

# Об'єктно-орієнтоване проектування з використанням



# UML 2.0

## Діаграми варіантів використання

# Зміст

---

- ▣ Про аналіз вимог до програмного продукту
- ▣ Прецеденти – інструмент аналізу функціональних вимог
- ▣ Діаграма варіантів використання як етап візуального моделювання
- ▣ Цілі діаграми варіантів використання
- ▣ Суть діаграми варіантів використання
- ▣ Виконавець (актор)
- ▣ Ідентифікація виконавців
- ▣ Прецедент (варіант використання)
- ▣ Виявлення прецедентів
- ▣ Відношення на діаграмі варіантів використання
- ▣ Типові помилки при моделювання прецедентів
- ▣ Приклад побудови діаграми варіантів використання
- ▣ Практичні завдання для самостійно роботи
- ▣ Теоретичні завдання для самостійно роботи
- ▣ Рекомендовані інформаційні джерела

# Про аналіз вимог до програмного продукту

---



- ▣ **Для чого необхідний аналіз вимог до програмного продукту?**
  - ▣ Наприклад, екстремальне програмування не особливо довіряє етапу аналізу, аргументуючи це великим ризиком потрапити в пастку “аналітичного паралічу”
    - ▣ Зводити до мінімуму або й взагалі відмовлятися від фази аналізу можна в тому випадку, коли розробка буде проводитися з використанням добре знайомої технології на ґрунтовно вивченій предметній області
  - ▣ Часто проекти потерпають невдачу через те, що вимоги не були до кінця проаналізовані
    - ▣ Тому перед власне розробкою необхідно впевнитися, що вимоги вивчені всесторонньо і їх правильно зрозуміли

## Прецеденти – інструмент аналізу функціональних вимог

---

- За допомогою прецедентів (варіантів використання) можна створити точну модель того, що повинна вміти робити система
- На базі прецедентів можна досліджувати інші аспекти розробки
- Прецеденти заповнюють розрив між кінцевими користувачами і вимогами до системи
- Прецеденти дозволяють визначити оптимальний шлях від функціональних вимог до реалізації
- Прецеденти використовуються не лише для аналізу вимог, а й створюють основу для наступних етапів аж до тестування

# Основні концепції моделювання прецедентів

---

## □ Основні складові моделі прецедентів:

- Виконавець (актор, actor)
- Прецедент (варіант використання, use case)

## □ Виконавець

- Представляє собою сутність, що знаходиться за межами системи, як правило, ним виступає користувач системи
- Виконавці взаємодіють із системою, що призводить до виконання деяких подій
- Кожна окрема роль представляється окремим виконавцем

## □ Прецедент

- Прецедент представляє собою послідовність кроків, що виконуються системою за дорученням актора
- Прецеденти дають виконавцю деякий важливий результат
- Прецедент складається із основних послідовностей подій і може включати додаткові послідовності подій

# Діаграма варіантів використання як етап візуального моделювання

---

- Візуальне моделювання в UML можна представити як процес послідовного переходу:
  1. Найбільш загальна і абстрактна концептуальна модель
  2. Логічна модель
  3. Фізична модель
  
- Початкова модель програмної системи будується у формі діаграми варіантів використання (use case diagram), яка описує функціональне призначення системи
  
- Діаграма варіантів використання є вихідним концептуальним представленням або концептуальною моделлю системи в процесі її проектування та розробки

# Цілі діаграми варіантів використання

---

- Визначити загальні гранці і контекст предметної області на початкових етапах проектування системи
- Сформулювати загальні вимоги до функціональної поведінки системи
- Розробити вихідну концептуальну модель системи для її наступної деталізації у формі логічних і фізичних деталей
- Підготувати вихідну документацію для взаємодії розробників системи з її замовниками та користувачами

# Суть діаграми варіантів використання

---

- Система представляється у вигляді множини виконавців (акторів), що взаємодіють із системою через варіанти використання (прецеденти)
- Актором називається будь-яка сутність, що взаємодіє з системою із зовні (користувача, інша система і т.п.)
- Варіант використання служить для опису сервісів, які система надає актору
- Варіант використання нічого не каже про те, яким чином буде реалізована взаємодія актора із системою



# Виконавець (актор)

---

- Актор представляє деяку зовнішню відносно системи сутність, яка взаємодії з системою і використовує її функціональні можливості для досягнення певних цілей
- Внутрішня структура актора ніяк не визначається; для актора важливо лише те, як він сприймається із сторони системи
- Актори взаємодіють із системою шляхом передачі та прийому повідомлень від варіантів використання
- Ця взаємодія може бути виражена через асоціації між окремими акторами і варіантами використання

# Ідентифікація виконавців

---

- Ключові запитання для ідентифікації виконавців:
  - Хто буде використовувати функціональні можливості системи?
  - Хто представляє або отримує інформацію?
  - Хто може змінювати інформацію?
  - Чи є якісь інші системи, які взаємодіють із системою, що розробляється?
  
- Виконавців, як правило, значно легше ідентифікувати, як прецеденти
  
- Типові помилки при ідентифікації виконавців
  - Призначення багатьох виконавців для однієї із тієї ж ролі
  - Виконавці можуть бути неявними, тобто не ідентифікуватися, як користувачі системи

# Прецедент (варіант використання)

---

- Прецедент, або варіант використання, використовується для специфікації загальних особливостей поведінки системи чи будь-якої іншої сутності предметної області без розгляду внутрішньої структури
- Кожний варіант використання визначає послідовність дій, які повинні бути виконані системою при взаємодії з відповідним актором
- Варіант використання позначається на діаграмі еліпсом, всередині якого знаходиться його скорочена назва

# Виявлення прецедентів

---

- Прецеденти завжди виражаються з точки зору виконавця (користувача системи)
- Спосіб виявлення прецедентів
  - Взяти кожного ідентифікованого виконавця і спробувати визначити поведінку або інформацію, яку даний виконавець вимагає від системи
- Головна задача при виявленні прецедентів
  - Намагатися уникнути надмірної деталізації, що веде до швидкого росту кількості прецедентів

## Відношення на діаграмі варіантів використання

---

- Стандартні види відношень між акторами та варіантами використання:
  1. Відношення узагальнення (generalization relationship);
  2. Відношення асоціації (association relationship);
  3. Відношення розширення (extend relationship);
  4. Відношення включення (include relationship).

# Відношення узагальнення між акторами

---

- Два і більше актори можуть мати однакові властивості, тобто взаємодіяти з однаковою множиною прецедентів схожим чином
- Така спільність властивостей та поведінки представляється у вигляді *відношення узагальнення* з іншими, можливо, абстрактним актором, який моделює відповідну спільність ролей

# Відношення асоціації

---

- ▣ *Відношення асоціації* в контексті діаграм варіантів використання служить для позначення ролі актора
- ▣ За допомогою асоціації відображається семантична взаємодія акторів та варіантів використання
- ▣ Відношення асоціації відображається суцільною лінією, що сполучає актора та варіант використання
- ▣ Ця лінія може містити такі позначення як ім'я та кратність

# Кратність відношення асоціації

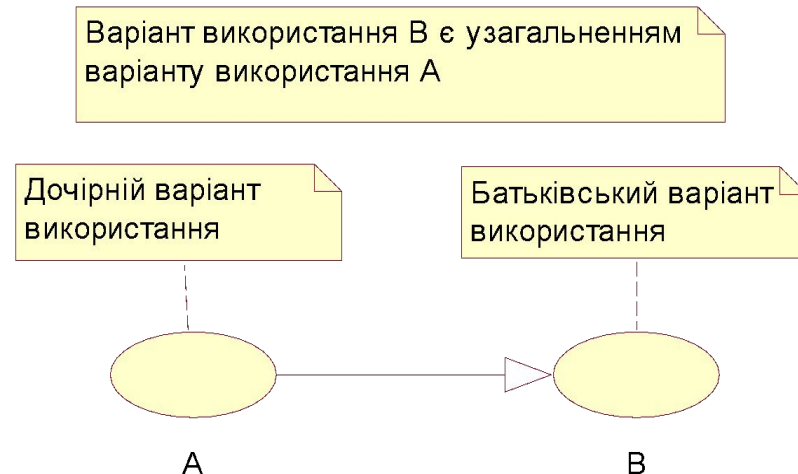
---

- Кратність характеризує загальну кількість екземплярів сутності, що можуть виступати в якості елементів даної асоціації
- Найбільш поширеними є наступні форми запису кратності:
  1. Ціле невід'ємне число;
  2. Два цілих невід'ємних числа, що розділені двома крапками;
  3. Два символи, розділений двома крапками, першим з яких є ціле невід'ємне число, а другим символ «\*»;
  4. Єдиний символ «\*».
- За замовчуванням значення кратності асоціації вважається рівним 1.



# Відношення узагальнення

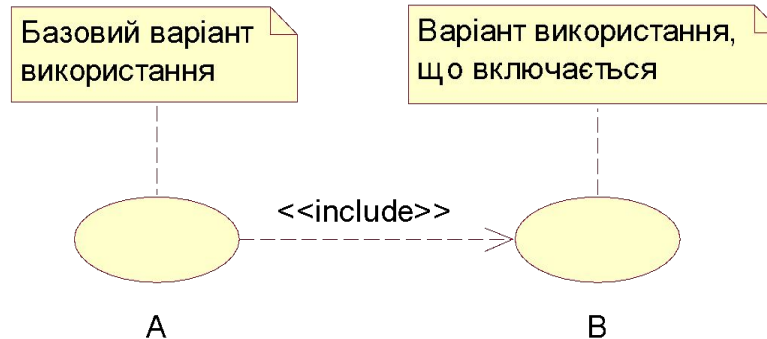
- *Відношення узагальнення (наслідування)* служить для вказівки на той факт, що деякий варіант використання **A** може бути узагальнений до варіанту використання **B**
- Варіант використання **A** є спеціалізацією **B**
- **B** називають батьківським відносно **A**
- **A** називається потомком **B**
- Слід зауважити, що потомок наслідує всі властивості та поведінку свого предка, а також може доповнювати їх новими властивостями та особливостями поведінки



# Відношення включення

- *Відношення включення* між двома варіантами використання вказує, що деяка поведінка для одного варіанту використання включається в якості складової в послідовність поведінки іншого варіанту використання
- Коли екземпляр першого варіанту використання в процесі свого виконання досягає точки включення в послідовність поведінки другого варіанту використання, екземпляр першого варіанту використання виконує послідовність дій, що визначає поведінку другого варіанту використання, після чого продовжує виконання дій своє поведінки.

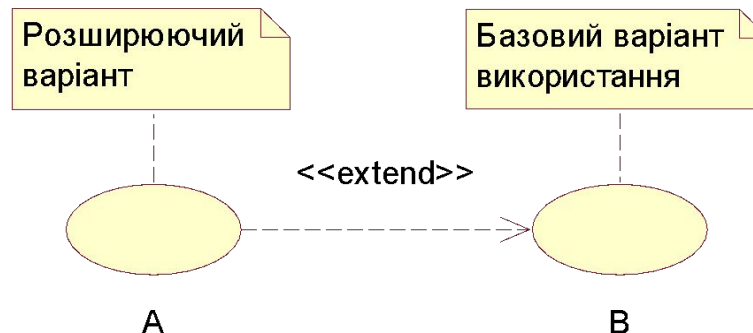
Поведінка варіанту використання А включає поведінку варіанту використання В



# Відношення розширення

- Якщо має місце *відношення розширення* від варіанту використання **A** до варіанту використання **B**, то це значить, що властивості екземпляру варіанту використання **B** можуть бути доповнені завдяки властивостям варіанту **A**.
- Відношення розширення означає, що деякий варіант використання може приєднати до своєї поведінки деяке додаткову поведінку, що визначена для іншого варіанту використання.

Поведінка варіанту використання B може бути доповнена поведінкою варіанту використання A



# Типові помилки при моделюванні прецедентів

---

- Створення надто обтяжених прецедентів
- Створення надто дрібних прецедентів
- Визначення прецедентів з точки зору системи
- Надмірне використання відношень розширення замість включення
- Надмірне вживання узагальнення для виконавців та прецедентів, оскільки узагальнення завжди можна провести на наступній ітерації, коли система більш детально вивчена

## Приклад побудови діаграми варіантів використання: постановка задачі

---

- Провести моделювання прецедентів для системи продажу товару за каталогом

# Приклад побудови діаграми варіантів використання: визначення акторів та базових прецедентів

- Визначення акторів:

- Продавець
- Клієнт

- Центральним варіантом використання буде *Оформити замовлення на купівлю товару*



- Значення вказаних на даній діаграмі кратностей відображає загальні правила оформлення замовлень на купівлю товару:

- Один продавець може брати участь в оформленні кількох замовлень
- Кожне замовлення може бути оформлене лише одним продавцем
- Кожен покупець може оформити на себе кілька замовлень
- Одне замовлення може бути оформлене лише на одного покупця

## Приклад побудови діаграми варіантів використання: уточнення

- На наступному етапі розробки даної діаграми варіант використання Оформити замовлення на покупку товару може бути уточнений



# Практичні завдання для самостійної роботи (1)

---

## ▣ 1. Провести моделювання прецедентів та для наступної програмної системи

### ▣ Система бронювання квитків для авіакомпанії

- ▣ На ринок вийшла нова авіакомпанія «GlobalAvia».
- ▣ Менеджери компанії вирішили замовити у вашої фірми розробку системи бронювання квитків. При замовленні фірма поставила ряд умов, які обов'язково повинні бути виконані.
- ▣ В першій версії системи вони хочуть бачити дві частини. Робота першої частини системи пов'язана із занесенням інформації. Друга частина системи призначена для спілкування з клієнтами.
- ▣ При формулювання вимог менеджери звернули увагу, що рейси сплановано так, що до пункту призначення можна долетіти з пересадками. Одна із вимог полягала в тому, щоб система допомагала підбирати оптимальні маршрути залежно від побажань клієнта авіакомпанії.



## Практичні завдання для самостійної роботи (2)

---

### ▣ 2. Провести моделювання прецедентів та для наступної програмної системи

#### ▣ Система для *автоматизації обліку замовлень на фото- та відеозйомку*

#### ▣ З точки зору замовника, до програми ставляться такі вимоги (С-вимоги):

- ▣ 1. Календар із замовленнями (доступ до замовлень через календар), який повинен містити максимум інформації про замовлення на визначену дату.
- ▣ 2. Друк замовлення клієнта (для підписування замовником та клієнтом).
- ▣ 3. Можливості резервного копіювання та відновлення бази даних системи.
- ▣ 4. Можливості очистки всієї бази даних системи.
- ▣ 5. Пароль на програмну оболонку.
- ▣ 6. Заставка та інформація про програму (із атрибутами фірми).
- ▣ 7. Можливості кількох замовлень на одну дату (інші замовлення виконуватиме співробітник за відповідну оплату організації).
- ▣ 8. Можливості змінюваного переліку послуг.
- ▣ 9. Наявність коментарів у анкеті замовлення.
- ▣ 10. Можливість відображення в замовленні найманих працівників із відповідною

# Теоретичні завдання для самостійної роботи

---

- Інтерфейси на діаграмі варіантів використання

# Рекомендовані інформаційні джерела

---

1. Амриш, Кэри и др. Разработка корпоративных Java-приложений с использованием J2EE и UML.: Пер. с англ. – М.: Издательский дом "Вильямс", 2002. – 272 с.: ил.
2. Нейбург, Эрик, Дж., Максимчук, Роберт, А. Проектирование баз данных з помощью UML.: Пер. с англ. – М.: Издательский дом «Вильямс», 2002. – 288 с.: ил.
3. Орлов С. А. Технологии разработки программного обеспечения: Учебник для вузов. 3-е изд. – СПб.: Питер, 2004. – 527 с.: ил.
4. Фаулер М., Скот К. UML. Основы. – Пер. с англ. – СПб: Символ-Плюс, 2002. – 192 с.
5. Шмуллер, Джозеф. Освой самостоятельно UML за 24 часа, 3-е издание.: Пер. с англ. – М.: Издательский дом «Вильямс», 2005. – 416 с.