**Лабораторна робота №4.** **Побудова діаграм кооперації**

**Мета:** **Навчитися будувати діаграми** **кооперації**

**Постановка задачі**. Продовжуємо працювати з постановкою задачі з минулих ЛР та ПР по створенню застосунку для викладачів та студентів гіпотетичного навчального закладу. Ті, хто обрав власну тему продовжують працювати за цією темою. На основі визначених в попередніх ЛР та ПР акторів, прецедентів та класів потрібно розробити діаграму кооперації.

**Завдання до лабораторної роботи**

1. Ознайомитись з теоретичною частиною та рекомендаціями до виконання роботи.
2. Запустити Umbrello UML Modeller (в разі відсутності роботу виконувати з іншим доступним засобом UML-моделювання).
3. Відкрити попередній проект, в ньому створити діаграму кооперації (в Umbrello UML Modeller *communication diagram –* діаграма комунікації, яка в UML 1.x  називалася діаграмою кооперації - collaboration diagram)для кожного раніш створеного прецеденту з урахуванням раніш створеної діаграми послідовностей. В разі потреби – уточнити діаграму прецедентів.
4. Додати нотатки, де записати своє прізвище та групу.
5. Збережіть проект у власній папці та закрийте проект. Для відсилки викладачу збережіть проект з іменем Вашої ЛР та розширенням xmi.
6. Оформите звіт, в якому надайте скріншоти створеної діаграми та відповіді на запитання в кінці ЛР.
7. Файл зі звітом та створену Umbrello UML Modeller модель (файл xmi) Файл зі звітом надсилати на електронну адресу викладача [t.i.lumpova@gmail.com](mailto:t.i.lumpova@gmail.com) з іменем у форматі:

**ОPI<Номер групи><Номер лекції / практичної / лабораторної>[-<Номер завдання>][літера позначення типу роботи L – лекція, P –практична, R – лабораторна]<Прізвищеанглійською>**. Наприклад, О**PI3104L**buts.doc.

Тему в заголовку листа записати

**ОPI <Номер групи>-ЛР <Номер лабораторної>-<Прізвищеанглійською>**

Всі запитання, що виникнуть, надсилайте на електронну адресу викладача, тему в заголовку листа записати

**ОPI <Номер групи>-Запитання-<Прізвище англійською>**.

**Строк виконання цієї роботи ІПЗ-31 02.05.2022**

**ТЕОРЕТИЧНА ЧАСТИНА**

**Діаграми взаємодії (interaction diagrams)**

Діаграми взаємодії є моделями, що описують поводження взаємодіючих груп об'єктів. Як правило, діаграма взаємодії охоплює поводження тільки одного варіанта використання. На такій діаграмі відображається ряд об'єктів і ті повідомлення, якими вони обмінюються між собою в рамках даного варіанта використання.

Цей підхід буде проілюстрований на прикладі простого варіанта використання, що описує таку поведінку:

• «Менеджер» проглядає поточний «Звіт»;

• якщо «Звіт» застарів, «Менеджер» надсилає запит «Виконавцеві» на оновлення «Звіту»;

• «Виконавець» створює новий «Звіт»;

• «Звіт» направляється «Менеджеру» ;

• «Менеджер» виконує перегляд нового «Звіту».

Існує два види діаграм взаємодії: діаграми послідовності і діаграми кооперації/співпраці або, як їх ще називають, кооперативні діаграми.

**Діаграми послідовності (sequence diagrams)**

Діаграми послідовності ми вже розглядали на ЛР№3. Пригадаємо та розглянемо її стосовно нашого прикладу. На діаграмі послідовності об'єкт зображується у вигляді прямокутника на вершині пунктирної вертикальної лінії (рис. 1). Ця вертикальна лінія називається лінією життя об'єкта (lifeline). Вона являє собою фрагмент життєвого циклу об'єкта в процесі взаємодії.

Кожне повідомлення представляється у вигляді стрілки між лініями життя двох об'єктів. Повідомлення з'являються в тому порядку, як вони показані на діаграмі (зверху вниз). Кожне повідомлення може бути позначено ім'ям, за бажанням можна вказати також аргументи й деяку керуючу інформацію. Також можна використовувати самоделегування - повідомлення, яке об'єкт посилає самому собі, при цьому стрілка повідомлення вказує на ту ж саму лінію життя.



Рис. 1. Діаграма послідовності

Із всієї можливої керуючої інформації два її види мають істотне значення. По-перше, це умова, що показує в якому випадку посилається повідомлення. Наприклад, можна ввести умову: [Звіт\_застарів() = true]. Тоді запит на оновлення звіту буде посилатися тільки при виконанні цієї умови. По-друге, корисним керуючим маркером є маркер ітерації, який показує, що повідомлення посилається багато разів для множини об'єктів-адресатів (наприклад, \*оновити).

Активізації - прямокутники на лініях життя - показують, коли метод стає активним (під час його виконання або при очікуванні результату виконання якої-небудь процедури). Використовуючи механізм активізацій, можна більш чітко показати зміст самоделегування. Без цього досить важко визначити, де саме виконуються наступні після самоделегування звернення - у методі, який викликає, чи у методі, який викликається.

**Діаграми кооперації (collaboration diagrams)**

Другим видом діаграми взаємодії є діаграма кооперації, яка з версії UML 2.x називається діаграмою комунікації *(communication diagram)*. Далі будемо використовувати стару назву. На діаграмах кооперації показується взаємодія між об’єктами, які беруть участь у певній події. Ця інформація більшою чи меншою мірою подібна до інформації, показаної на діаграмі послідовностей, але там наголос робиться на часовій характеристиці взаємодії, а на діаграмах кооперації основний наголос робиться на взаємодії між об’єктами та її топології на передньому плані.

Головна особливість діаграми кооперації полягає у можливості графічно представити не тільки послідовність взаємодії, але й всі структурні відношення між об’єктами, що приймають участь в цій взаємодії. Поведінка системи може описуватися на рівні окремих об'єктів, які обмінюються між собою повідомленнями, щоб досягти потрібної мети або реалізувати певний сервіс. Таким чином, за допомогою діаграми кооперації можна описати повний контекст взаємодій як своєрідний часовий "зріз" сукупності об'єктів, які взаємодіють між собою для виконання певного завдання або цілі програмної системи.

На діаграмах кооперації повідомлення надіслані від одного з об’єктів до іншого позначаються стрілочками, поряд з якими показано назву повідомлення, параметри і послідовність повідомлення. Діаграми кооперації найкраще підходять для показу специфічного перебігу виконання або ситуацій у програмі. Такі діаграми є найкращим засобом для швидкого показу і пояснення окремого процесу у програмній логіці.

Діаграма кооперації щодо розглянутого вище прикладу представлена на рис. 2.



Рис. 2. Діаграма кооперації

На діаграмі кооперації екземпляри об'єктів показані у вигляді піктограм. Лінії між ними позначають повідомлення, обмін якими здійснюється в межах даного варіанта використання.

Діаграма кооперації є іншим варіантом діаграми послідовностей.

*Переваги діаграми кооперацій*:

1) Діаграма кооперацій більш компактна, відображає взаємодію, прив'язуючись до архітектури системи.

2) На ній зручніше розташовувати компоненти і відстежувати потоки подій.

3) Краще видно зв'язок з БД

На діаграмі кооперацій вісь часу відсутня, тому нумерація повідомлень обов'язкове.

На рис 3. показано діаграма кооперації у Umbrello UML Modeller.

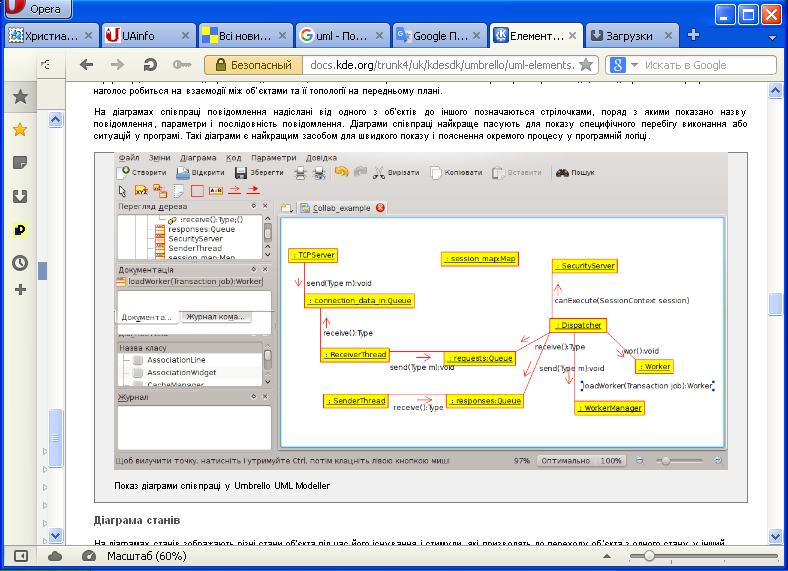


Рис. 3. Показ діаграми кооперації у Umbrello UML Modeller

Розглянемо більш детально діаграми кооперації в загальному випадку (без прив’язки до конкретного інструменту UML-моделювання).

Головна особливість діаграм кооперації – можливість графічно надати не тільки послідовність взаємодії, але й всі структурні відносини між об’єктами, що в ній приймають участь.

Прямокутником на діаграмі кооперації позначаються об’єкти, що приймають участь у взаємодії. В прямокутнику позначається ім’я об’єкту, його клас і, можливо, значення атрибутів. Як і в діаграмі класів асоціації між об’єктами вказуються у вигляді різних з’єднуючих ліній, до яких можна вказати імена асоціацій та ролей, які грають об’єкти в цій асоціації. Додатково можуть бути зображені динамічні зв’язки – потоки повідомлень. Вони надаються у вигляді з’єднувальних ліній між об’єктами, над якими знаходяться стрілка, що вказує напрямок, ім’я повідомлення та порядковий номер в загальній послідовності ініціалізації повідомлень.

На відміну від діаграми послідовностей на діаграмі кооперації не вказується час у вигляді окремого виміру, тому послідовність взаємодії та паралельні потоки можуть бути визначеними за допомогою порядкових номерів. Тому, якщо необхідно розроблювати специфікацію взаємодії між об’єктами в реальному часі, краще використовувати діаграму послідовностей.

**Кооперація** *(collaboration)***.**

Це поняття є одним з фундаментальних у мові UML. Воно слугує для позначення множини об’єктів, що взаємодіють з певною метою, в загальному контексті моделюємої системи. Мета кооперації – специфікувати особливості окремих найбільш значущих операцій у системі. Кооперація визначає структуру поведінки системи у термінах взаємодії учасників цієї кооперації.

Кооперація може бути представлена на двох рівнях:

* рівні специфікації (Specification-Level) - показує ролі класифікаторів и ролі асоціацій у розглядаємій взаємодії;
* рівні прикладів - вказує екземпляри (instance) та зв’язки, які створюють окремі ролі в кооперації.

Діаграма кооперації рівня специфікації показує ролі, які виконують елементи – учасники взаємодії. Елементами кооперації на цьому рівні є класи та асоціації, які позначають окремі ролі класифікаторів (які відповідають класам на діаграмі класів) та асоціації між учасниками кооперації.

Кооперація рівня специфікації зображується на діаграмі пунктирним еліпсом, всередині якого записується ім’я цієї кооперації. Таке представлення кооперації відноситься до окремого варіанту використання і деталізує особливості його подальшої реалізації. Символ еліпса кооперації з’єднується відрізками пунктирної лінії з кожним з учасників цієї кооперації, в якості яких можуть виступати об’єкти або класи. Кожна з цих пунктирних ліній помічується роллю (*role*) учасника. Ролі відповідають іменам елементів в контексті всієї кооперації. Ці імена трактуються як параметри, які обмежують специфікацію елементів при будь-якій їх появі в окремих представленнях моделі.

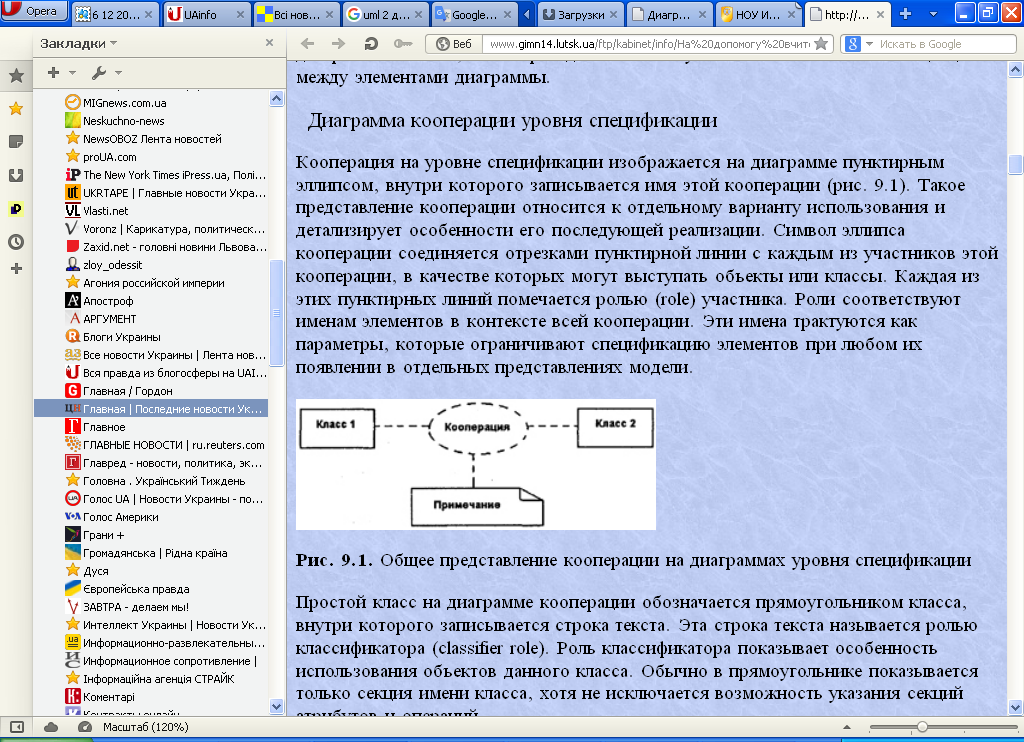
******

Рис. 4. Загальне представлення кооперації на діаграмах рівня специфікації

Простий клас на діаграмі кооперації позначається прямокутником класу, всередині якого записується рядок тексту, який називається роллю класифікатору (*classifier role*). Роль класифікатора показує особливість використання об’єктів даного класу. зазвичай в прямокутнику показується лише секція імені класу.

Для визначення окремих кооперацій, які є спеціалізацією іншої кооперації, використовується звичайна стрілка узагальнення, направлена від символу дочірньої кооперації до символу кооперації-предка. Ролі дочірніх кооперацій можуть бути спеціалізаціями ролей кооперацій-предків.

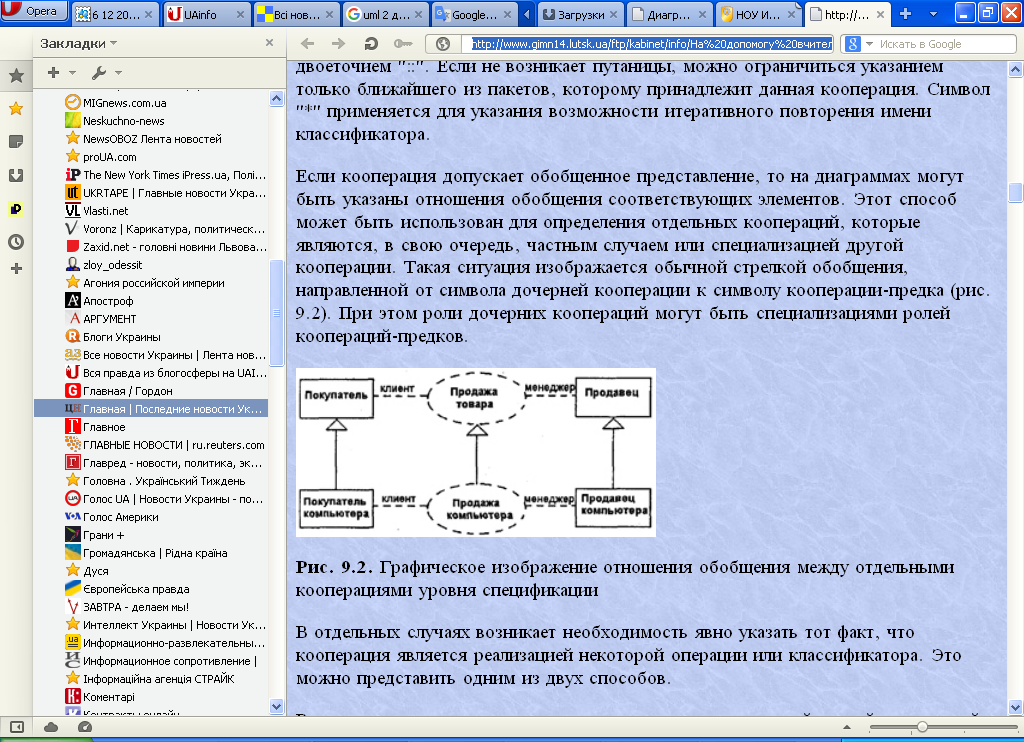


Рис. 5. Графічне зображення відношення узагальнення між окремими коопераціями рівня специфікації

Діаграма кооперації рівня прикладів є сукупністю об’єктів (екземпляри класів) та зв’язків (екземпляри асоціацій). При цьому зв’язки доповнюються стрілками повідомлень. На цьому рівні показуються тільки об’єкти, що мають безпосереднє відношення до реалізації операції або класифікатора. Тут зовсім не обов’язково зображувати усі властивості або усі асоціації, оскільки на діаграмі кооперації присутні тільки роли класифікаторів, а не самі класифікатори. Таким чином, якщо класифікатор потребує повного опису всіх своїх екземплярів, роль класифікатора потребує лише опису тих властивостей та асоціацій, які необхідні для участі в окремій кооперації.

Таким чином, ***одна і та ж сукупність об’єктів може брати участь в різних коопераціях.*** В залежності від розглядаємої кооперації можуть змінюватися як властивості окремих об’єктів, так і зв’язки між ними. Це є відмінністю діаграми кооперації від діаграми класів, для якої потрібно вказувати усі все властивості та асоціації між елементами діаграми.

Об’єкти є основними елементами з яких будується діаграма кооперації рівня прикладів. Оскільки об’єкт (*object*) є окремим екземпляром класу, який створюється на етапі виконання програми, він може мати своє власне ім’я та конкретні значення атрибутів. В загальному випадку ім’я задається так

*<Ім’я об’єкту>'/' <Ім’я ролі класифікатору> ':' <Ім’я класифікатору >*

*[':' < Ім’я класифікатору >]*

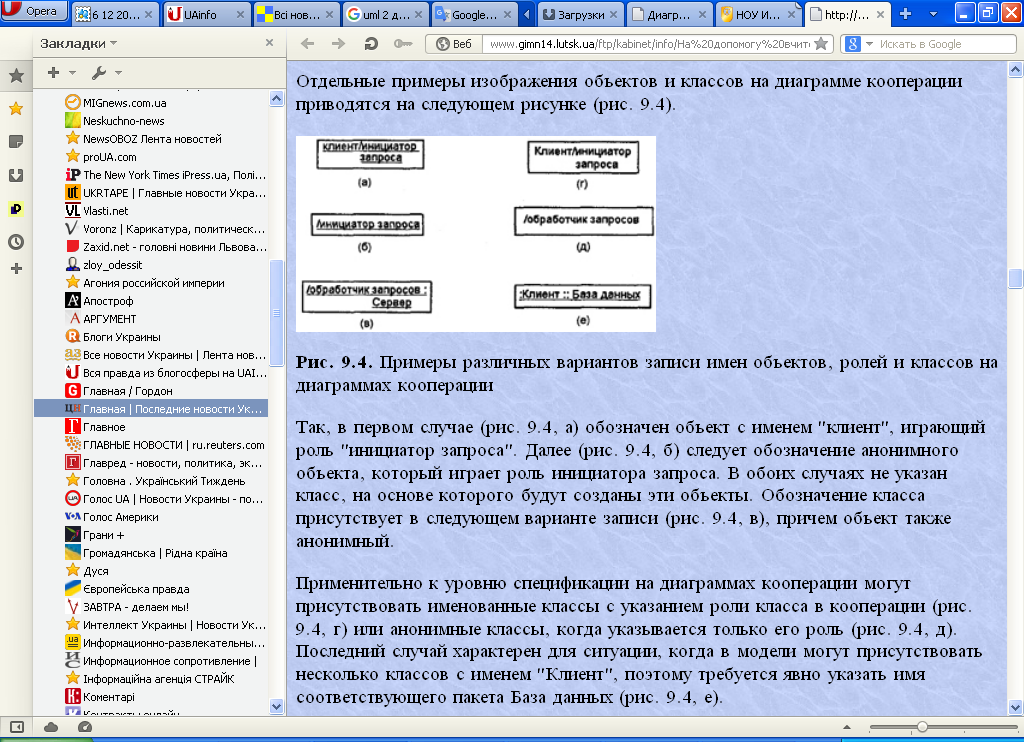


Рис. 6. Приклади різних варіантів запису імен об’єктів, ролей та класів на діаграмах кооперації

У першому випадку (рис.6, а) позначений об’єкт с іменем "клієнт", що грає роль "ініціатор запиту". Далі (рис.6, б) надано позначення анонімного об’єкту, який грає роль ініціатору запиту. В обох випадках не вказаний клас, на основі якого будуть створені ці об’єкти. Позначення класу присутнє в наступному варіанті запису (рис.6,ºв), причому об’єкт також анонімний. На діаграмах кооперації можуть бути присутніми іменовані класи з вказівкою ролі класу в кооперації (рис., г) або анонімні класи, коли вказується тільки його роль (рис.6, д). Останнє характерне для ситуації, коли в моделі можуть бути декілька класів з іменем "Клієнт", тому потрібно явно вказати ім’я відповідного пакета База даних (рис.6, е).

***Мультіоб’єкт*** *(multiobject)* є множиною об’єктів на одному з кінців асоціації. На діаграмі кооперації мультіоб’єкт використовується для того, щоб показати операції та сигнали, які адресовані всій множені об’єктів, а не тільки одному. Мультіоб’єкт зображається двома прямокутниками, один з яких виступає з під правої верхньої вершини іншого (рис.7, а). Стрілка повідомлення відноситься до всієї множини об’єктів, які позначають даний мультіоб’єкт. На діаграмі кооперації може бути явно вказано відношення композиції між мультіоб’єктом та окремим об’єктом з його множини (рис.7, б).

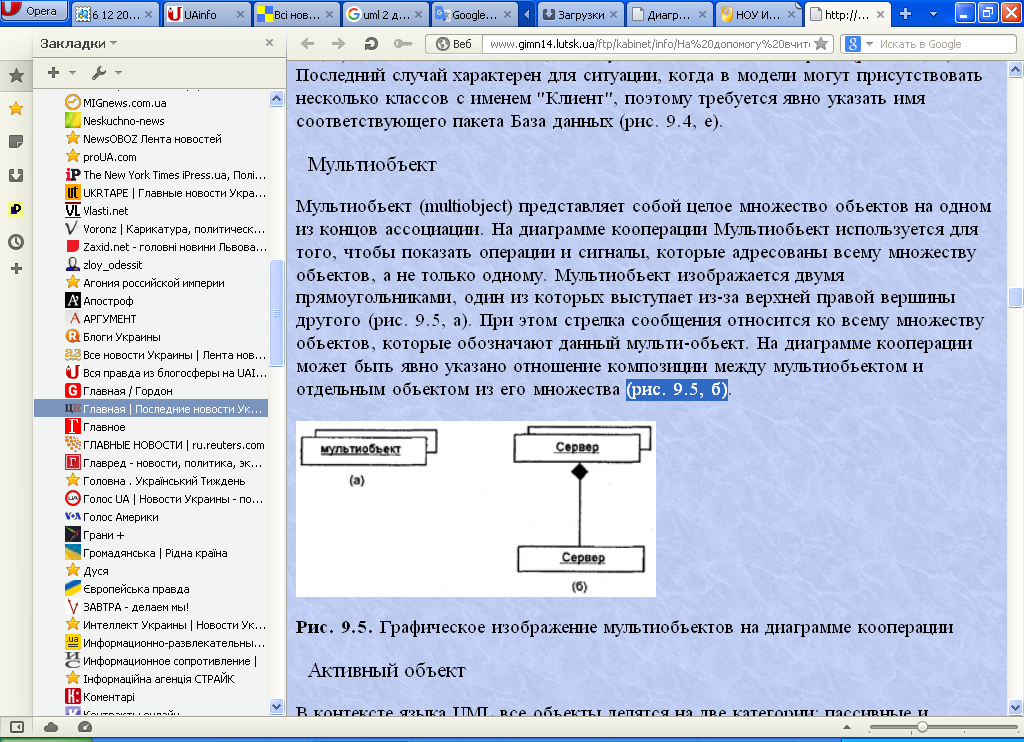


Рис. 7. Графічне зображення мультіоб’єктів на діаграмі кооперації

В прикладі на рис.8 показано ситуацію з викликом функції друку з текстового редактора. Анонімний активний об’єкт "Текстовый редактор" спочатку посилає повідомлення анонімному мультіоб’єкту "Принтер", яке ініціює вибір єдиного об’єкту "Принтер", можливо, задовольняючого деяким додатковим умовам. Після цього обраному об’єкту посилається повідомлення про необхідність друку документу з текстового редактору. ***Активний об’єкт*** *(active object)* має свій деякий полегшений потік (*thread*) керування - нитку і може ініціювати діяльність по керуванню іншими об’єктами. Потік керування може виконуватися паралельно с іншими обчислювальними нитями або нитями керування в межах одного обчислювального процесу або процесу керування. Активні об’єкти на канонічних діаграмах позначаються прямокутником з більш широкими границями, інколи може бути явно вказано ключове слово *{active}* або префікс "а:".

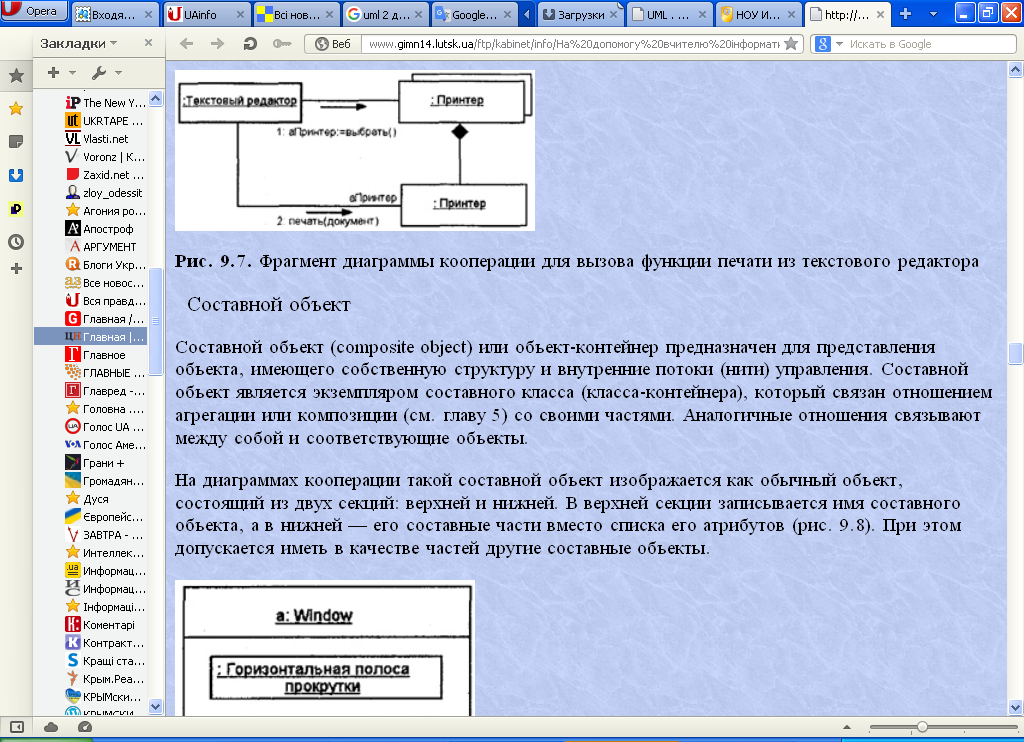


Рис. 8. Фрагмент діаграми кооперації для виклику функції друку з текстового редактора

***Складений об’єкт*** *(composite object)* або об’єкт-контейнер призначений для представлення об’єкта з власною структурою і внутрішніми потоками (нитями) керування. Складений об’єкт є екземпляром класу-контейнера, зв’язаного відношенням агрегації або композиції зі своїми частинами.

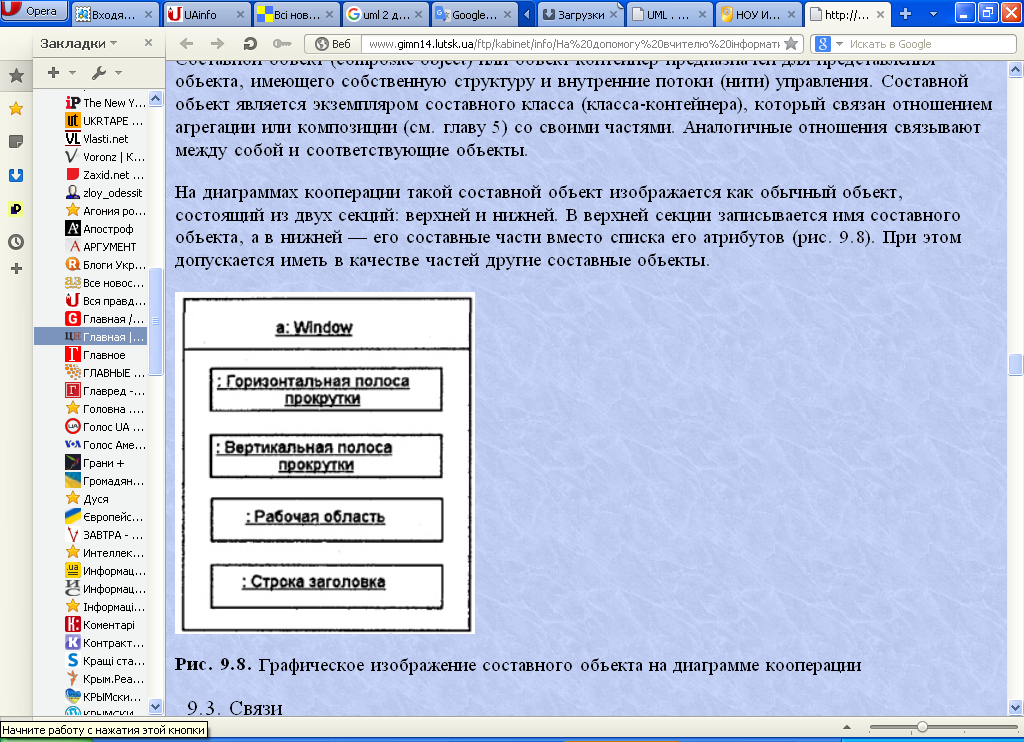


Рис. 9. Графічне зображення вкладеного об’єкта на діаграмі кооперації

**Зв’язки**

Все зв’язки на діаграмі кооперації можуть бути тільки анонімними, для них не вказується кратність. Використовуються такі стереотипи:

* «association» - асоціація (передбачається по замовчуванню, тому цей стереотип можна не вказувати);
* «parameter» - параметр метода. Відповідний об’єкт може бути тільки параметром деякого метода;
* «local» - локальна змінна метода. Її область видимості обмежена лише сусіднім об’єктом;
* «global» - глобальна змінна. Її область видимості поширюється на всю діаграму кооперації;
* «self» - рефлексивний зв’язок об’єкту з самим собою, що допускає передачу об’єктом повідомлення самому себе. На діаграмі кооперації рефлексивний зв’язок зображається петлею в верхній частині прямокутника об’єкту.

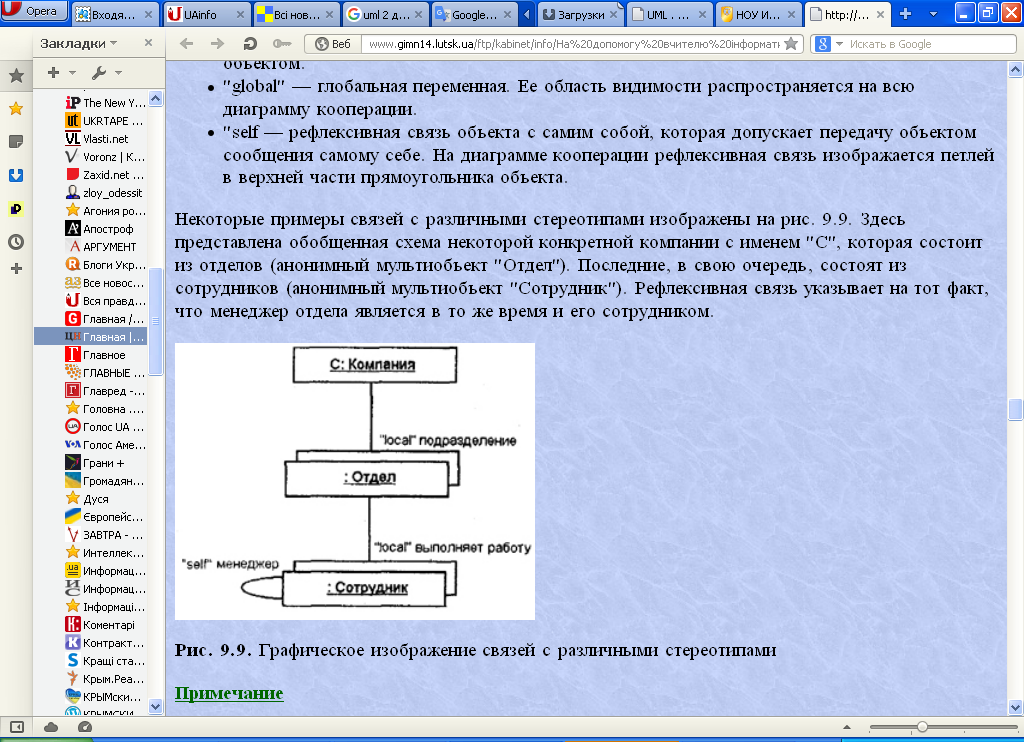


Рис. 10. Графічне зображення зв’язків з різними стереотипами

**Повідомлення**

Повідомлення на діаграмі кооперації специфікує комунікацію між двома об’єктами, один з яких передає іншому деяку інформацію. При цьому, перший об’єкт очікує, що після отримання повідомлення другим об’єктом буде виконана деяка дія.

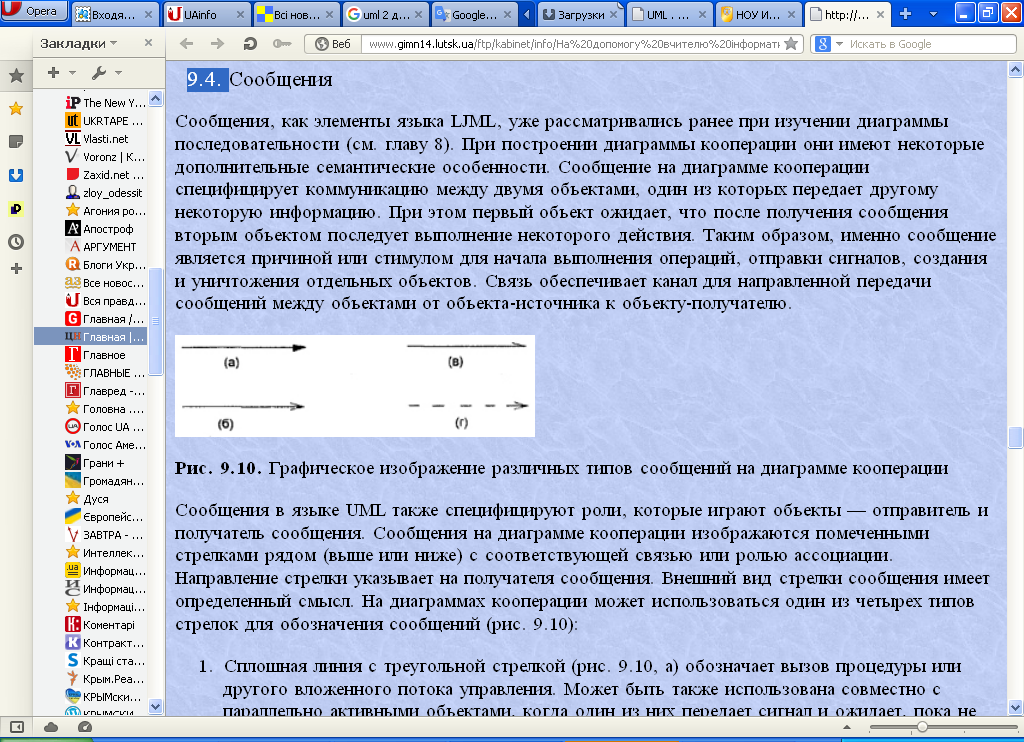


Рис. 11. Графічне зображення різних типів повідомлень на діаграмі кооперації

Зовнішній вид стрілки повідомлення має такий смисл:

1. суцільна лінія з трикутною стрілкою (рис. 11, а) позначає виклик процедури або іншого вкладеного потоку керування. Зазвичай такі повідомлення є синхронними, тобто такі, що ініціюються по завершенні деякої діяльності або при виконання деякої умови;
2. суцільна лінія з V-образною стрілкою (рис. 11, б) позначає простий потік керування. Зазвичай такі повідомлення є асинхронними;
3. суцільна лінія с полустрілкою (рис. 11, в) позначає асинхронний потік керування. Відповідні повідомлення формуються в довільні, заздалегідь не відомі моменти часу, як правило, активними об’єктами. Зазвичай повідомлення такого типа є початковими в послідовності потоку керування й часто ініціюються акторами;
4. пунктирна лінія с V-образною стрілкою (рис. 11, г) позначає повернення з виклику процедури.

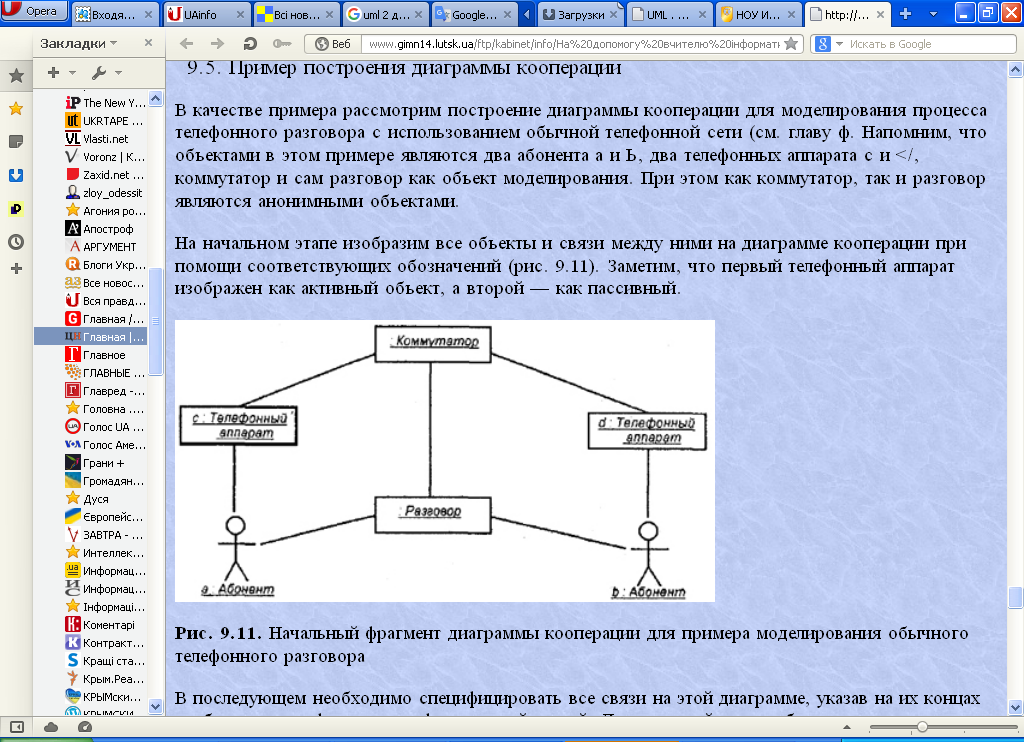
**Приклад побудови діаграми кооперації**.

Рис. 12. Початковий фрагмент діаграми кооперації для прикладу моделювання звичайної телефонної розмови

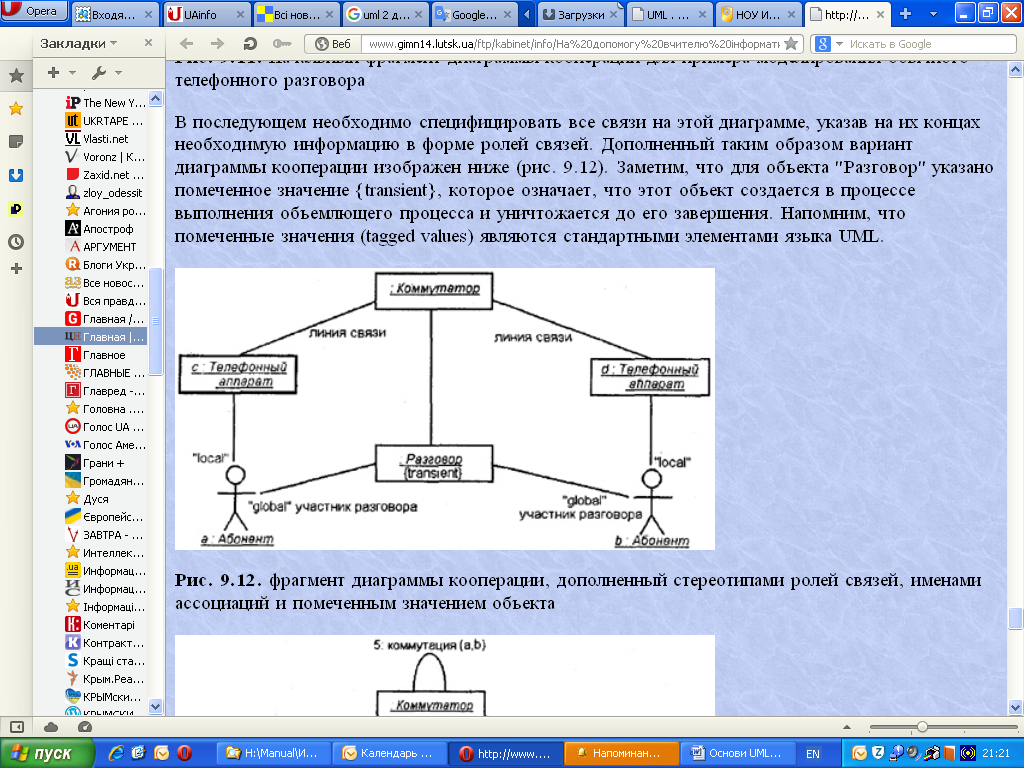


Рис. 13. Фрагмент діаграми кооперації, доповнений стереотипами ролей зв’язків, іменами асоціацій та поміченим значенням об’єкта



Рис. 14. Остаточний варіант діаграми кооперації для прикладу моделювання звичайної телефонної розмови

***Деякі зауваження щодо використання Umbrello UML Modeller.***

В результаті реалізації прецедентів в системі можуть створюватися нові об’єкти. Наприклад, прецедент "Формування звітності" передбачає створення нового об’єкту або сукупності об’єктів, якщо звітів декілька. Аналогічно, відомості про склад груп студентів по навчальним предметам та про склад навчальних предметів групи студентів попередньо повинні бути створені як об’єкти.

Актор з роллю Адміністратора по суті є екземпляром певного класу (в залежності від того, який екземпляр класу буде виконувати цю роль). Це також потрібно враховувати при побудові діаграм кооперації.

Таким чином, формування діаграм кооперації змусить переглянути раніш створені діаграми прецедентів, класів, послідовностей з метою їхнього розширення.

В безкоштовній версії Umbrello UML Modeller не реалізований весь потужний інструментарій UML 2.х, але наявними засобами можна побудувати діаграму кооперації.

**Контрольні питання**

1. Яке призначення діаграм взаємодії? Як співвідносяться між собою діаграми варіантів використання й діаграми взаємодії?
2. Назвіть два види діаграм взаємодії. Що таке «життєва лінія» на діаграмі послідовності?
3. Як на діаграмі послідовності зображені повідомлення? Що таке самоделегування? Що показує активізація об'єкта?
4. Відмінність кооперативних діаграм від діаграм послідовності. Переваги й недоліки кожного виду взаємодії.