



## ЛЕКЦІЯ 16

Засоби UML для моделювання в програмній інженерії

- **Метод UML** (Unified Modelling Language — уніфікована мова моделювання) є рідкісним прикладом плідної кооперації групи (G. Booch, I. Jacobson, J. Rumbaugh) провідних спеціалістів з програмної інженерії і авторів відповідних методів інженерії вимог, що набули значного визнання і широко застосовуються.
- UML став базовим для багатьох провідних розробників програмного забезпечення, і тепер експерти прогнозують, що він набуде статусу міжнародного стандарту як метод моделювання продуктів усіх стадій життєвого циклу розробки програмних систем.
- Автори визначають свій метод як мову для специфікації, візуалізації, конструювання й документування артефактів програмних систем, а також для моделювання бізнесу.

## КОНЦЕПЦІЯ МЕТОДУ

- В основу методу покладено парадигму об'єктного підходу, за якою концептуальне моделювання проблеми відбувається в термінах взаємодії об'єктів:
- онтологія домену визначає склад класів об'єктів домену, їхніх атрибутів та взаємовідношень, а також послуг (операцій), які можуть виконувати об'єкти класів;
- модель поведінки визначає можливі стани об'єктів, інциденти, що ініціюють переходи з одного стану в інший, повідомлення, які об'єкти надсилають одне одному;
- модель процесів визначає дії, які виконують об'єкти.

- В комплексі сукупність включених до методу діаграм відображає найважливіші випадки функціонування системи:
  - 1) діаграми класів (Class diagram);
  - 2) діаграми сценаріїв (Use-case diagram);
  - 3) діаграми поведінки об'єктів, а саме:
    - а) діаграми послідовності (Sequence diagram);
    - б) діаграми співробітництва (Collaboration diagram);
    - в) діаграми активності (Activity diagram);
    - г) діаграми станів (Statecards diagram);
  - 4) діаграми реалізації, а саме:
    - а) діаграми компонент;
    - б) діаграми розміщення (Deployment diagram).

- Більшість діаграм може відображати кількома ступенями подробиць об'єкти кількох етапів: об'єкти аналізу вимог, проекту, реалізації тощо.
- Для цього передбачено можливість вказувати або замовчувати (залежно від стадії розробки) окремі подробиці визначення.
- Кожний вид діаграм відображає різні перспективи бачення й розуміння моделі. У моделі може бути по кілька діаграм кожного з описаних видів.

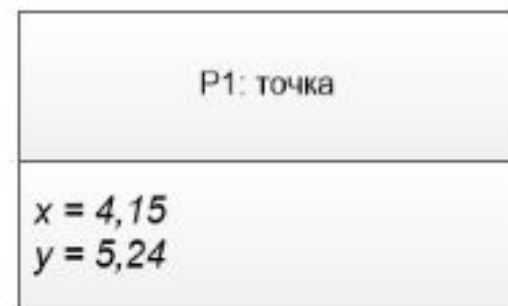
- **Діаграма має вигляд символів класів** — так званих ікон та зв'язків між ними.
- Терміном ікона позначають стандартизоване, фіксованої форми, візуальне зображення (так би мовити, ієрогліф) певного поняття, яке легко розпізнається.
- Ікона класу має форму прямокутника, який може поділятися на дві або три частини. Верхня його частина обов'язкова, вона містить ім'я класу. Друга й третя частини прямокутника можуть наводитися або пропускатися і містять: друга — список атрибутів класу, третя — список операцій класу



- Атрибути можуть бути такими, типи значень яких вважаються наперед визначеними в UML, як-от: розмір, площа, кут, видимість. Останній атрибут може мати такі значення:
- **спільна (public)** означає, що операцію класу можна викликати з будь-якої частини програми будь-яким об'єктом системи;
- **захищена (protected)** означає, що операцію можна викликати тільки об'єктом того класу, в якому її визначено, або його спадкоємцями;
- **приватна (private)** означає, що операцію можна викликати тільки об'єктом того класу, в якому її визначено.
- Воднораз користувач може визначати специфічні для нього атрибути.

▣ **Операція** — це сервіс, який може надавати екземпляр класу, якщо буде відповідний виклик. Операція має назву і список аргументів.

На діаграмі може бути показано не лише класи, а й окремі їхні екземпляри. Може бути побудовано діаграму екземплярів класів. З метою відрізнити класи від їхніх екземплярів назви других у зображенні ікони класу підкреслюються.





- Класи можуть перебувати у певних відношеннях або зв'язках. Розглядаються бінарні асоціації, в яких об'єкт з кожної сторони має свою роль



## ДІАГРАМИ КЛАСІВ

- ▣ **Асоціація.** Це — взаємна залежність між об'єктами різних класів, кожен з яких є рівноправним членом залежності.
- ▣ Для асоціації може позначатися кількість екземплярів об'єктів кожного класу, які беруть участь у зв'язку (0 — якщо жодного, 1 — якщо один, \* — якщо багато). Можуть вказуватися мінімальна й максимальна кількість, наприклад, 0,1...\* означає, що на відповідному кінці асоціації може не бути жодного екземпляра, бути один або багато.

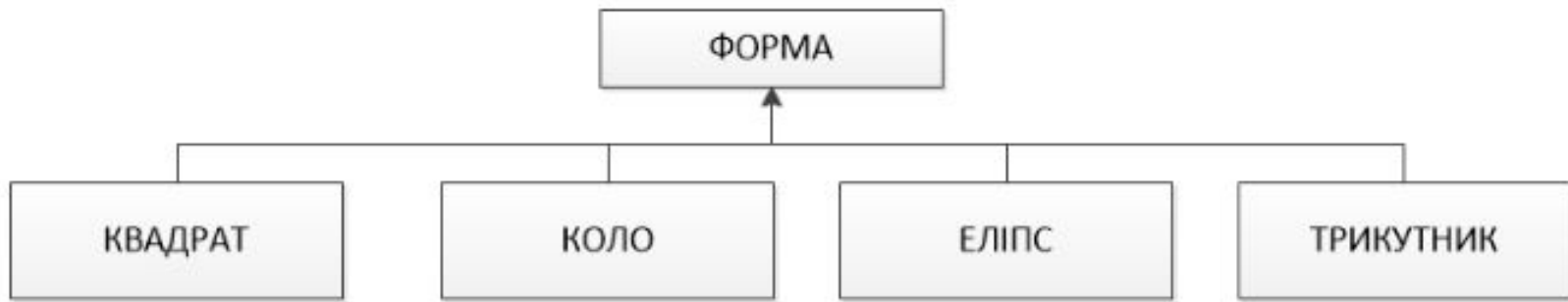


- ▣ **Агрегація** або відношення частина-до-цілого. Особливість цього відношення полягає в тому, що час існування об'єкта-частини збігається з часом існування об'єкта-цілого. Стрілка з ромбом на кінці, яка позначає відношення агрегації, спрямована від об'єкта-частини до об'єкта-цілого.

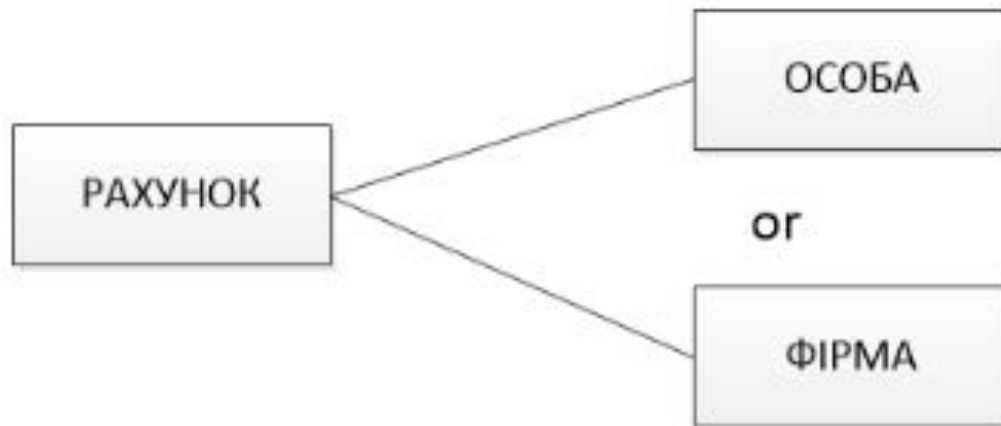


## ДІАГРАМИ КЛАСІВ

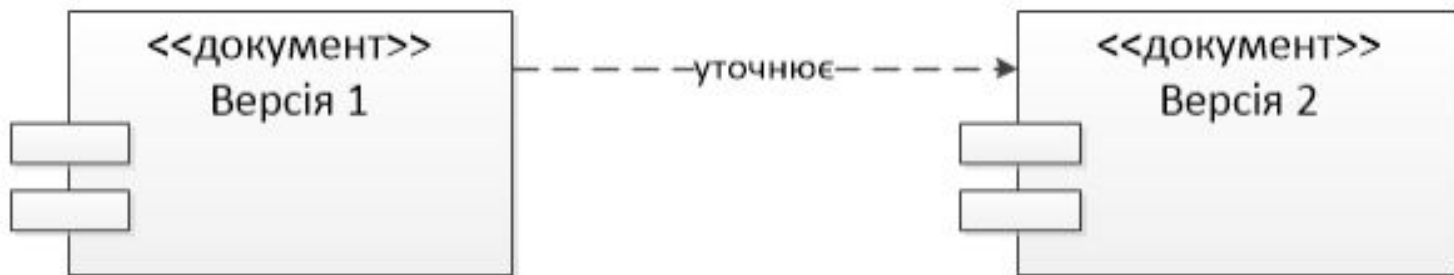
- ▣ **Наслідування** підкласом властивостей суперкласу. Може мати позначку “один до багатьох”. Різновидами наслідування можуть бути відношення узагальнення й спеціалізації.



- ▣ **Альтернативна асоціація.** Деякий клас одночасно може перебувати у зв'язку тільки з одним елементом певної множини класів. Можливі альтернативи позначаються тим, що відповідні їм дуги перетинаються пунктирною лінією з позначкою {or} (або)



- ▣ **Залежність.** Є багато видів залежностей між класами: деякий клас-клієнт може використовувати певний сервіс (операцію) іншого класу; класи можуть бути пов'язані відношенням трасування, коли один трансформується в другий унаслідок певного процесу життєвого циклу, наприклад, клас аналізу перетворюється в клас проекту, а потім у клас реалізації. Один клас може бути уточненням другого.

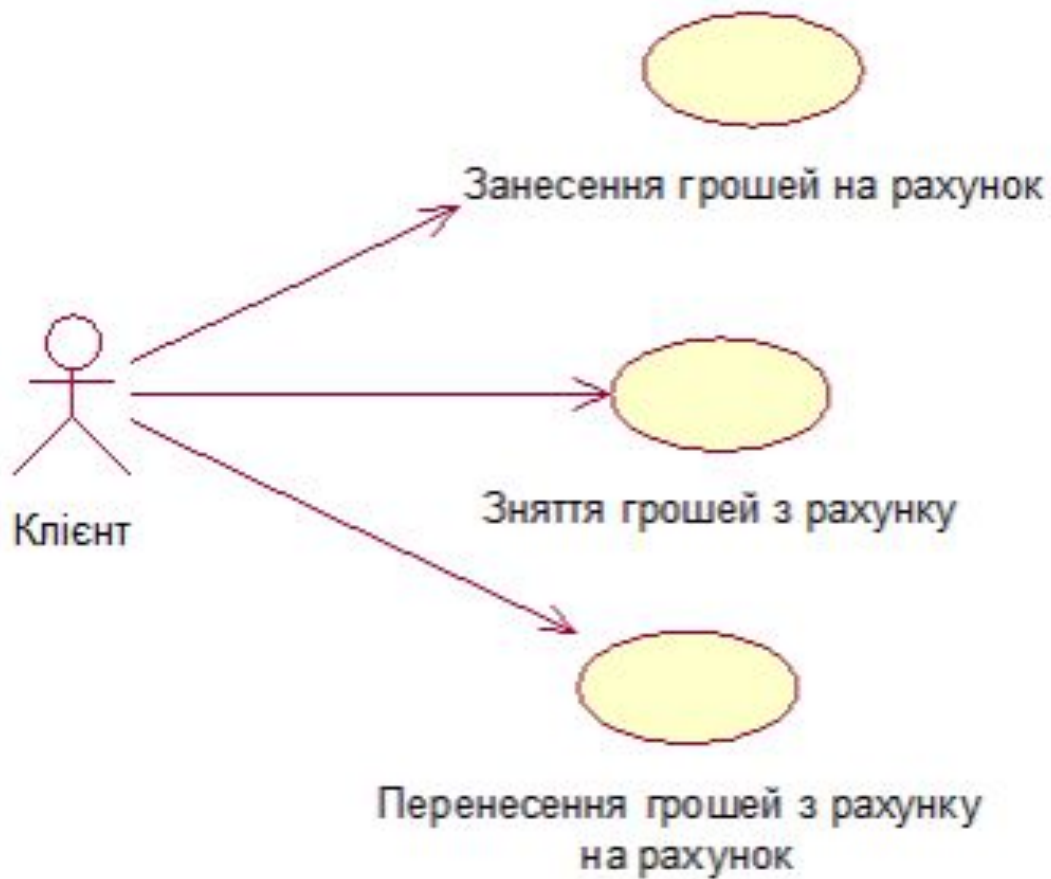


- ▣ **Екземпляризація.** Це залежність між параметризованим абст-рактним класом-шаблоном (template) і реальним класом, який ініційовано шляхом визначення параметрів шаблону. Прикладом параметризованих класів є контейнерні класи для мови програмування C++. На діаграмі класів параметризований клас позначається так: на рамці ікони класу зверху праворуч зображається штрихами маленький прямокутник, всередині якого подаються назви формальних параметрів шаблону, як на рис. 9, де T є параметр, що визначає тип елемента множини. На ім'я класу, який створюється внаслідок визначення параметрів шаблону, посилаються в кутових дужках після імені параметризованого класу-шаблону як префікса.



- Діаграма класів може належати до екземплярів класу, суперкласів (абстрактних класів) або підкласів (конкретних класів). У кожному з конкретних прикладів на іконі класу перед його назвою зазначається його стереотип («підклас», «суперклас» тощо), при цьому за умовчанням вважається клас.
- Для стереотипів, які позначають відношення, фіксованими є такі: «асоціація», «наслідування», «екземпляризація», «узагальнення», «розширення» та інші. Крім того, користувач може вводити стереотипи, властиві специфіці його проблемної галузі, як-от: «успадковує», «контролює», «є наслідком» тощо.

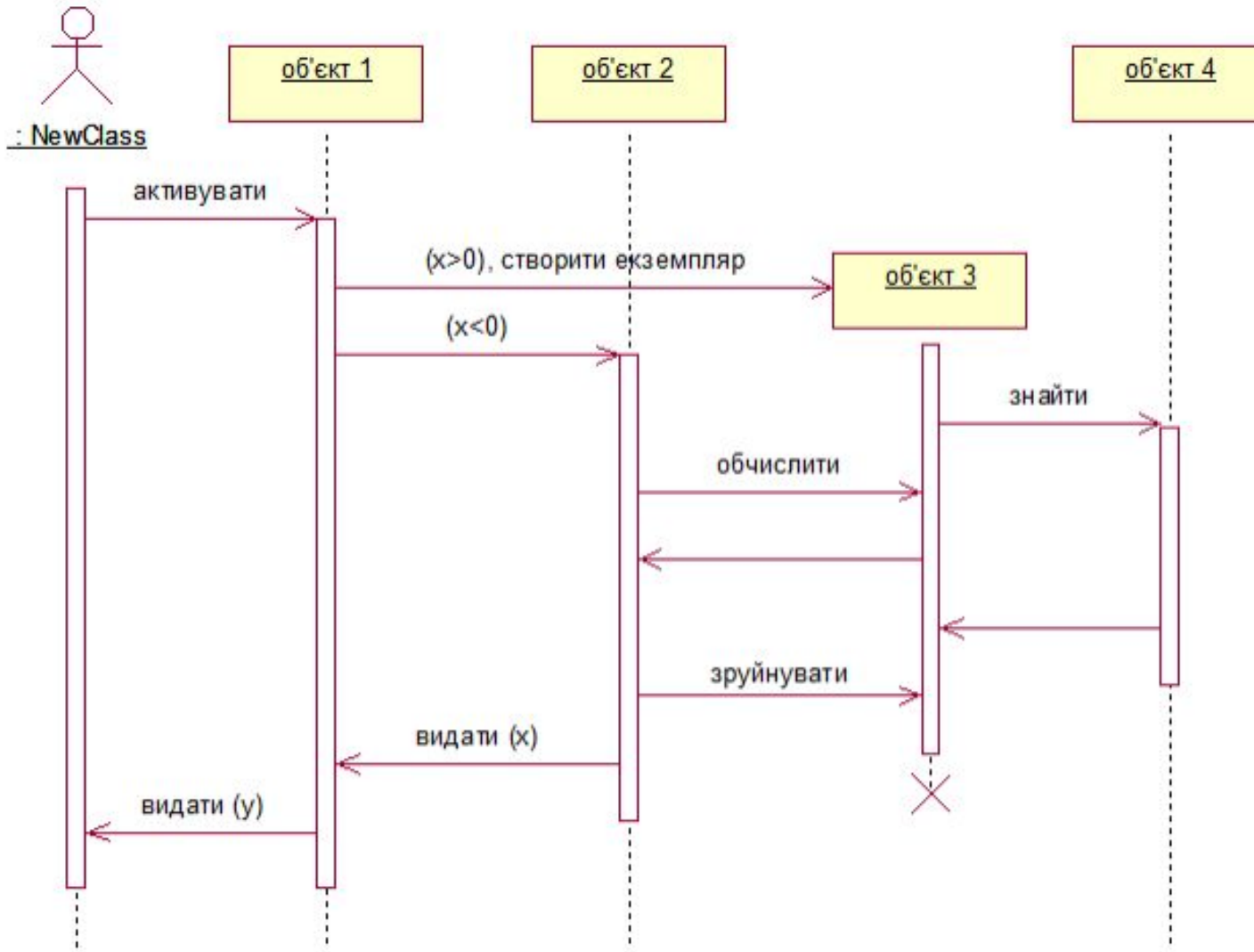




## ДІАГРАМИ ПОСЛІДОВНОСТІ

- Для наочного представлення поведінки об'єктів у сценаріях застосовується спеціальна нотація, так звана діаграма взаємодії.
- Для її побудови кожному з об'єктів сценарію ставиться у відповідність його лінія життя, яка відображає перебіг подій між його створенням та руйнуванням. На діаграмі вона позначається вертикальною пунктирною лінією, на верхівці якої в прямокутнику зображається назва об'єкта.
- Діаграма представляє всі об'єкти, які беруть участь у взаємодії.
- Допускається кілька зовнішніх інтерфейсів для одного сценарію.
- Лінії життя можуть розгалужуватися, що демонструє умовні варіації поведінки об'єкта, або зливатися.

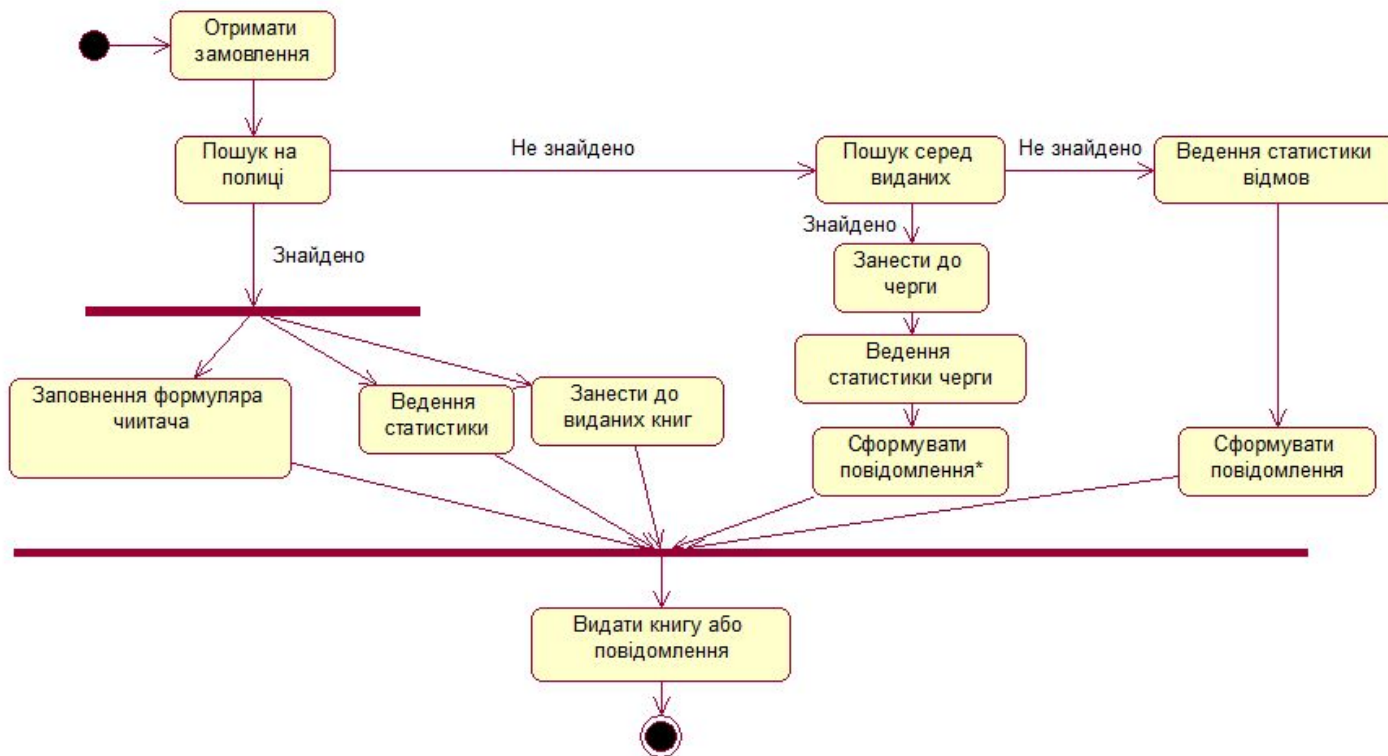
# ДІАГРАМИ ПОСЛІДОВНОСТІ



- Діаграма послідовності відображає потоки керування.
  1. Якщо екземпляр об'єкта існував до старту діаграми, перша стрілка, яка веде до об'єкта, проводиться нижче верхівки його лінії життя.
  2. Якщо об'єкт створюється в певний момент часу, стрілка, яка відповідає повідомленню про його створення, спрямовується до верхівки його лінії життя.
  3. Якщо екземпляр об'єкта руйнується, нижня межа його лінії життя позначається перехрестям як символом руйнування.

- Діаграми співробітництва представляють сукупність об'єктів, поведінка яких значуща для досягнення складових мети системи, та взаємовідношення тих ролей, які об'єкти відіграють у співробітництві.
- На даному виді діаграм моделюється статична взаємодія об'єктів, при цьому фактор часу не враховується і не відображається на діаграмі співробітництва.
- Діаграма співробітництва може бути параметризовано. Тоді вона представляє абстрактну схему співробітництва — так званий патерн, для якого шляхом довизначення параметрів можна створити певну множину конкретних схем співробітництва.

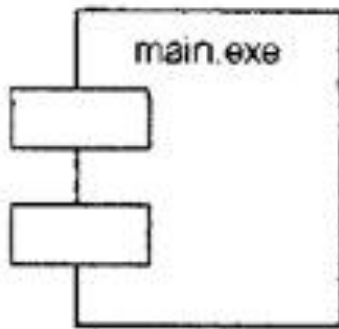
- Модель діяльності в UML представляє поведінку системи як певні роботи, котрі можуть виконувати як система, так і актор, причому послідовність робіт може залежати від прийняття певних рішень залежно від умов, що склалися.



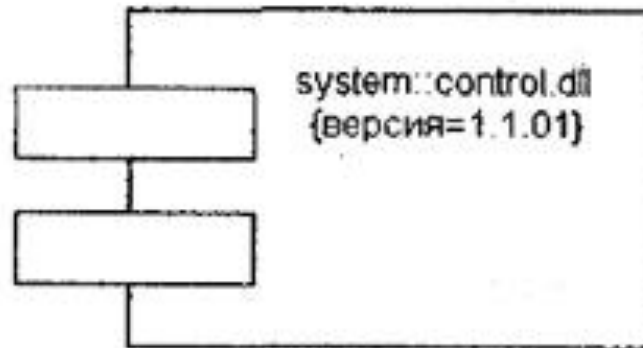
- Як і діаграма переходів у стани , модель станів UML базується на використанні розширеної моделі скінченного автомата.
  
- Нею визначаються:
  1. умови переходів (застороги — guards on transitions)
  2. переходи, зумовлені певними подіями;
  3. дії при переході;
  4. дії при вході в стан;
  5. діяльність, яка триває доти, доки стан є активним;
  6. дії при виході зі стану;
  7. вкладені стани;
  8. паралельно діючі стани

## ДІАГРАМИ КОМПОНЕНТ

- Призначенням діаграми компонент є відображення структури системи як композиції компонент і зв'язків між ними так, як їх уявляє собі програміст.
- Це граф, вузлами якого є компоненти, а дуги відображають відношення залежності.



(a)

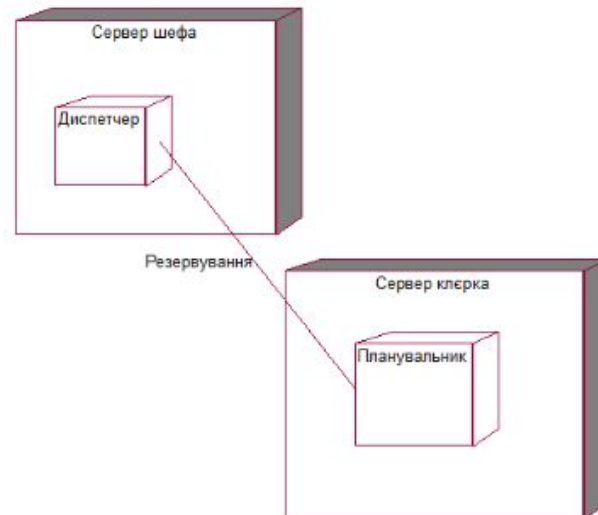


(б)

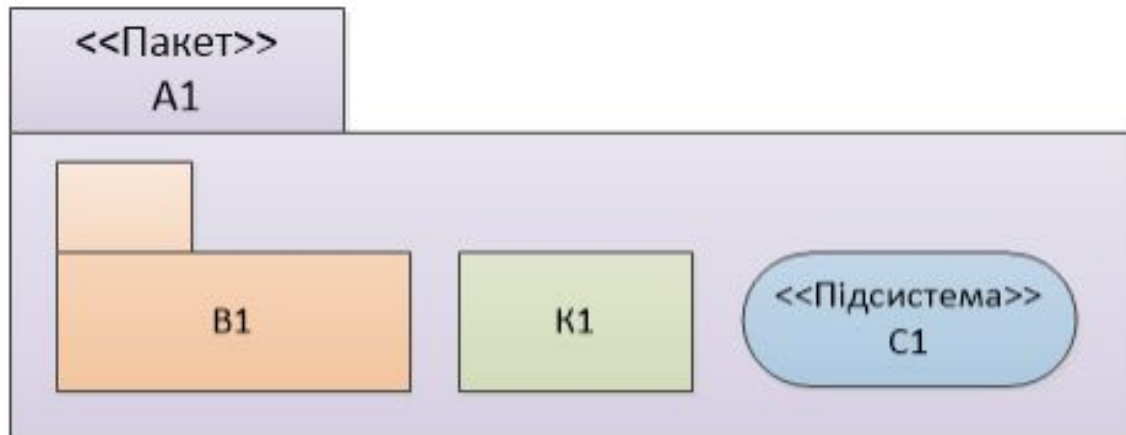


## ДІАГРАМИ РОЗМІЩЕННЯ

- Призначенням даної діаграми є визначення складу фізичних ресурсів системи (що позначаються як вузли системи) та відношень між ними.
- Системи реального часу в багатьох випадках базуються на різноманітних платформах замовника. Інженер повинен розробити не лише програмну частину, а й визначити необхідні апаратні пристрої. *Ці пристрої мають органічно взаємодіяти з програмними компонентами.*



- ▣ **Призначення пакета** — бути елементом конфігурації, тобто елементом, який можна включати як визначену складову композиції в побудові певної системи. На пакет можна посилатися у різних діаграмах, котрі можуть розробляти окремі команди спеціалістів.



**ДЯКУЮ ЗА УВАГУ!**