

Специфікація функціональних вимог: діаграма станів, діаграма послідовності

Проектування інформаційних систем

Лекція 2

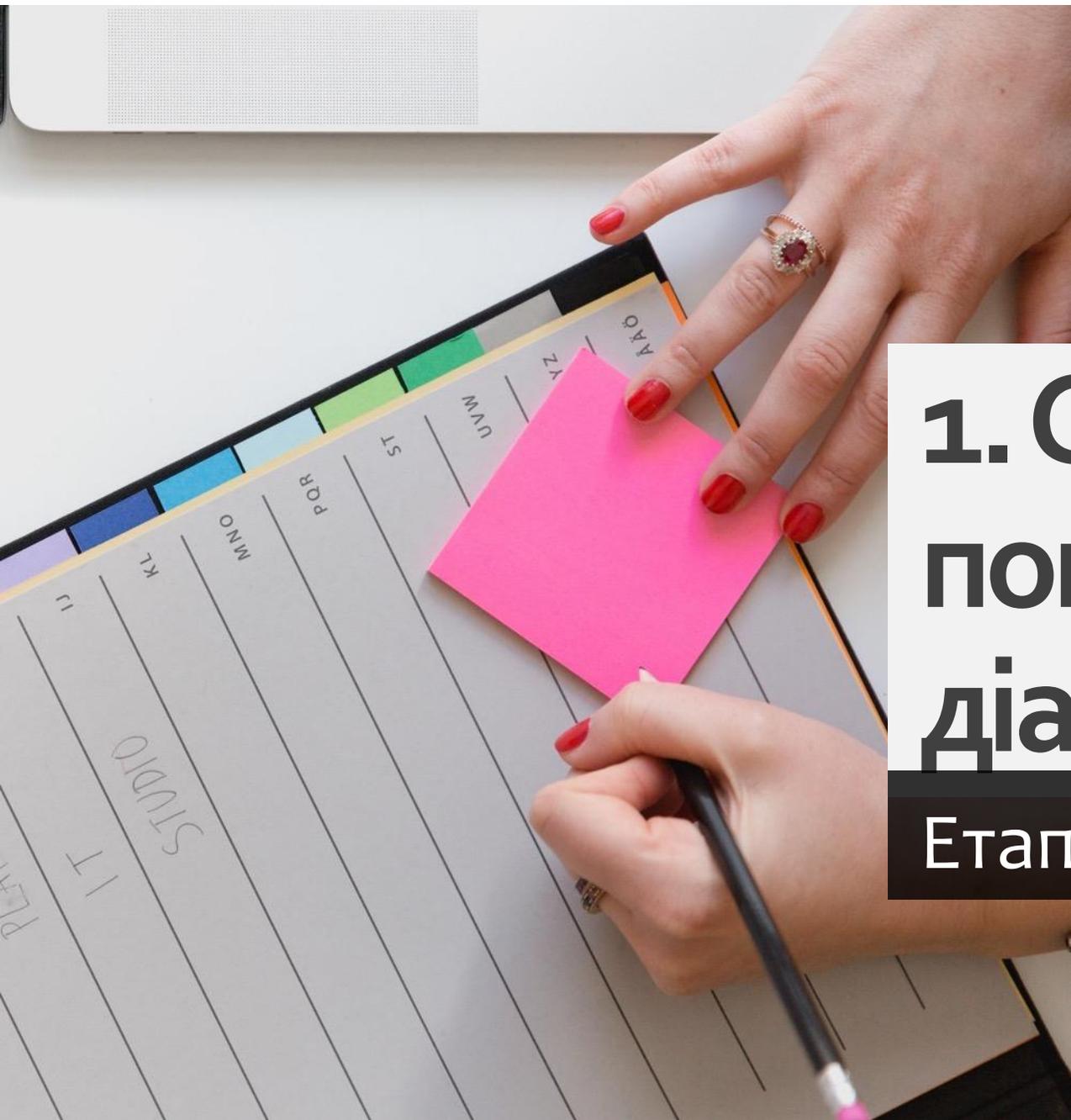
Глазунова О.Г.,
д.п.н., проф.

1. Статичні та поведінкові діаграми UML
2. Діаграма станів
3. Діаграма послідовності
4. Кейси

План

Посилання на ЕНК:

<https://elearn.nubip.edu.ua/course/view.php?id=231>



1. Статичні та поведінкові діаграми UML

Етап проектування

[Градї Буч](#) (BOOCH)

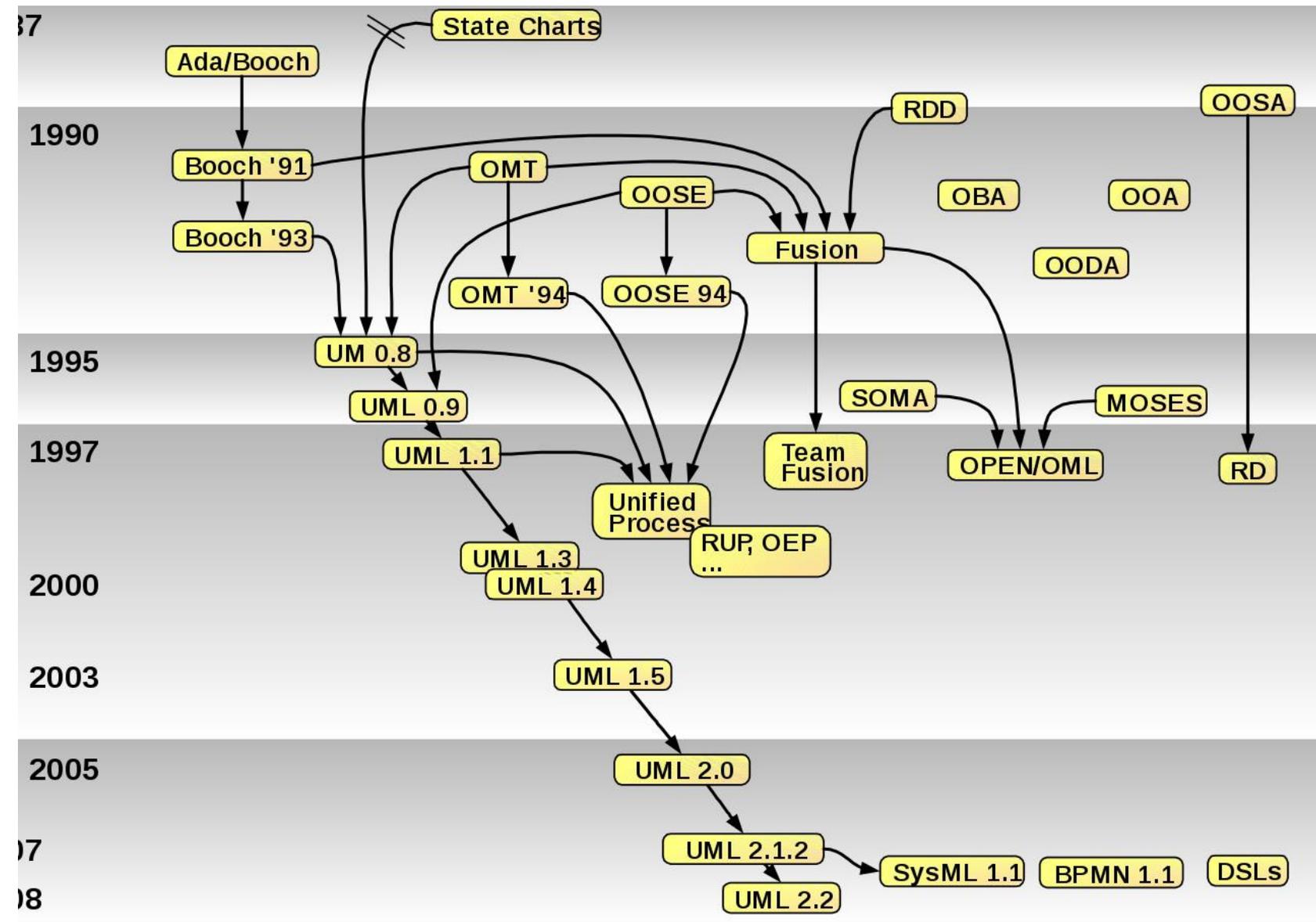
[Джим Рамбо](#) (OMT)

[Айвар Якобсон](#) (OOSE)

Версія	Рік
1.1.	1997
1.3.	2000
1.4.	2002
1.5.	2003
2.0.	2005
2.1.	
2.1.1.	2007
2.1.2.	2007
2.2.	2009
2.3.	2010
2.4.	2011
2.5.	2015
2.5.1.	2017

Автори UML

- Грейді Буч
- Джеймс Рамбо
- Айвар Якобсон



Загальна характеристика UML

Історія розвитку

Архітектурний базис UML



- Базові поняття – сутності, відношення та діаграми
- Сутності – абстракції, які використовуються в моделях UML
(структурні, поведінкові, групування та анотаційні)

Загальна характеристика UML

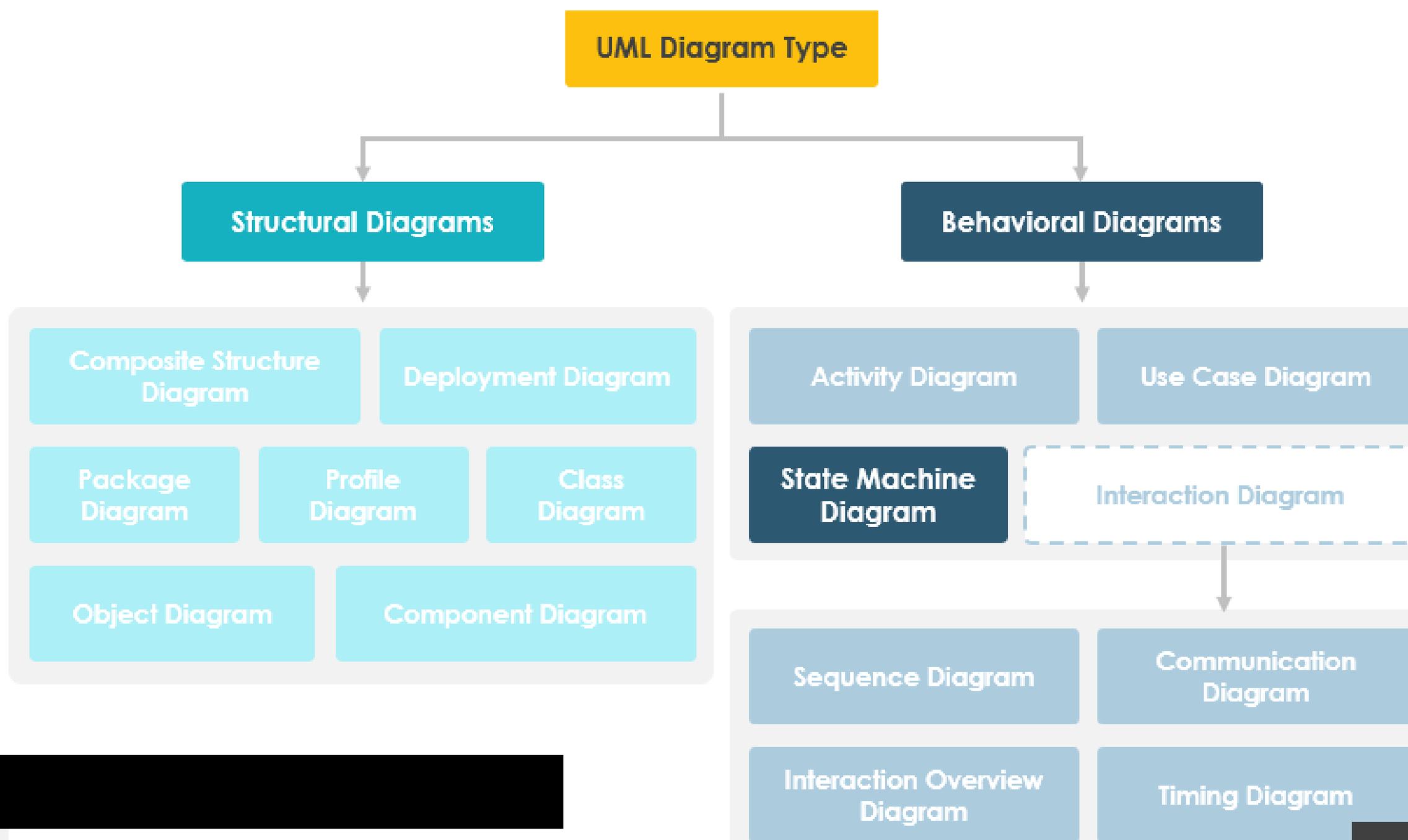
UML діаграми – поведінкові та статичні



Детальний план системи

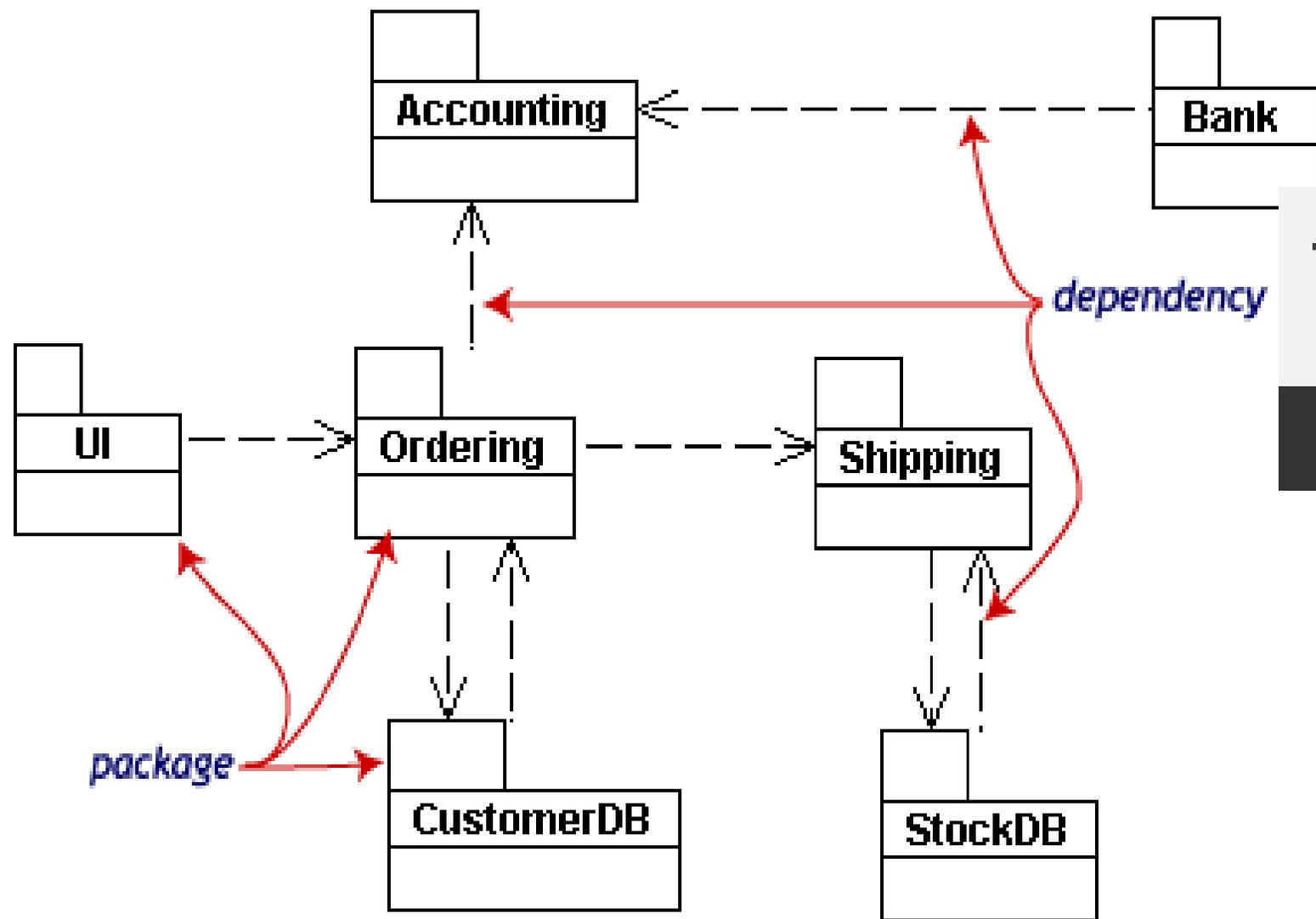
Концептуальний рівень (системні функції, бізнес-процеси)

Рівень реалізації (класи, поведінка, послідовність, взаємодія)



Діаграми UML

- *Діаграми класів (class diagram)* – зображають класи, інтерфейси, об'єкти і кооперації, а також відношення між ними. Ці діаграми відображають *статичні* аспекти системи.
- *Діаграми об'єктів (object diagram)* – зображають об'єкти і відношення між ними. Це *статичні* знімки екземплярів сутностей, зображених на діаграмах класів.
- *Діаграми прецедентів (use case diagram)* – зображають прецеденти й акторів, а також відношення між ними. Ці діаграми відображають *статичні* аспекти системи.



Типи класів

- Сутність
- Управляючий
- Інтерфейсний

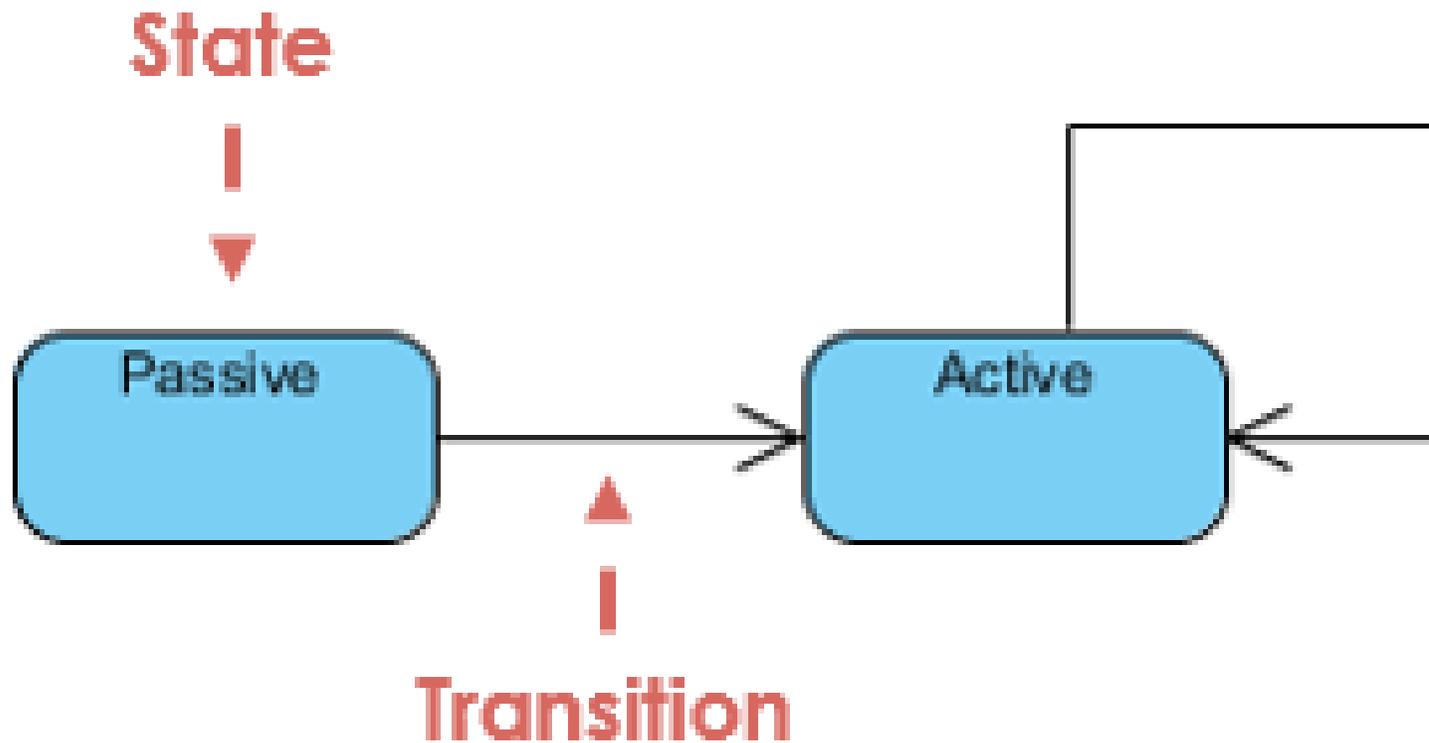
Діаграми UML

- *Діаграми взаємодії* – зображають об'єкти та повідомлення, якими об'єкти можуть обмінюватися. Зазвичай, розглядають два часткові випадки таких діаграм: *діаграми послідовностей* (sequence diagram), що відображають *часову* упорядкованість повідомлень, і *діаграми кооперації* (collaboration diagram), на яких зображають структурну організацію об'єктів, що обмінюються повідомленнями. Ці типи діаграм є ізоморфними. Діаграми взаємодії відображають *динамічні* аспекти системи.
- *Діаграми станів* (statechart diagram) – зображають автомат, що налічує стани, переходи, події і види дій. Ці діаграми відображають *динамічні* аспекти системи.

Діаграми UML

- *Діаграми станів (statechart diagram)* – зображають автомат, що налічує стани, переходи, події і види дій. Ці діаграми відображають *динамічні* аспекти системи.
- *Діаграми діяльностей (activity diagram)* – це окремий випадок діаграми станів (на діаграмі зображають переходи потоку керування від одного виду діяльності до іншого усередині системи).
- *Діаграми компонентів (component diagram)* – зображають сукупність компонентів та існуючі між ними залежності. Ці діаграми відображають *статичні* аспекти системи.
- *Діаграми розміщення (deployment diagram)* – зображають конфігурацію вузлів системи і розміщених у них компонентів. Ці діаграми відображають *статичні* аспекти системи. Вони зв'язані з діаграмами компонентів, оскільки у вузлі розміщують один чи декілька компонентів.

2. Діаграма станів



- Діаграма станів показує, як об'єкт переходить з одного стану в інший.

Стан об'єкта

- Стан (state) - це якесь положення в житті об'єкта, при якому він задовольняє визначеній умові, виконує деяку дію або очікує події. Поточний стан об'єкта можна описати сукупністю поточних значень одного або декількох атрибутів класу та зв'язків об'єкта .

Стабільний стан являє собою стан, при якому об'єкт може існувати протягом певного проміжку часу.

Коли відбувається подія, об'єкт може переходити із стану в стан. Ці події можуть також спричинити самостійні та внутрішні переходи, в яких джерело та ціль переходу є однаковими.

У відповідь на подію або зміну стану об'єкт може відповісти, надіславши дію. Стани об'єкта залежать від значень його атрибутів та посилань на інші об'єкти.

Стан об'єкта

Переходи

Переходи між станами (state transition) являють собою зміну одного стану іншим.

Перебування об'єкта, що моделюється, в першому стані може супроводжуватися виконанням деяких дій, а перехід у другий стан буде можливий після завершення цих дій, а також після задоволення деяких додаткових умов. У цьому випадку вважають, що перехід спрацьовує.

До спрацьовування переходу об'єкт знаходиться в попередньому від нього стані, а після його спрацьовування об'єкт знаходиться в наступному від нього стані (цільовому стані).

Позначки

Перелік позначок дії має фіксовані значення в мові UML, що не можуть бути використані як імена подій. Ці значення наступні:

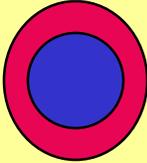
entry - ця позначка вказує на дію, що виконується в момент входу в даний стан (вхідна дія);

exit - ця позначка вказує на дію, що виконується в момент виходу з даного стану (вихідна дія);

do - ця позначка специфікує діяльність ("do activity"), що виконується протягом усього часу, поки об'єкт знаходиться в даному стані, або доти, поки не закінчиться обчислення. В останньому випадку при завершенні події генерується відповідний результат;

include - ця позначка використовується для звертання до підавтомату, при цьому наступне за нею вираження дії містить ім'я цього підавтомата.

Синтаксис

Стан	
Початковий стан	
Кінцевий стан	
Подія	anEvent
Перехід	

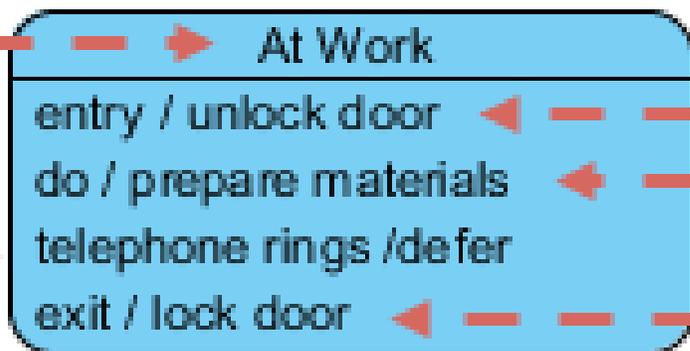
Компоненти

- Стани
- Події
- Переходи
- Дії

Нотація діаграми станів

State Name

Name of state



Deferrable Trigger

Event that does not trigger any state transition, but remain in the event pool ready for processing when the object transitions to another state

Entry Activity

Action performed on entry to state

Do Activity

Action performed while in state

Exit Activity

Action performed on leaving state

Графічне зображення станів

Введення паролю

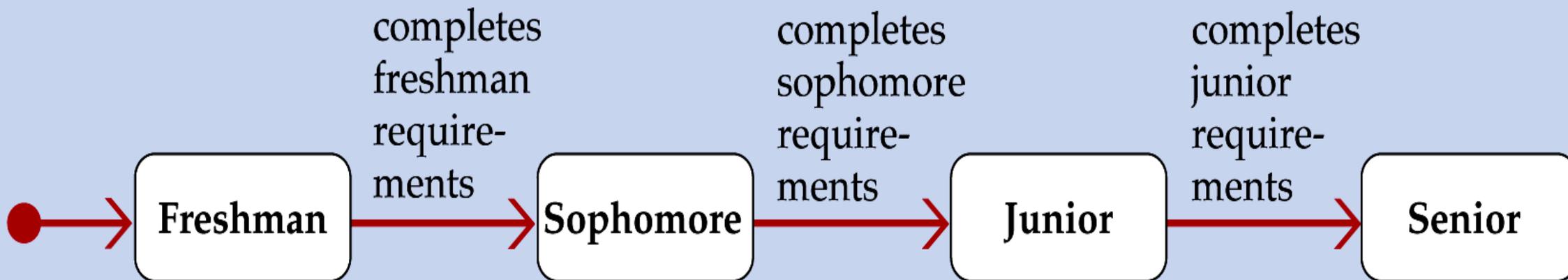
`entry/` Відключити відображення символів

`exit/` Встановити режим відображення символів

`do/` Прочитати символ

`do/` Вивести підказку

Matriculated



Кожен об'єкт є екземпляром класу і має повний життєвий цикл, від створення до знищення.

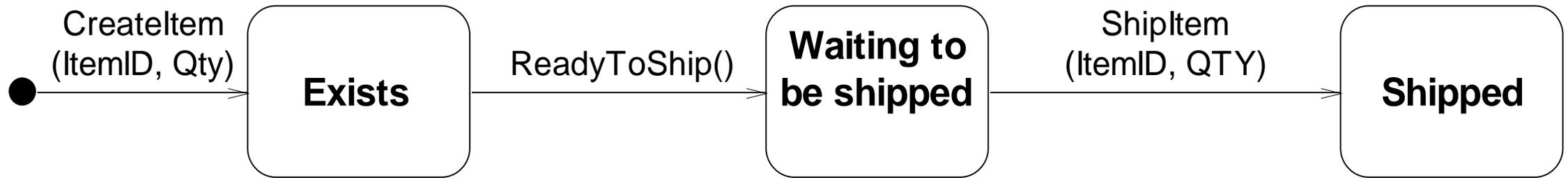
Об'єкт виникає в системі.

Протягом періоду свого існування він перебуває у “певних” станах і здійснює перехід від одного стану до іншого.

Розуміння станів

Стани і переходи: розуміння

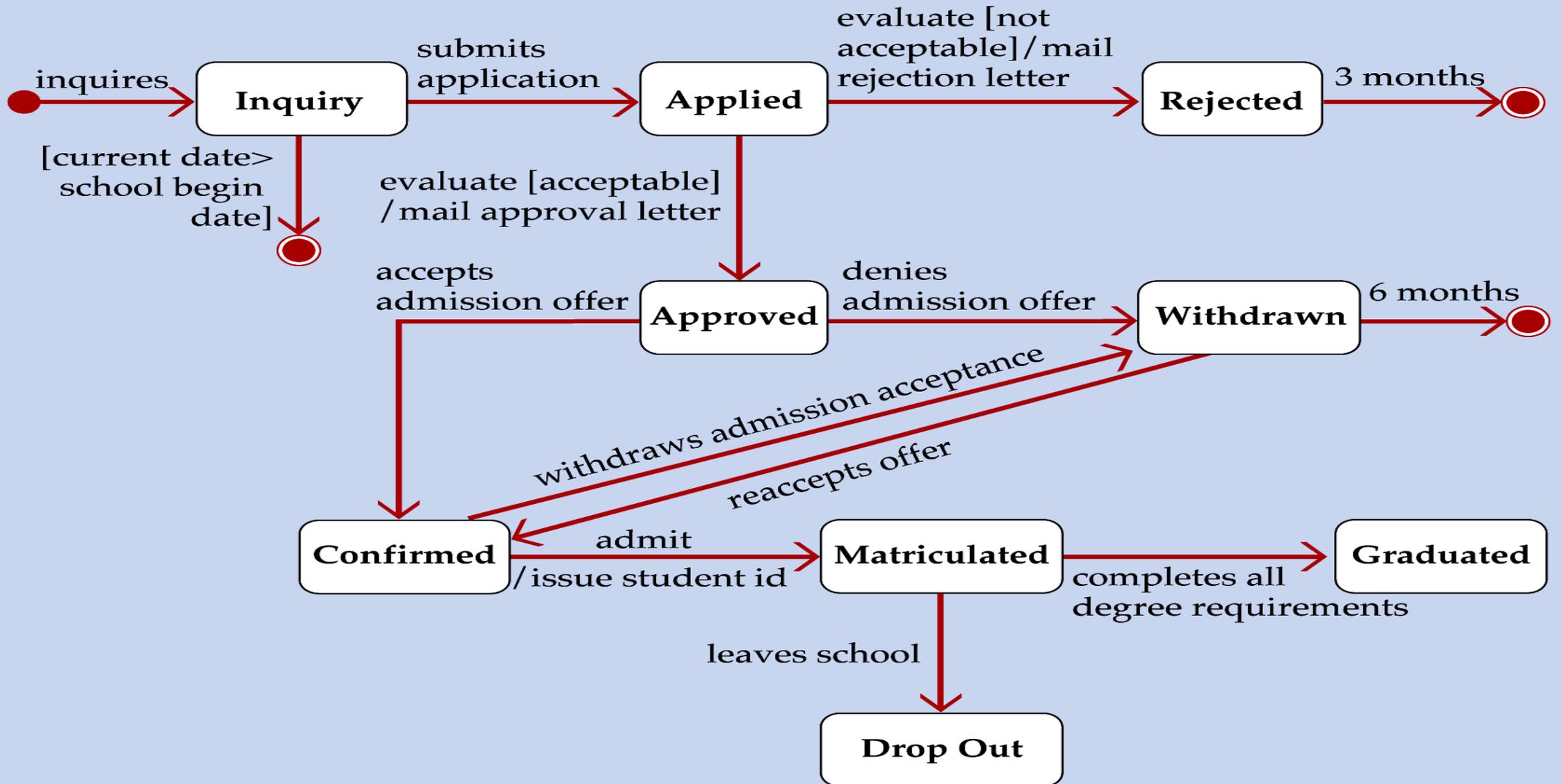
.



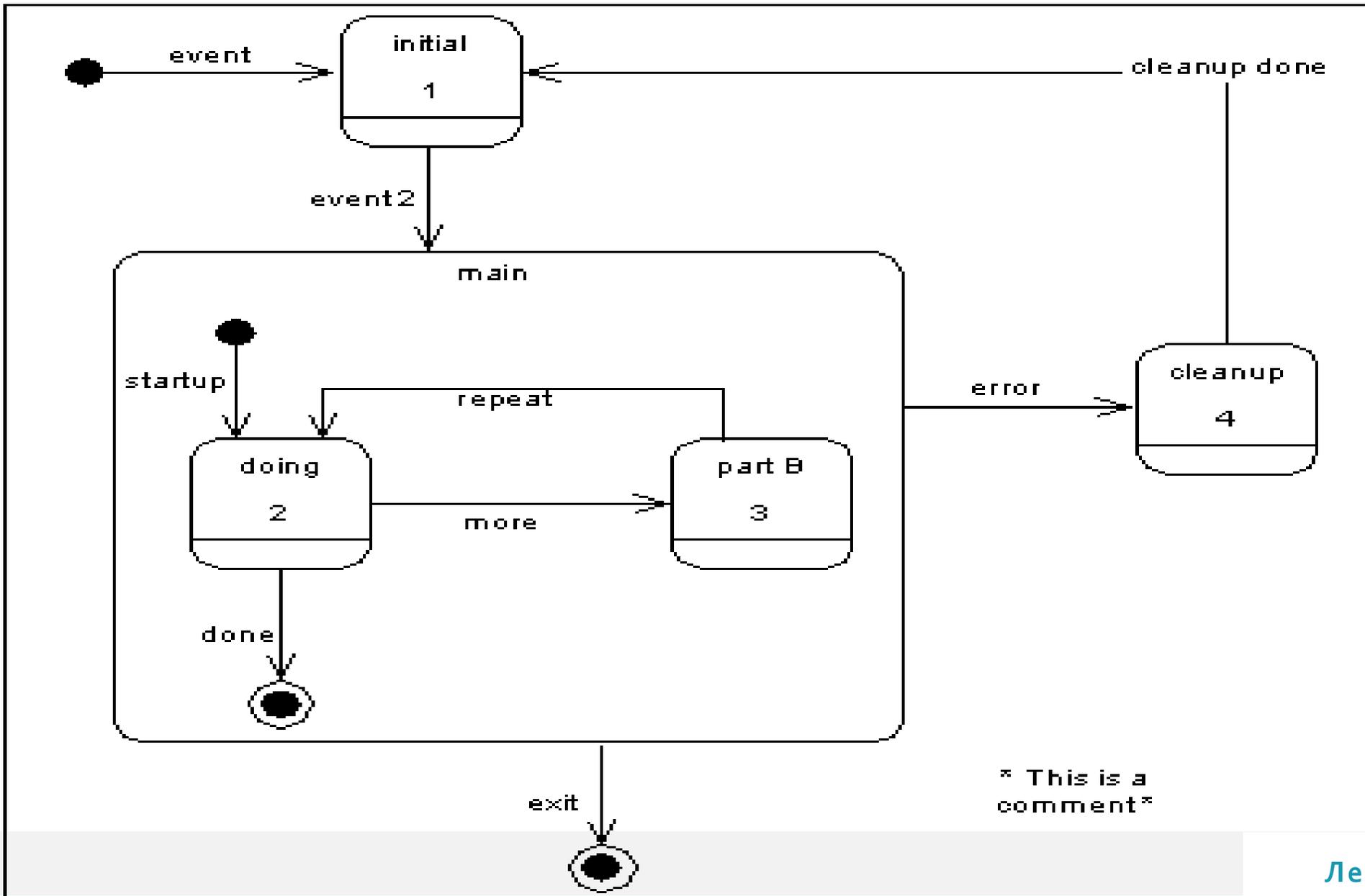
1. Коли у вас є складний об'єкт і не впевнені, як це буде функціонувати.
2. Коли ДІЙСНО потрібно знати стан об'єкта тому що це життєво важливо для успіху програми, над якою ви працюєте.
3. Діаграми чудово підходять для моделювання поведінки об'єктів користувальницького інтерфейсу, щоб ви могли легко кодувати з них.

**Коли
використовувати
діаграму станів**

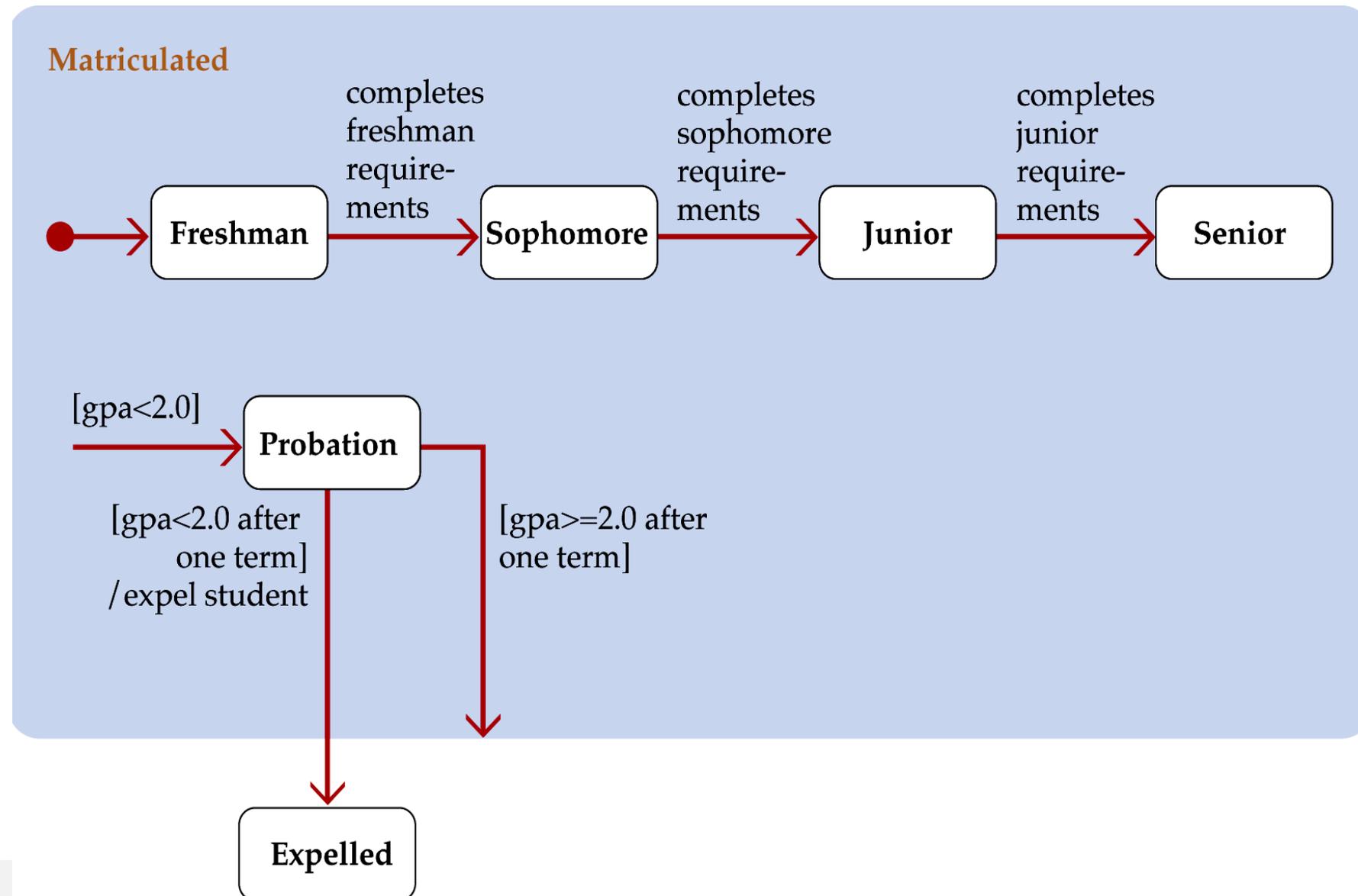
Завдання - «прочитати» діаграму



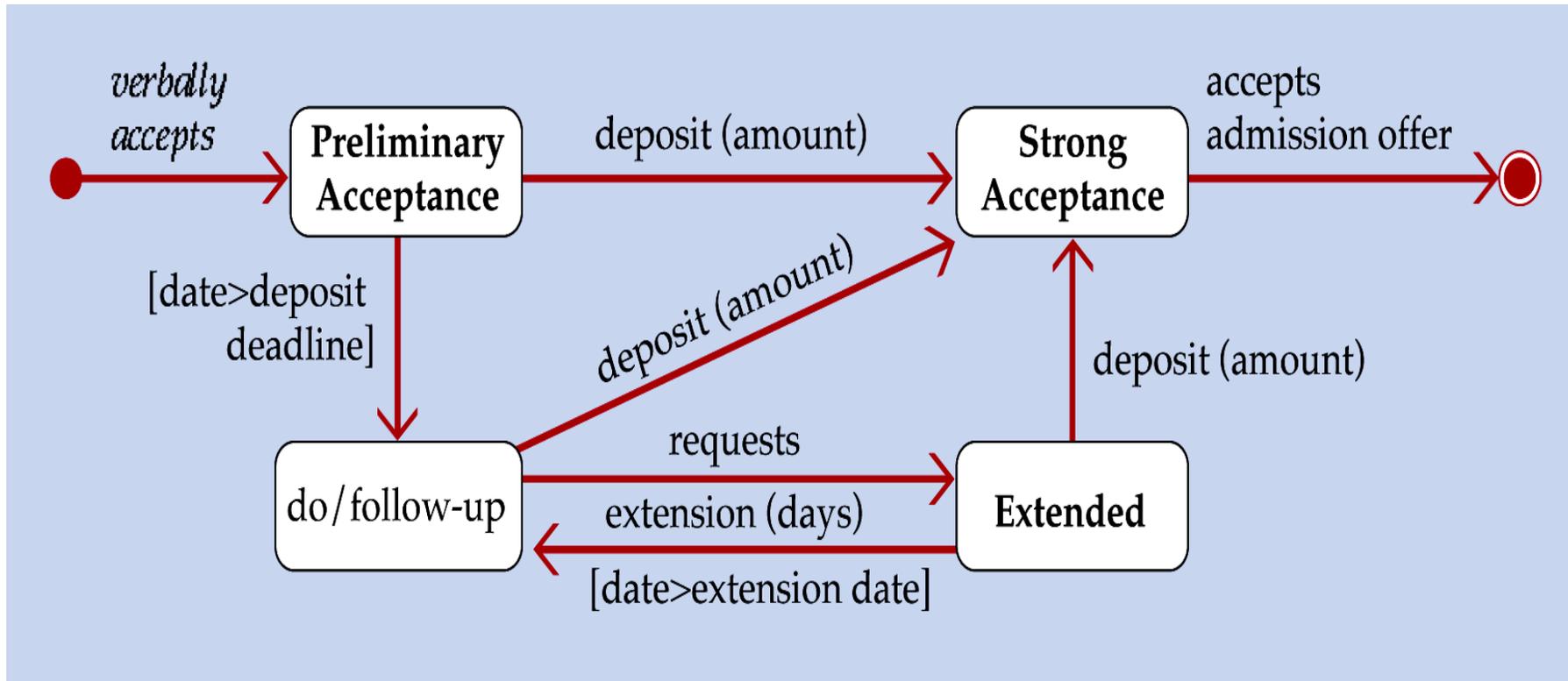
Групування



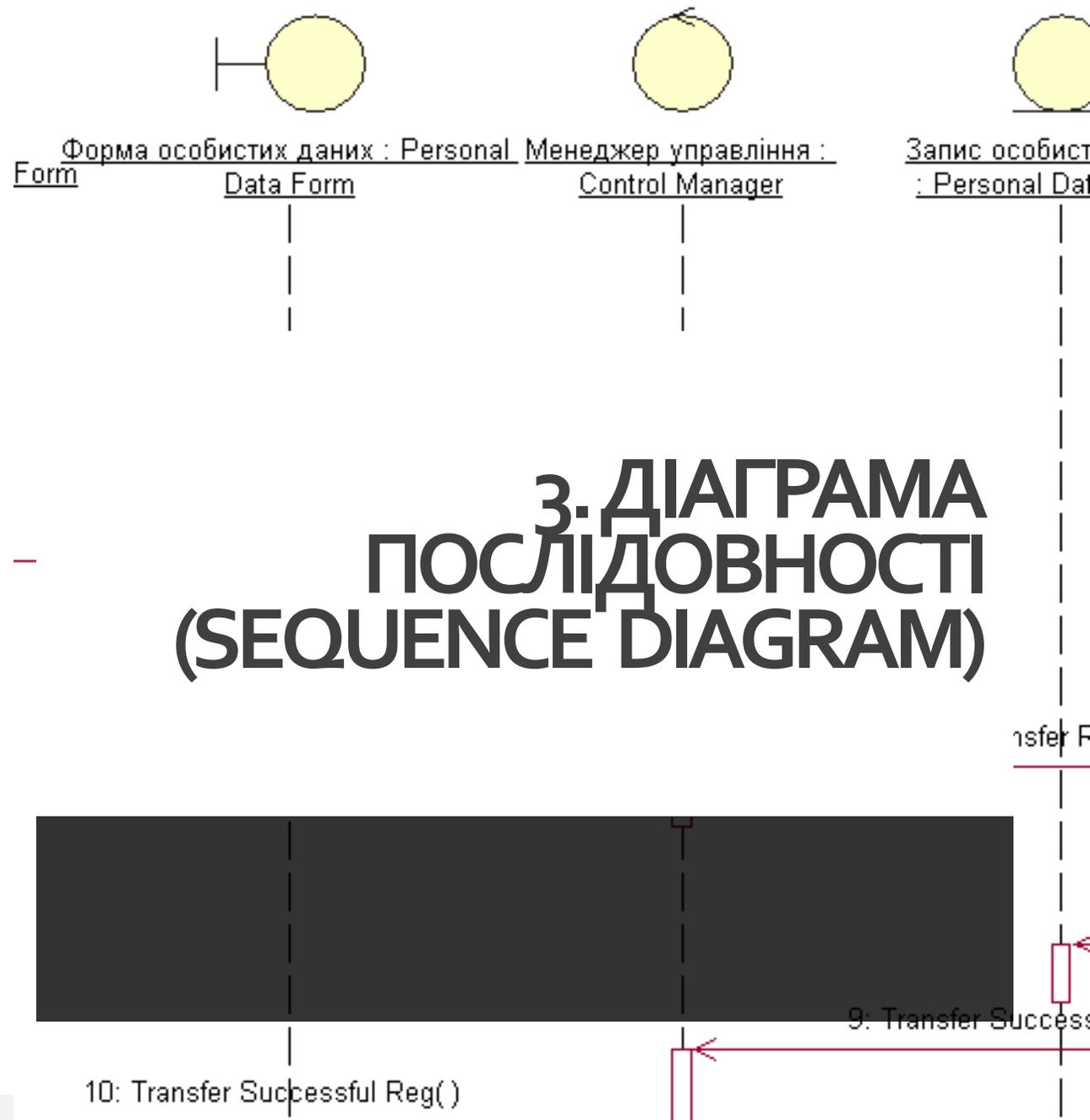
Nested State Diagram for Matriculated State of Student Object



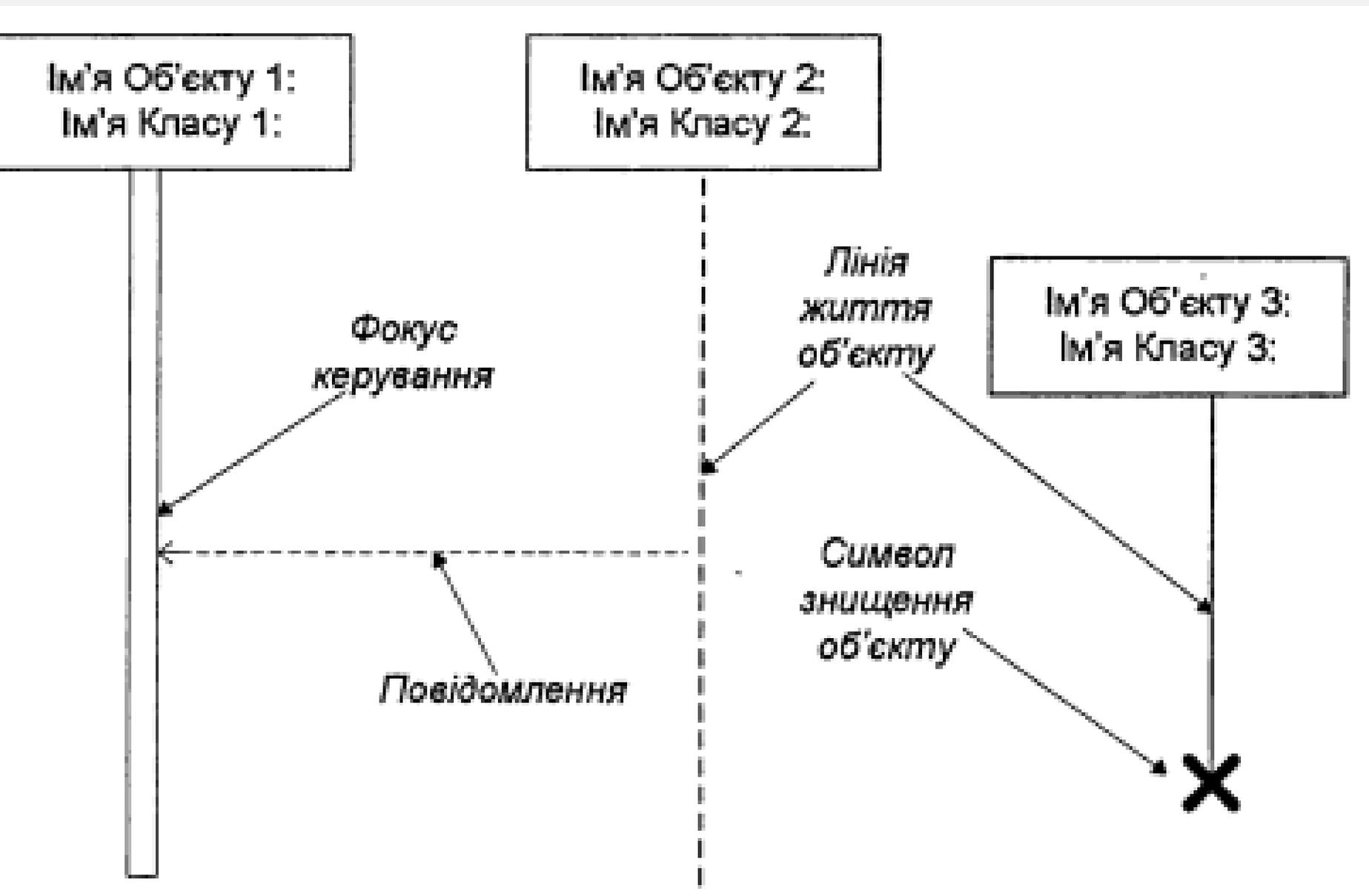
Nested State Diagram for the Accepts Admission Offer Event



- Часовий аспект поведінки може мати істотне значення під час моделювання синхронних процесів, що описують взаємодію об'єктів.
- На діаграмі послідовності зображаються винятково ті об'єкти, які безпосередньо беруть участь у взаємодії і не показуються можливі статичні асоціації з іншими об'єктами.
- Діаграма послідовності має як би два виміри. Один - зліва направо у вигляді вертикальних ліній, кожна з яких зображає лінію життя окремого об'єкту, що бере участь у взаємодії. Графічно кожний об'єкт зображається прямокутником і розташовується у верхній частині своєї лінії життя



3. ДІАГРАМА ПОСЛІДОВНОСТІ (SEQUENCE DIAGRAM)



Приклад

Горизонталь

- Крайнім ліворуч на діаграмі зображається об'єкт, що є ініціатором взаємодії
- Правіше зображається інший об'єкт, що безпосередньо взаємодіє з першим.
- Таким чином всі об'єкти на діаграмі послідовності утворюють деякий порядок, обумовлений ступенем активності цих об'єктів під час взаємодії один з одним.

Вертикаль

- Другий вимір діаграми послідовності - вертикальна тимчасова вісь, спрямована зверху вниз.
- Початковому моменту часу відповідає найвища частина діаграми.
- Взаємодії об'єктів реалізуються за допомогою повідомлень, які посилають одні об'єкти іншим.

Повідомлення

- Повідомлення зображаються у вигляді горизонтальних стрілок з іменем повідомлення і також утворюють порядок відповідно до часу свого виникнення.
- Масштаб на осі часу не вказується, оскільки діаграма послідовності моделює лише тимчасову впорядкованість взаємодій типу "пізніше".

Лінія життя об'єкту

- Лінія життя об'єкту (object lifeline) зображається пунктирною вертикальною лінією, що асоціюється з єдиним об'єктом на діаграмі послідовності.
- Лінія життя служить для позначення періоду часу, протягом якого об'єкт існує в системі й, отже, може потенційно брати участь у всіх її взаємодіях.

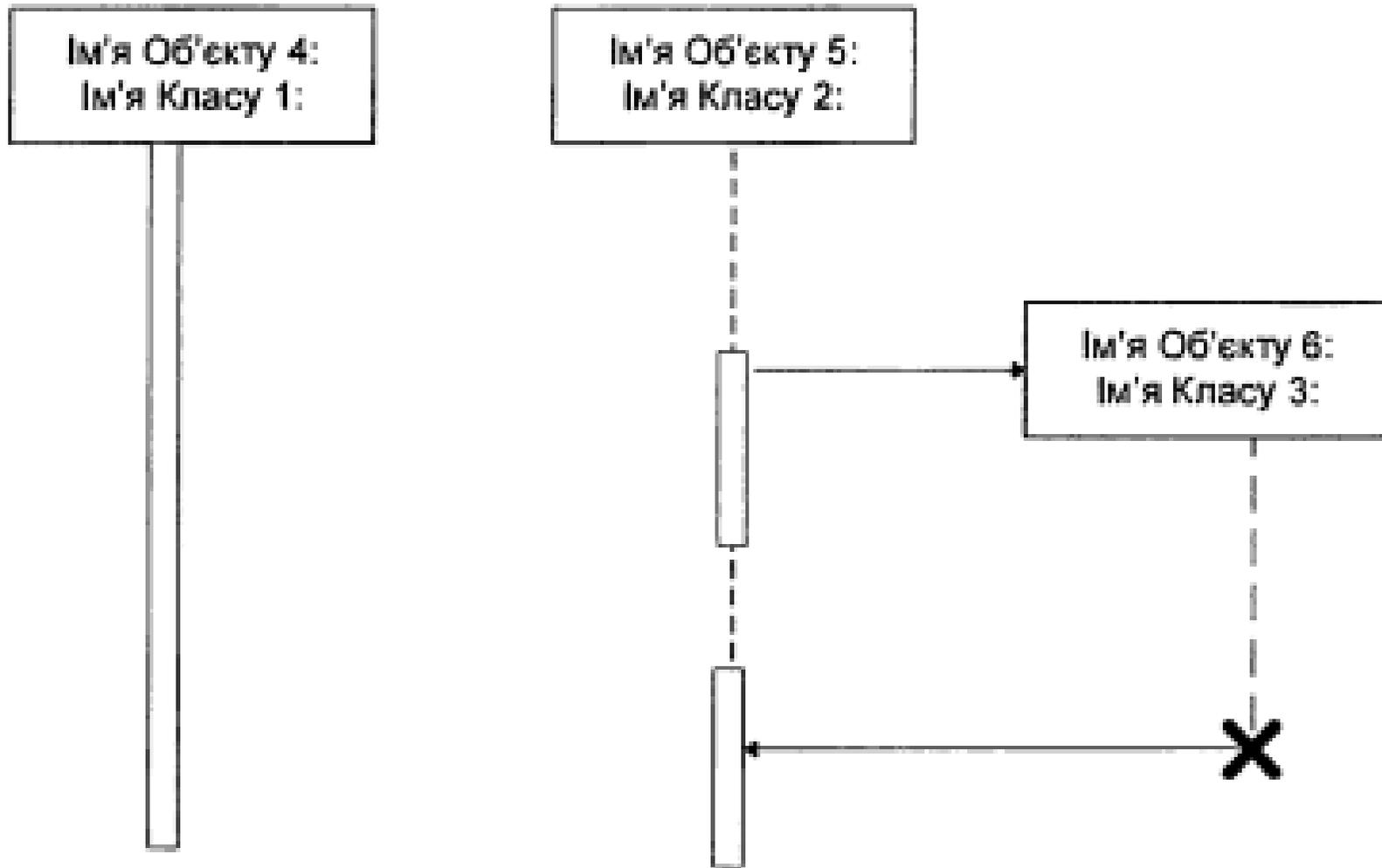
Знищення об'єктів

- Окремі об'єкти, виконавши свою роль у системі, можуть бути знищені (зруйновані), щоб звільнити займані ними ресурси. Для таких об'єктів лінія життя обривається в момент його знищення.

Фокус керування

- У процесі функціонування об'єктно-орієнтованих систем одні об'єкти можуть перебувати в активному стані, безпосередньо виконуючи певні дії або у стані пасивного очікування повідомлень від інших об'єктів.
- Щоб явно виділити подібну активність об'єктів, у мові UML застосовується спеціальне поняття - фокус керування (focus of control).
- Фокус керування зображається у формі витягнутого вузького прямокутника.

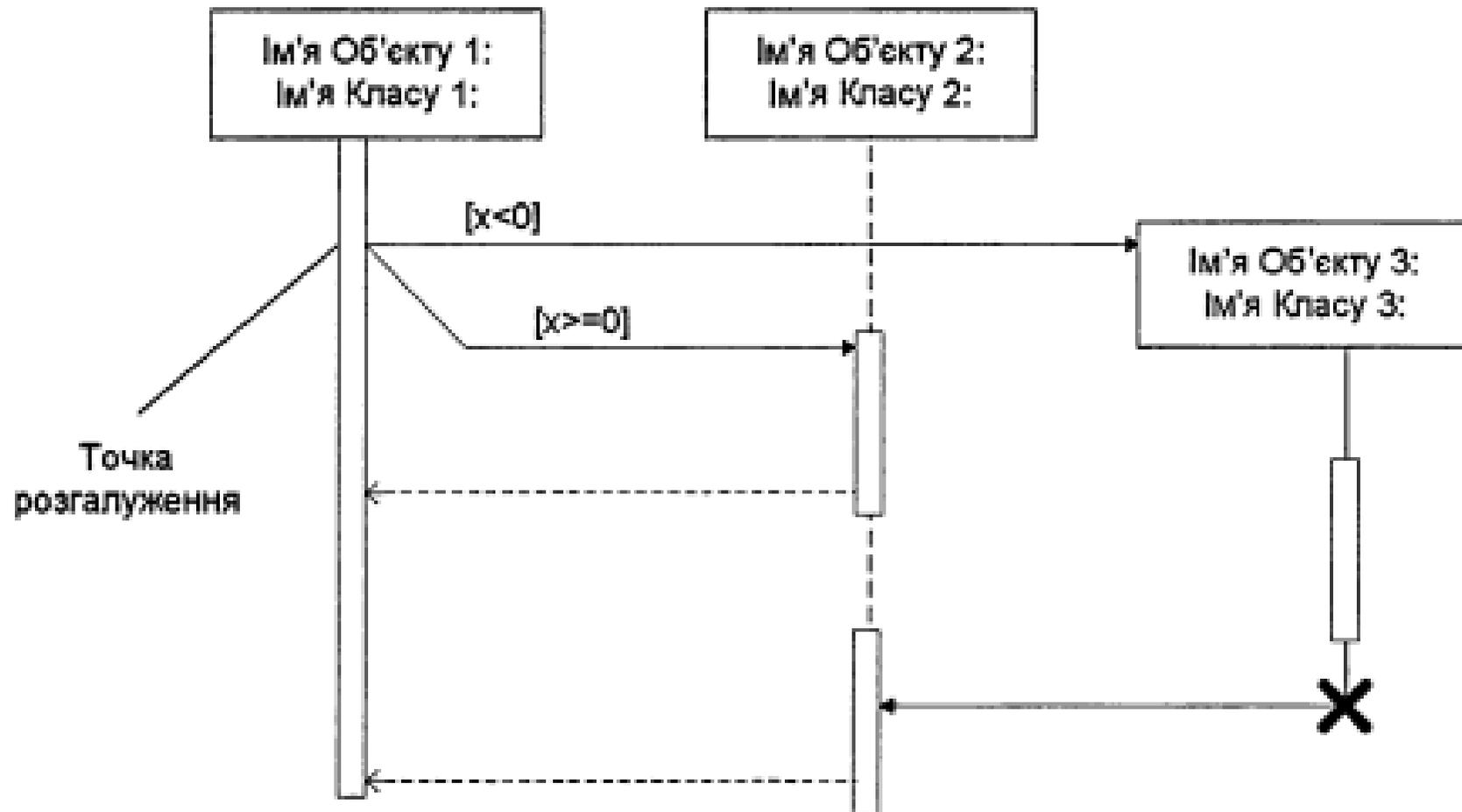
Приклад



Повідомлення

- Взаємодія в контексті мови UML полягає в тому, щоб специфікувати комунікацію між множиною об'єктів, які взаємодіють.
- Кожна взаємодія описується сукупністю повідомлень, якими об'єкти обмінюються між собою.
- У цьому сенсі повідомлення (message) є закінченим фрагментом інформації, що відсилається одним об'єктом іншому.
- Приймання повідомлення ініціює виконання певних дій, спрямованих на рішення окремого завдання тим об'єктом, якому це повідомлення відіслане.

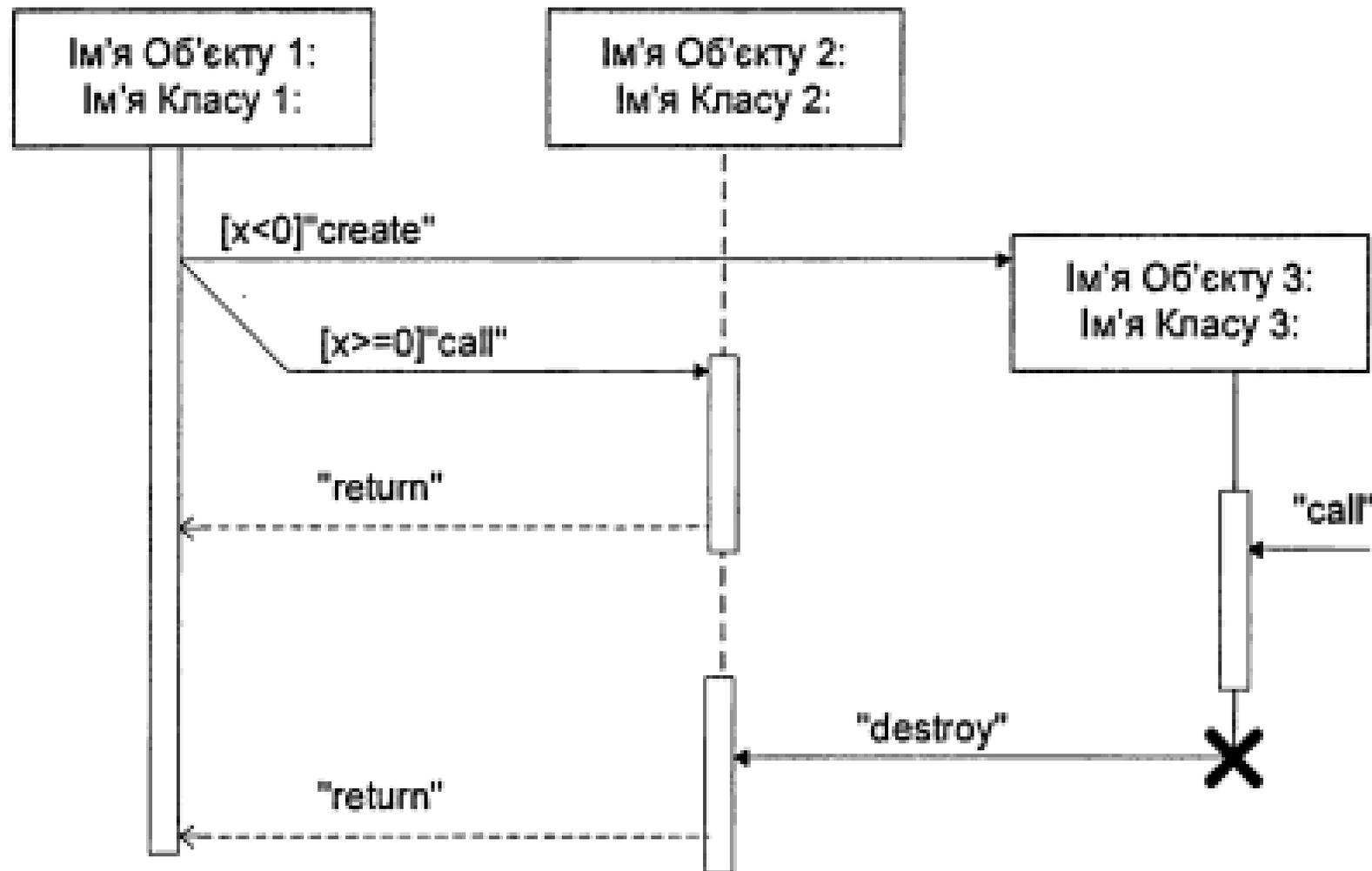
Розгалуження



Стереотипи повідомлень

- "call" (викликати) - повідомлення, що викликає операції або процедури об'єкту, який приймає;
- "return" (повернути) - повідомлення, що повертає значення виконаної операції або процедури об'єкту, якщо її викликав;
- "create" (створити) - повідомлення, що вимагає створення іншого об'єкту для виконання певних дій;
- "destroy" (знищити) - повідомлення з явною вимогою знищити відповідний об'єкт;
- "send" (послати) - позначає посилення іншому об'єкту деякого сигналу, що асинхронно ініціюється одним об'єктом і приймається (перехоплюється) іншим;

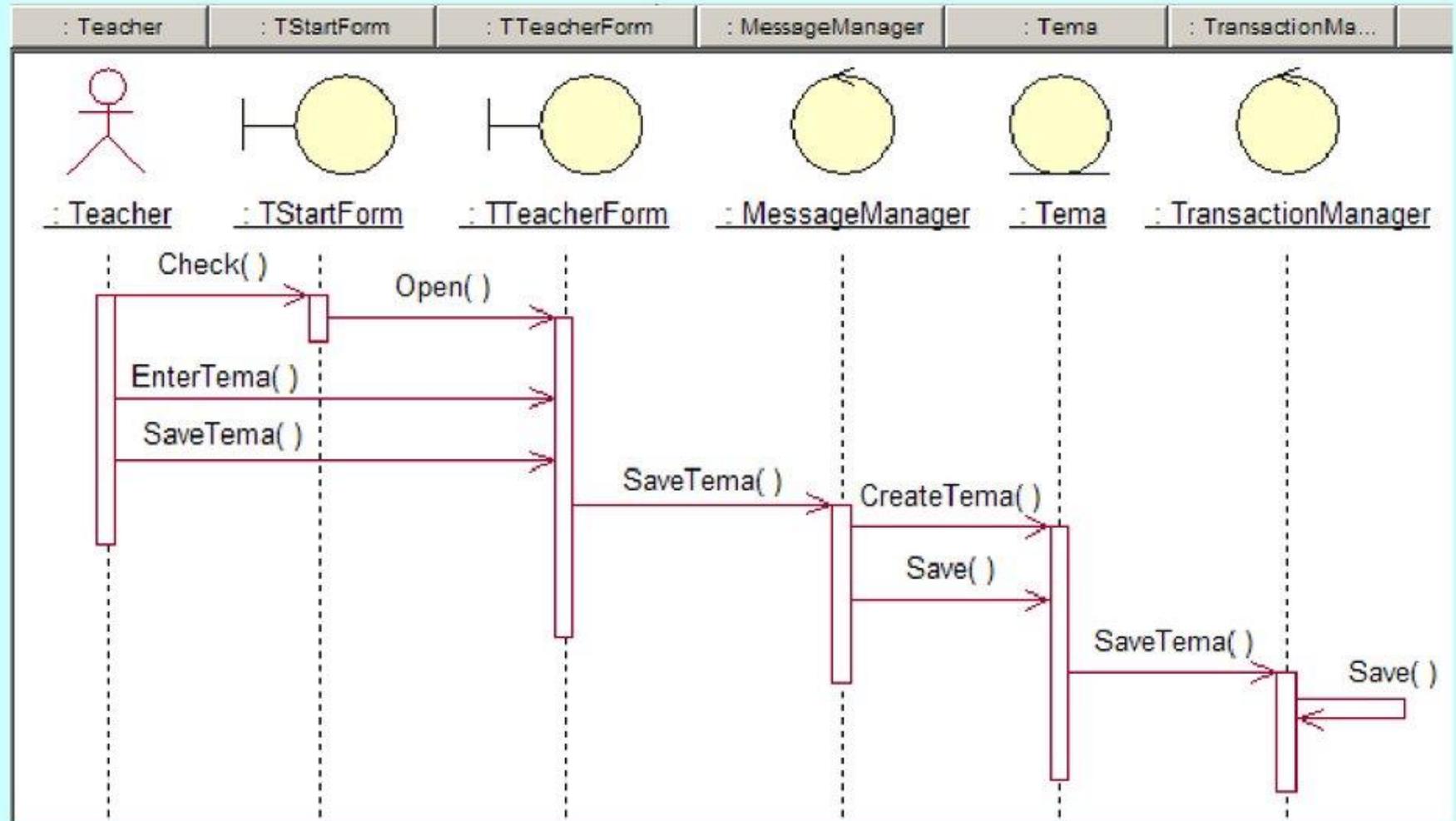
Стереотипи повідомлень

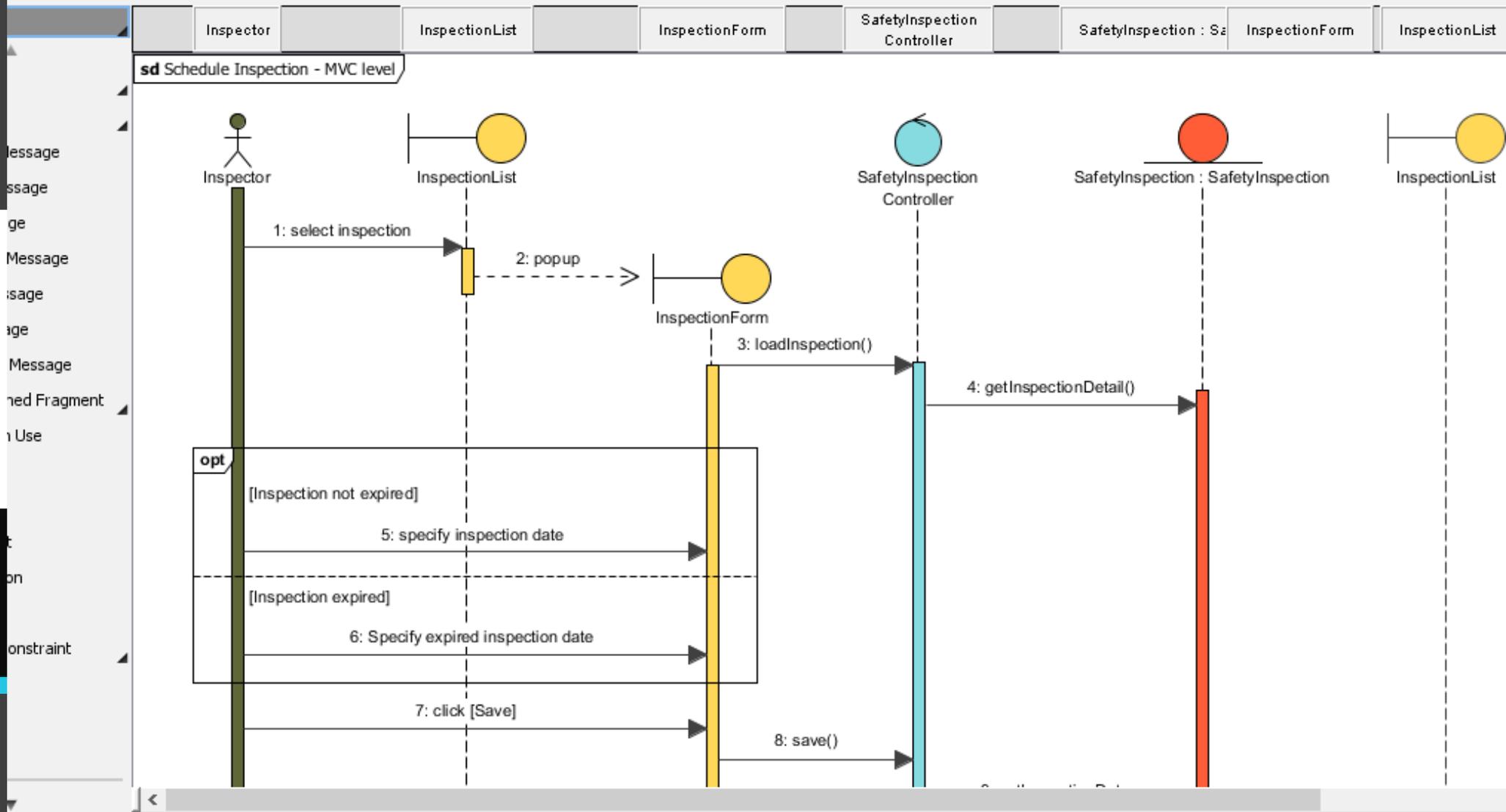


4. Кейси

Діаграма послідовності для сценарію "Викладач формулює тему дипломної роботи" (другий етап)

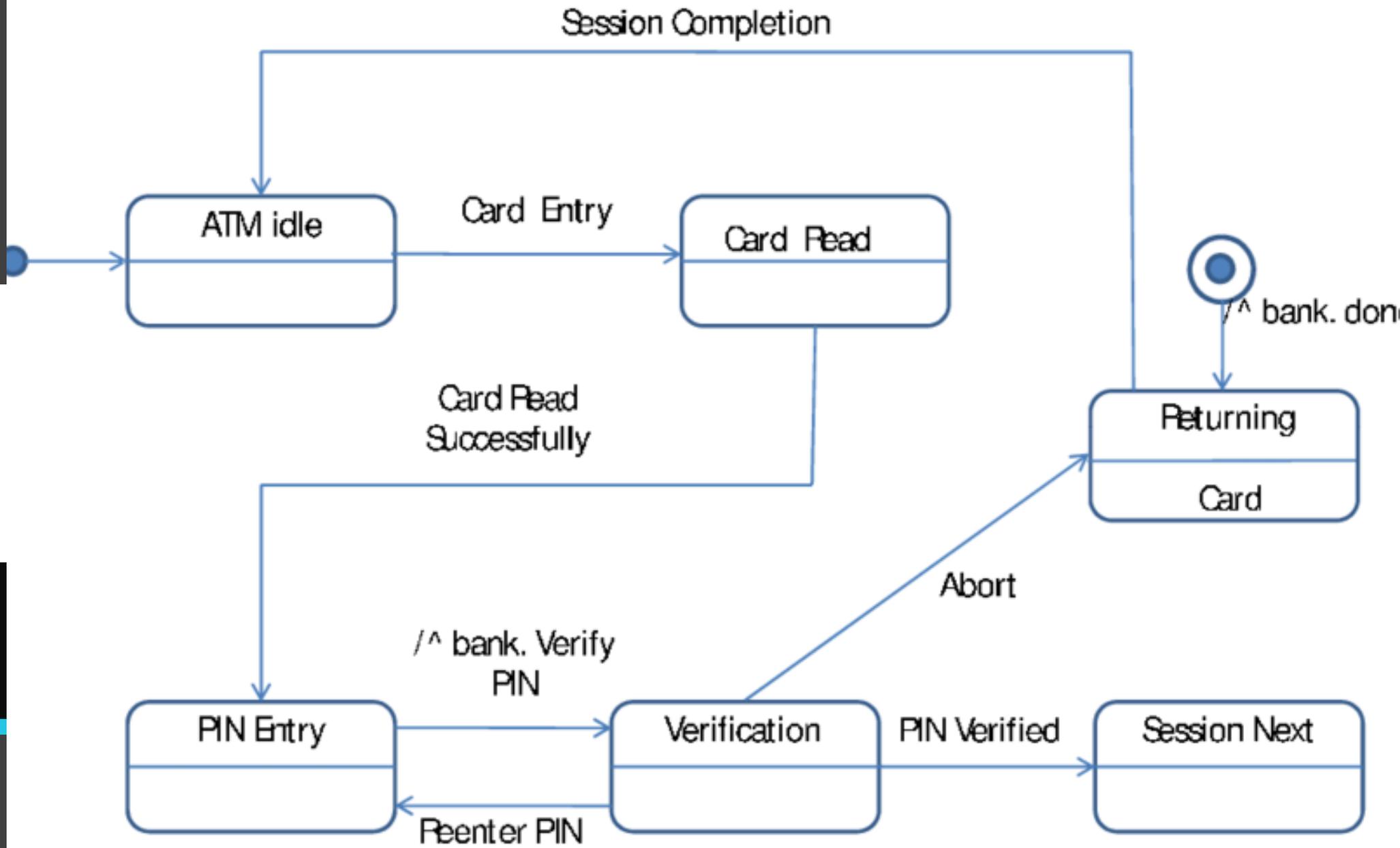
1.





2.

3.





Запитання?

