведіть лінію зв'язку до іншого вузла.

Щоб призначити зв'язку стереотип:

1. Відкрийте вікно специфікації зв'язку.
2. Перейдіть на вкладку General.
3. Введіть стереотип у полі Stereotype (Стереотип).

Щоб додати процес:

1. Клацніть правою кнопкою миші на процесорі в браузері.
2. У меню виберіть New > Process.
3. Введіть назву нового процесу.

Щоб показати процеси на діаграмі:

1. Клацніть правою кнопкою миші на процесорі.
2. У меню виберіть пункт Show Processes.

6.3. Проектування класів

Класи аналізу перетворюються на проектні класи:

- Проектування граничних класів - залежить від можливостей середовища розробки інтерфейсу користувача (GUI Builder);
 - Проектування класів-сутностей - з урахуванням міркувань продуктивності (виділення в окремі класи атрибутів із різною частотою використання);
 - Проектування класів, що управляють - видалення класів, що реалізують просту передачу інформації від граничних класів до сутностей;
 - Ідентифікація стійких (persistent) класів, що містять інформацію, що зберігається.
- Обов'язки класів, визначені у процесі аналізу, перетворюються на операції. Кожній операції надається ім'я, що характеризує її результат. Визначається повна сигнатура операції: `operationName(parameter:class,...):returnType`.

Створюється короткий опис операції, включаючи зміст її параметрів. Визначається видимість операції: `public`, `private`, `protected`. Визначається область дії (scope) операції: екземпляр чи класифікатор.

Визначаються (уточнюються) атрибути класів:

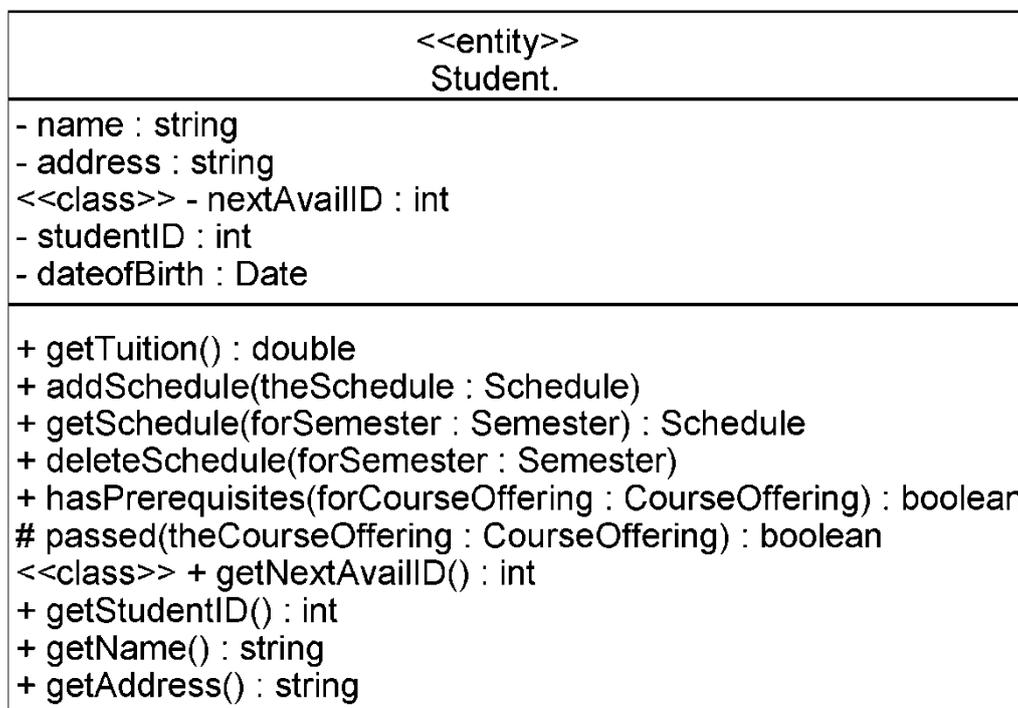
- Крім імені, задається тип і значення за замовчуванням (необов'язкове): `attributeName:Type = Default`;
- Враховуються угоди щодо іменування атрибутів, прийняті у проекті та мові реалізації;
- Встановлюється видимість атрибутів: `public`, `private`, `protected`;
- При необхідності визначаються похідні (обчислювані) атрибути.

Крок12. Визначення атрибутів та операцій для класу Student

Щоб задати тип даних, значення за промовчанням та видимість атрибута:

1. Клацніть правою кнопкою миші на атрибуті браузера.
2. У меню виберіть пункт Open Specification.

3. Вкажіть тип даних у списку типів або введіть власний тип даних.
4. У полі Initial Field (Початкове значення) введіть значення стандартного атрибута.
5. У полі Export Control виберіть видимість атрибута: Public, Protected, Private або Implementation. За промовчанням видимість усіх атрибутів відповідає Private.



Мал. 25. Клас Student з повністю визначеними операціями та атрибутами

Щоб змінити нотацію для позначення видимості:

1. У меню моделі виберіть Tools > Options.
2. Перейдіть на вкладку Notation.
3. Позначте контрольний перемикач Visibility as Icons, щоб використати нотацію Rose, або зніміть позначку, щоб використовувати нотацію UML.

Примітка. Зміна значення цього параметра призведе до зміни нотації тільки нових діаграм і не торкнеться вже існуючі діаграми.

Щоб задати тип значення, що повертається, стереотип і видимість операції:

1. Клацніть правою кнопкою миші на операції у браузері.
2. Відкрийте вікно специфікації класу цієї операції.
3. Вкажіть тип значення, що повертається в розкритому списку або введіть свій тип.
4. Вкажіть стереотип у відповідному списку, що розкривається, або введіть новий.
5. У полі Export Control вкажіть значення видимості операції: Public, Protected, Private або Implementation. За замовчуванням видимість всіх операцій встановлена в public.

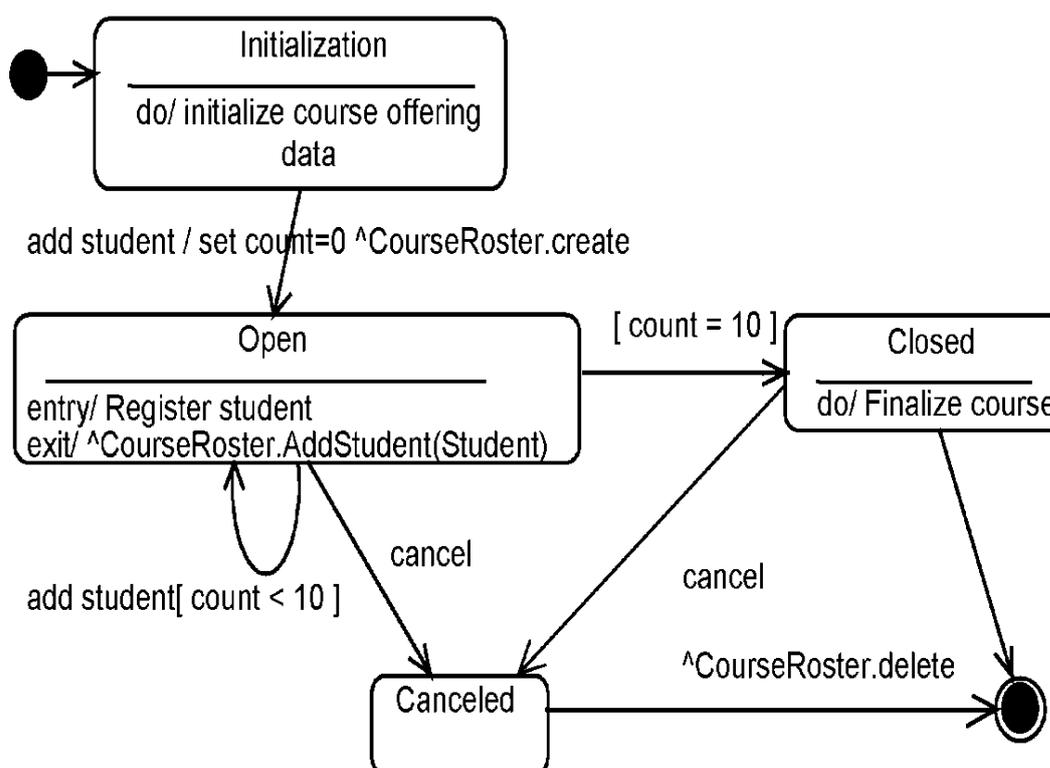
Щоб додати до операції аргумент:

1. Відкрийте вікно специфікації операції.
2. Перейдіть на вкладку Detail.
3. Клацніть правою кнопкою миші в області аргументів, у меню виберіть Insert.
4. Введіть ім'я аргументу.
5. Клацніть на колонці Data type і введіть тип даних аргументу туди.
6. Якщо потрібно, клацніть на колонці default та введіть значення аргументу за промовчанням.

Визначення станів класів моделюється з допомогою діаграм станів.

Діаграми станів створюються для опису об'єктів із високим рівнем динамічної поведінки.

Як приклад розглянемо поведінку об'єкта класу CourseOffering. Він може перебувати у відкритому стані (можливе додавання нового студента) або у закритому стані (максимальна кількість студентів вже записалася на курс). Таким чином, конкретний стан залежить від кількості студентів, пов'язаних з об'єктом CourseOffering. Розглядаючи кожен варіант використання, можна виділити ще два стани: ініціалізація (до початку реєстрації студентів на курс) та скасування (курс виключається із розкладу).



Мал. 26. Діаграма станів для класу CourseOffering

Крок13. Створення діаграми станів для класу CourseOffering**Для створення діаграми станів:**

1. Клацніть правою кнопкою миші у браузері на потрібному класі.
2. У меню виберіть New > Statechart Diagram.

Щоб додати стан:

1. На панелі інструментів натисніть кнопку State
2. Клацніть на діаграмі станів в тому місці, куди хочете його помістити.

Усі елементи стану можна додати за допомогою вкладки Detail вікна специфікації стану.

Щоб додати діяльність:

1. Відкрийте вікно специфікації необхідного стану.
2. Перейдіть на вкладку Detail.
3. Клацніть правою кнопкою миші у вікні Actions.
4. У меню виберіть пункт Insert.
5. Двічі клацніть на новій дії.
6. Введіть дію у полі Actions.
7. У вікні When вкажіть Do, щоб зробити нову дію діяльністю.

Щоб додати вхідну дію, у вікні When вкажіть On Entry. Щоб додати вихідну дію, у вікні When вкажіть On Exit. Щоб надіслати подію:

1. Відкрийте вікно специфікації необхідного стану.
2. Перейдіть на вкладку Detail.
3. Клацніть правою кнопкою миші у вікні Actions.
4. У меню виберіть пункт Insert.
5. Двічі клацніть на новій дії.
6. Як тип дії вкажіть Send Event.
7. У відповідні поля введіть подію (event), аргументи (arguments) та цільовий об'єкт (Target).

Щоб додати перехід:

1. Натисніть кнопку Transition на панелі інструментів.
2. Клацніть на стані, звідки здійснюється перехід.
3. Проведіть лінію переходу до того стану, де він завершується.

Щоб додати рефлексивний перехід:

1. Натисніть кнопку Transition to Self на панелі інструментів.
2. Клацніть на тому стані, де здійснюється рефлексивний перехід.

Щоб додати подію, її аргументи, що захищає умову та дію:

1. Двічі клацніть на переході, щоб відкрити вікно специфікації.
2. Перейдіть на вкладку General.
3. Введіть подію у полі Event.
4. Введіть аргументи у полі Arguments.
5. Введіть умову захисту в полі Condition.
6. Введіть дію у полі Action.

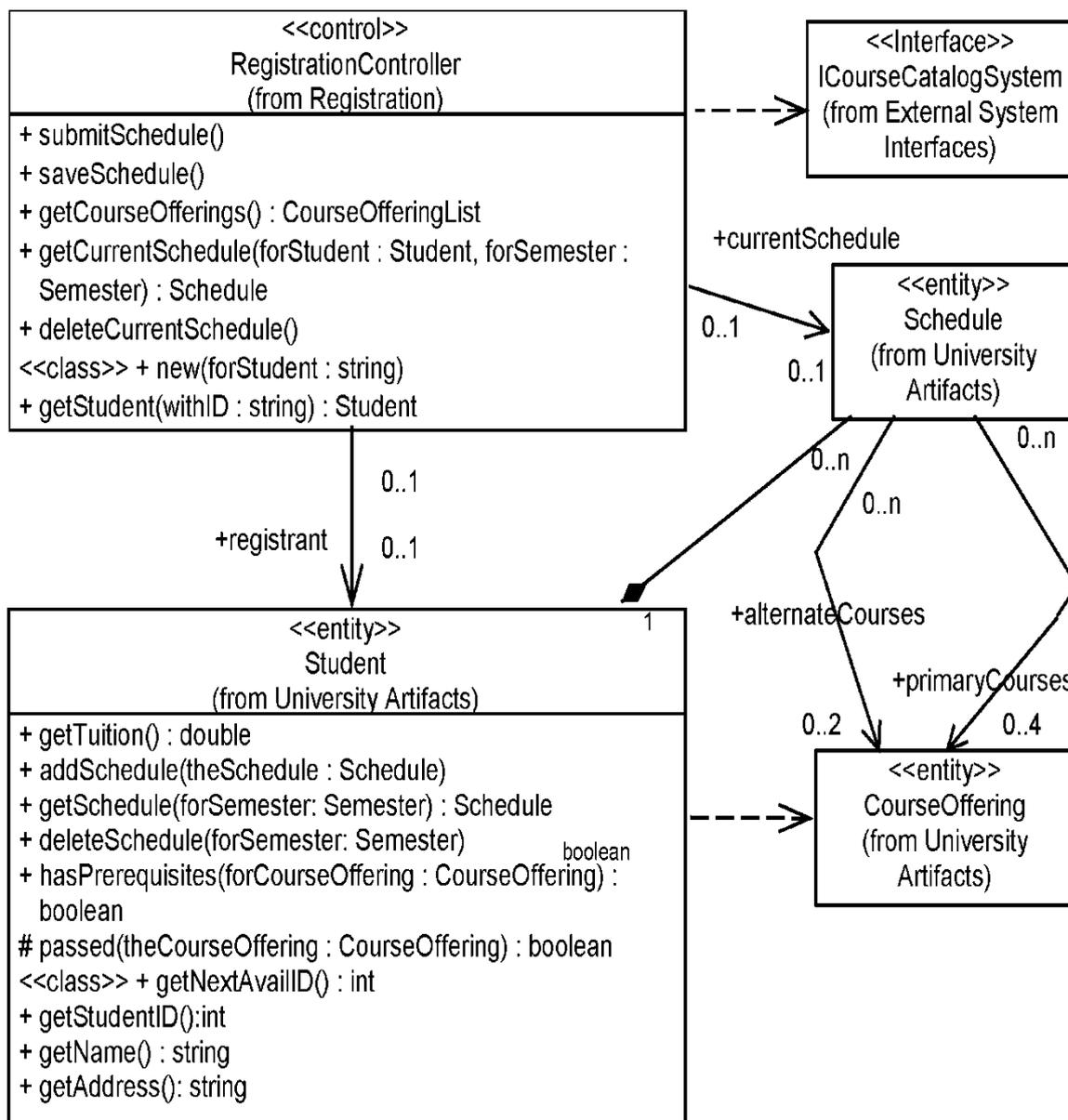
Щоб надіслати подію:

1. Двічі клацніть на переході, щоб відкрити вікно специфікації.
2. Перейдіть на вкладку Detail.
3. Введіть подію у полі Send Event.
4. Введіть аргументи у полі Send Arguments.
5. Задайте ціль у полі Send Target.

Для вказівки початкового чи кінцевого стану:

1. На панелі інструментів натисніть кнопку Start State або End State.
2. Клацніть на діаграмі станів в тому місці, куди хочете помістити стан.

Уточнення асоціацій: Деякі асоціації (семантичні, структурні, стійкі зв'язки за даними) можуть бути перетворені в залежності (неструктурні, тимчасові зв'язки, що відображають видимість), а агрегації - у композиції.



Мал. 27. Приклад перетворення асоціацій та агрегацій

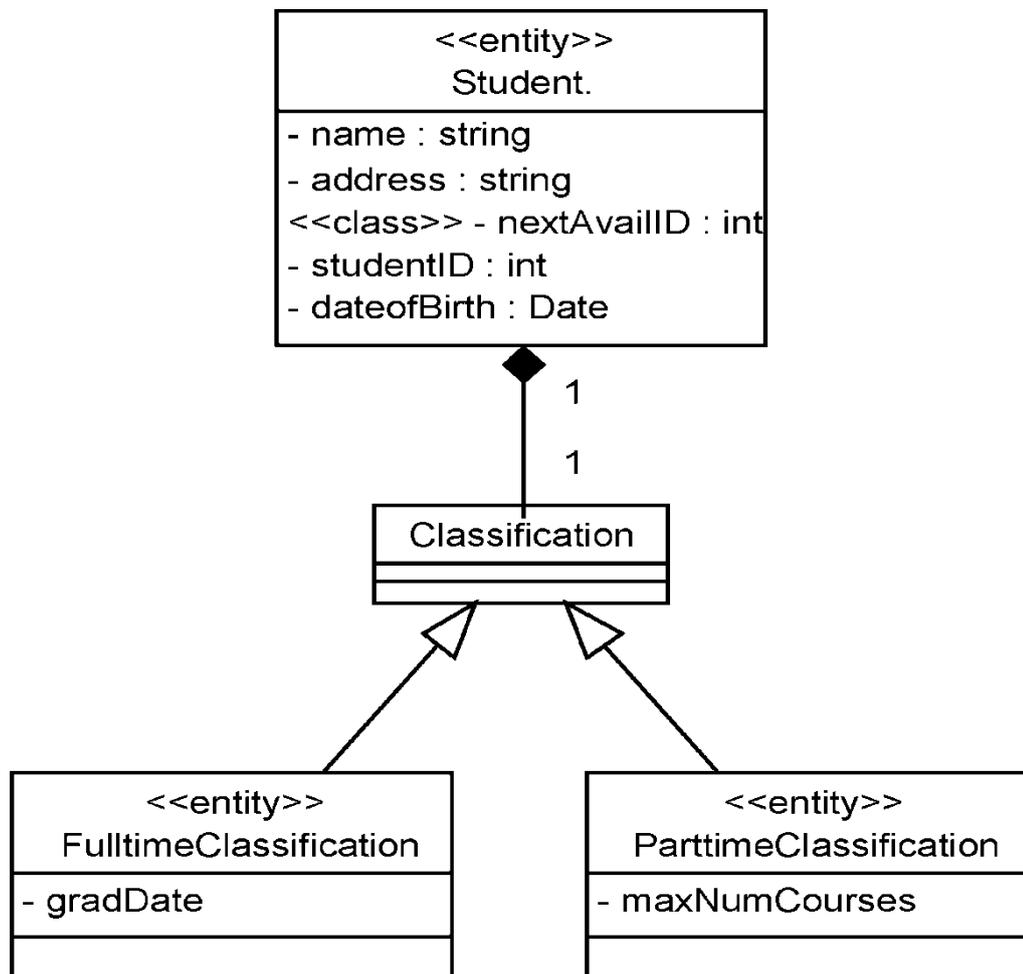
Щоб встановити конвертувати агрегацію в композицію:

1. Клацніть правою кнопкою миші на тому кінці агрегації, який упирається в клас-частину (на рис.27 – Schedule).
2. У меню виберіть пункт Containment.
3. Вкажіть метод включення By Value.

Примітка. Значення *By Value* передбачає, що ціле і частина створюються і руйнуються одночасно, що відповідає композиції. Агрегація (*By Reference*) передбачає, що ціле і частина створюються і руйнуються у час.

Уточнення узагальнень: у разі ситуації з міграцією підкласів (студент може переходити з очної форми навчання на вечірню), ієрархія успадкування реалізується так, як

показано на рис. 28 . Таке рішення підвищує стабільність системи (не потрібно модифікувати опис об'єкта).



Мал. 3.28 Перетворення узагальнення

6.4. Проектування баз даних

Проектування реляційних баз даних виконується за допомогою засобу Data Modeler. Його робота заснована на відомому механізмі відображення об'єктної моделі реляційну. Результатом є побудова діаграми «сутність-зв'язок» та подальша генерація опису БД на SQL.

Крок14. Проектування реляційної бази даних

Проектування БД складається з наступних кроків: Створення нового компонента – бази даних:

1. Клацніть правою кнопкою миші на поданні компонентів.
2. У меню виберіть Data Modeler > New > Database.

3. Відкрийте вікно специфікації новоствореного компонента DB_0 і у списку Target виберіть Oracle 8.x.

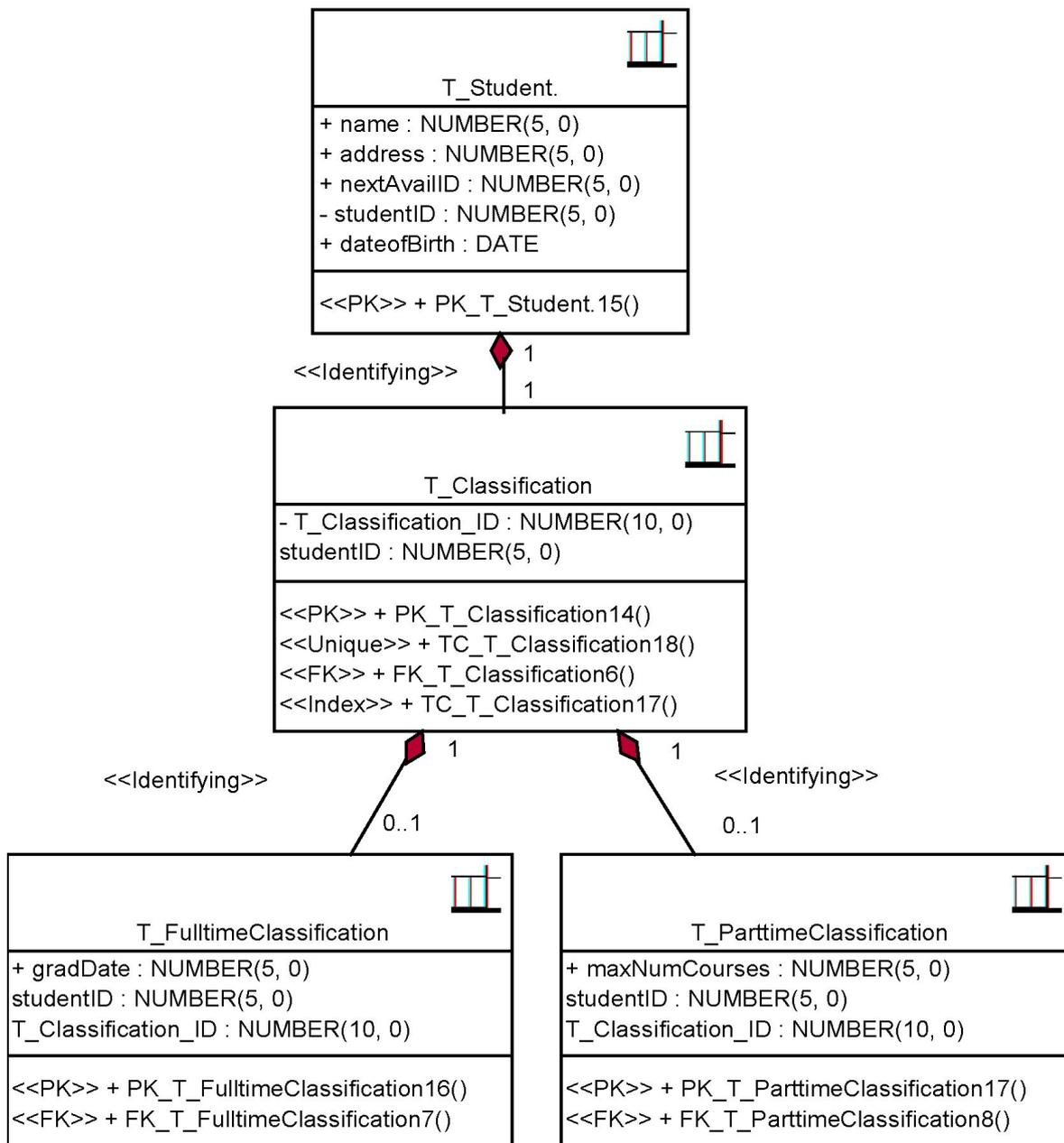
Визначення стійких (persistent) класів:

1. Відкрийте вікно специфікації класу Student у пакеті University Artifacts.
2. Перейдіть на вкладку Detail.
3. Встановіть перемикач Persistence в Persistent.
4. Виконайте такі ж дії для класів Classification, FulltimeClassification та ParttimeClassification.
5. Відкрийте клас Student у браузері, натиснувши "+".
6. Клацніть правою кнопкою миші на атрибуті studentID.
7. У меню виберіть Data Modeler > Part of Object Identity (вказівка атрибута як частина первинного ключа).

Примітка. Кроки 5, 6 та 7 можна виконувати в Rational Rose, починаючи з версії 2001 року.

Створення схеми БД:

1. Клацніть правою кнопкою миші на пакеті University Artifacts.
2. У меню виберіть Data Modeler > Transform to Data Model.
3. У вікні в списку Target Database вкажіть DB_0 і натисніть ОК. В результаті в логічному виставі з'явиться новий пакет Schemas.
4. Відкрийте пакет Schemas та клацніть правою кнопкою миші на пакеті <<Schema>> S_0.
5. У меню виберіть Data Modeler > New > Data Model Diagram.
6. Відкрийте пакет, потім відкрийте новостворену діаграму "сутність-зв'язок" NewDiagram і перенесіть на неї всі класи-таблиці, що знаходяться в пакеті <<Schema>> S_0. Діаграма, що вийшла, показана на рис. 29.



Мал. 29. Діаграма «сутність-зв'язок»

Генерація опису БД на SQL:

1. Клацніть правою кнопкою миші на пакеті <<Schema>> S_0.
2. У меню виберіть Data Modeler > Forward Engineer.
3. У вікні майстра Forward Engineering Wizard натисніть Next.
4. Позначте всі прапорці генерації DDL та натисніть Next.
5. Вкажіть ім'я та розташування текстового файлу з результатами генерації та натисніть Next.
6. Після завершення створення відкрийте створений текстовий файл і перегляньте результати.

7. Реалізація системи

7.1. Створення компонентів

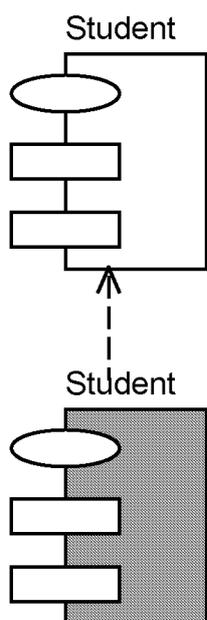
У Rational Rose діаграми компонентів створюються у поданні компонентів системи. Окремі компоненти можна створювати безпосередньо на діаграмі або перетягувати їх туди з браузера.

Крок15. Створення компонентів

Виберемо як мову програмування C++ і класу Student створимо відповідні цій мові компоненти.

Створення діаграми компонентів:

- 1 . Двічі клацніть на головній діаграмі компонентів у поданні компонентів.
3. На панелі інструментів натисніть кнопку Package Specification.
4. Покладіть специфікацію пакета на діаграму.
5. Введіть ім'я специфікації пакета Student та вкажіть у вікні специфікації мову C++.
6. На панелі інструментів натисніть кнопку Package Body.
7. Покладіть тіло пакета на діаграму.
8. Введіть ім'я тіла пакета Student та вкажіть у вікні специфікації мову C++.
9. На панелі інструментів натисніть кнопку Dependency.
- 10.Проведіть лінію залежності від тіла пакету до специфікації пакет.



Мал. 30. Діаграма компонентів

Співвіднесення класів із компонентами:

1. У логічному уявленні браузера знайдіть клас Student.
2. Перетягніть цей клас на специфікацію пакета компонента Student у поданні компонентів браузера. В результаті клас Student буде співвіднесений із специфікацією пакета компонента Student.
3. Перетягніть клас Student на тіло пакету компонента Student у поданні компонентів браузера. В результаті клас Student буде співвіднесений із тілом пакету компонента Student.

7.2. Генерація коду

Процес генерації коду складається з чотирьох основних кроків:

1. Перевірка правильності моделі.
2. Встановлює властивості генерації коду.
3. Вибір класу, компонента чи пакета.
4. генерація коду.

Для перевірки моделі:

1. Виберіть у меню Tools > Check Model.
2. Проаналізуйте усі знайдені помилки у вікні журналу.

До найпоширеніших помилок відносяться такі, наприклад, як повідомлення на діаграмі послідовності або кооперативної діаграми, які не співвіднесені з операцією, або об'єкти цих діаграм, не співвіднесені з класом.

За допомогою пункту меню Check Model можна виявити більшу частину неточностей та помилок у моделі. Пункт меню Access Violations дозволяє виявляти порушення правил доступу, що виникають тоді, коли існує зв'язок між двома класами різних пакетів, але зв'язок між самими пакетами немає.

Щоб виявити порушення правил доступу:

1. Виберіть у меню Report > Show Access Violations.
2. Проаналізуйте усі порушення правил доступу у вікні.

Можна встановити кілька параметрів створення коду для класів, атрибутів, компонентів та інших елементів моделі. Цими властивостями визначається спосіб створення програм. Для кожної мови Rose передбачено ряд певних властивостей генерації коду. Перед генерацією коду рекомендується аналізувати ці властивості та вносити необхідні зміни.

Для аналізу властивостей генерації коду виберіть Tools > Options, а потім вкладку відповідної мови. У списку можна вибрати клас, атрибут, операцію та інші елементи моделі.

Для кожної мови в цьому списку наведено власні елементи моделі. Під час вибору різних значень на екрані з'являються різні набори властивостей.

Будь-які зміни, що вносяться до набору властивостей у вікні Tools > Options, впливають на всі елементи моделі, для яких цей набір використовується.

Іноді потрібно змінити властивості генерації коду одного класу, атрибуту, однієї операції тощо. Для цього відкрийте вікно специфікації елемента моделі. Виберіть вкладку мови (C++, Java, ...) та змініть властивості тут. Усі зміни, які вносяться у вікні специфікації елемента моделі, впливають лише цей елемент.

При генерації коду за один раз можна створити клас, компонент або пакет. Код генерується за допомогою діаграми чи браузера. При генерації коду з пакета можна вибрати або пакет логічного подання на діаграмі класів, або пакет подання компонентів на діаграмі компонентів. При виборі пакета логічного уявлення генеруються класи цього пакета. Вибираючи пакет представлення компонентів, генеруються всі компоненти цього пакета.

Після вибору класу або компонента на діаграмі виберіть у меню відповідний варіант створення коду. Повідомлення про помилки, що виникають у процесі генерації коду, з'являються у вікні журналу.

Під час генерації коду Rose вибирає інформацію з логічного та компонентного уявлень моделі та генерує великий обсяг «скелетного» (skeletal) коду:

- **Класи.** Генеруються усі класи моделі.
- **Атрибути.** Код включає атрибути кожного класу, у тому числі видимість, тип даних та значення за промовчанням.
- **Сигнатури операцій.** Код містить визначення операцій з усіма параметрами, типами даних параметрів і типом значення операції, що повертається.
- **Зв'язки.** Деякі зв'язки моделі викликають створення атрибутів при генерації коду.
- **компоненти.** Кожен компонент реалізується у вигляді файлу з вихідним кодом.

Крок16. Генерація коду C++

1. Відкрийте діаграму компонентів системи.
2. Виберіть усі об'єкти на діаграмі компонентів.
3. Виберіть Tools > C++ > Code Generation у меню.
4. Виконайте генерацію коду.
5. Перегляньте результати генерації (меню Tools > C++ > Browse Header та Tools > C++ > Browse Body).