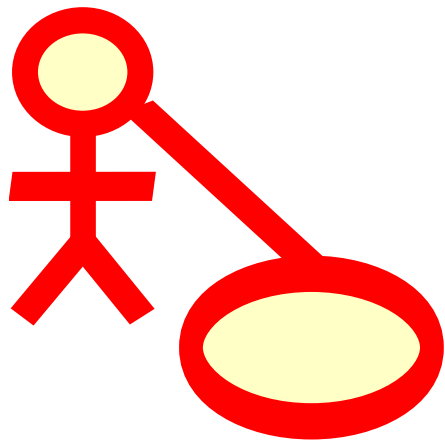


# Підручник з Umbrello UML Modeller





# Зміст

<b>1</b>	<b>Вступ</b>	<b>7</b>
<b>2</b>	<b>Основи UML</b>	<b>8</b>
2.1	Про UML . . . . .	8
2.2	Елементи UML . . . . .	9
2.2.1	Діаграма випадків використання . . . . .	9
2.2.1.1	Випадок використання . . . . .	9
2.2.1.2	Актор . . . . .	10
2.2.1.3	Опис випадків використання . . . . .	10
2.2.2	Діаграма класів . . . . .	10
2.2.2.1	Клас . . . . .	11
2.2.2.1.1	Атрибути . . . . .	11
2.2.2.1.2	Операції . . . . .	11
2.2.2.1.3	Шаблони . . . . .	12
2.2.2.2	Асоціації класів . . . . .	12
2.2.2.2.1	Узагальнення . . . . .	12
2.2.2.2.2	Асоціації . . . . .	12
2.2.2.2.3	Агрегація . . . . .	13
2.2.2.2.4	Композиція . . . . .	13
2.2.2.3	Інші елементи діаграми класів . . . . .	13
2.2.2.3.1	Інтерфейси . . . . .	13
2.2.2.3.2	Типи даних . . . . .	13
2.2.2.3.3	Переліки . . . . .	13
2.2.2.3.4	Пакунки . . . . .	14
2.2.3	Діаграми послідовностей . . . . .	14
2.2.4	Діаграми співпраці . . . . .	14
2.2.5	Діаграма станів . . . . .	15
2.2.5.1	Стан . . . . .	16
2.2.6	Діаграма діяльності . . . . .	16
2.2.6.1	Діяльність . . . . .	17
2.2.7	Допоміжні елементи . . . . .	17
2.2.8	Діаграми компонентів . . . . .	18
2.2.9	Діаграми впровадження . . . . .	18

2.2.10	Діаграми взаємозв'язків сутностей . . . . .	18
2.2.10.1	Сутність . . . . .	18
2.2.10.1.1	Атрибути сутності . . . . .	19
2.2.10.1.2	Обмеження . . . . .	19
2.2.11	Концепції розширеної діаграми взаємозв'язку сутностей . . . . .	19
2.2.11.1	Спеціалізація . . . . .	19
2.2.11.1.1	Несумісна спеціалізація . . . . .	20
2.2.11.1.2	Перекрита спеціалізація . . . . .	20
2.2.11.1.3	Категорія . . . . .	21
<b>3</b>	<b>Як працювати з Umbrello UML Modeller</b>	<b>22</b>
3.1	Інтерфейс користувача . . . . .	22
3.1.1	Перегляд дерева . . . . .	23
3.1.2	Вікно документації та журналу команд . . . . .	23
3.1.3	Робоча область . . . . .	23
3.2	Створення, завантаження і збереження моделей . . . . .	24
3.2.1	Нова модель . . . . .	24
3.2.2	Збереження моделі . . . . .	24
3.2.3	Завантажити модель . . . . .	24
3.3	Редагування моделей . . . . .	24
3.4	Додавання і вилучення діаграм . . . . .	25
3.4.1	Створення діаграм . . . . .	25
3.4.2	Вилучення діаграм . . . . .	25
3.4.3	Перейменування діаграм . . . . .	25
3.5	Редагування діаграм . . . . .	25
3.5.1	Як вставити елементи . . . . .	26
3.5.2	Вилучення елементів . . . . .	26
3.5.3	Редагування елементів . . . . .	26
3.5.4	Редагування класів . . . . .	27
3.5.4.1	Загальні параметри класу . . . . .	27
3.5.4.2	Параметри атрибутів класу . . . . .	27
3.5.4.3	Параметри операцій класу . . . . .	27
3.5.4.4	Параметри шаблонів класу . . . . .	27
3.5.4.5	Сторінка асоціацій класів . . . . .	27
3.5.4.6	Сторінка показу класу . . . . .	27
3.5.4.7	Сторінка стилю класу . . . . .	28
3.5.5	Асоціації . . . . .	28
3.5.5.1	Точки якорів . . . . .	28
3.5.6	Нотатки, текст і блоки . . . . .	28
3.5.6.1	Якорі . . . . .	29

<b>4</b>	<b>Імпортування та створення коду</b>	<b>30</b>
4.1	Створення коду . . . . .	30
4.1.1	Як створити код . . . . .	30
4.1.1.1	Параметри створення . . . . .	31
4.1.1.1.1	Докладність коментарів . . . . .	31
4.1.1.1.2	Теки . . . . .	31
4.1.1.1.3	Правила перезапису . . . . .	32
4.1.1.1.4	Мова . . . . .	32
4.1.1.2	Створення за допомогою майстра створення . . . . .	32
4.2	Імпортування коду . . . . .	32
<b>5</b>	<b>Інші можливості</b>	<b>34</b>
5.1	Інші можливості Umbrello UML Modeller . . . . .	34
5.1.1	Копіювання об'єктів у вигляді зображень PNG . . . . .	34
5.1.2	Експортування до зображення . . . . .	34
5.1.3	Друк . . . . .	34
5.1.4	Логічні теки . . . . .	34
<b>6</b>	<b>Автори та історія програми</b>	<b>36</b>
<b>7</b>	<b>Авторські права</b>	<b>37</b>

### **Анотація**

Umbrello UML Modeller допомагає у процесі розробки програмного забезпечення. Програма використовує загальноприйнятий стандарт уніфікованої мови моделювання (Unified Modelling Language або UML), за допомогою якого можна створювати діаграми для розробки і документування ваших систем.

# Розділ 1

## Вступ

Umbrello UML Modeller — це програма для роботи з діаграмами UML, яка допоможе вам у розробці програмного забезпечення, особливо під час стадій аналізу і розробки компонування проекту. Umbrello UML Modeller допоможе вам створити високоякісний продукт. Крім того, UML можна використовувати для створення документації до компонування вашого програмного забезпечення, ця документація допоможе вам і вашим партнерам-розробникам.

Створення доброї моделі вашого програмного забезпечення є найкращим шляхом для обміну інформацією з іншими розробниками, які працюють над проектом, та користувачами проекту. Добра модель особливо важлива для проектів середнього та великого розмірів, але вона не буде зайвою і для невеличких проектів. Навіть якщо ви працюєте над маленьким проектом, у якого лише один розробник, добра модель буде корисною, оскільки вона дасть вам змогу бачити загальну картину і писати код правильно з першого разу.

UML — це мова створення діаграм, яка використовується для опису таких моделей. Ви можете відтворити свої ідеї у UML за допомогою діаграм різних типів. У Umbrello UML Modeller 2.11 передбачено підтримку таких діаграм:

- Діаграма класу
- Діаграма послідовностей
- Діаграма співпраці
- Діаграма випадків використання
- Діаграма станів
- Діаграма діяльності
- Діаграма компонентів
- Діаграма впровадження
- Діаграма взаємозв'язків сутностей

Докладніші відомості щодо UML можна знайти на вебсайті OMG, <http://www.omg.org>, компанії, яка створила стандарт UML.

Ми сподіваємося, що вам сподобається Umbrello UML Modeller, і що ця програма допоможе вам створити високоякісне програмне забезпечення. Програма Umbrello UML Modeller є вільним програмним забезпеченням, отже доступна безкоштовно, розробники програми Umbrello UML Modeller лише просять вас повідомляти про будь-які вади, проблеми або пропозиції за адресою електронної пошти [umbrello-devel@kde.org](mailto:umbrello-devel@kde.org) або за допомогою мережі, <http://bugs.kde.org>.

## Розділ 2

# Основи UML

### 2.1 Про UML

Цю главу присвячено короткому огляду основ UML. Пам'ятайте, що це далеко не повний підручник з UML, а скоріше короткий вступ до підручника з UML. Якщо ви бажаєте дізнатися більше про універсальну мову моделювання (Unified Modelling Language або UML) або отримати загальні відомості щодо аналізу і розробки програмного забезпечення, зверніться до однієї з багатьох книжок, присвячених цим темам. Крім того, у мережі Інтернет ви знайдете багато навчальних посібників, якими можна скористатися для вивчення основних відомостей.

Універсальна мова моделювання (Unified Modelling Language або UML) — це мова позначень або побудови діаграм, призначена для визначення, візуалізації і документування моделей зорієнтованих на об'єкти систем програмного забезпечення. UML не є методом розробки, іншими словами, у конструкціях цієї мови не повідомляється про те, що робити першим, а що останнім, і не надається інструкцій щодо побудови вашої системи, але ця мова допомагає вам наочно переглядати компонування системи і полегшує співпрацю з іншими її розробниками. Розробкою UML керує Object Management Group (OMG). Ця мова є загальноприйнятим стандартом графічного опису програмного забезпечення.

UML розроблено для розробки структури зорієнтованого на об'єкти програмного забезпечення, ця мова має дуже обмежену користь для програмування на основі інших парадигм.

Конструкції UML створюються з багатьох модельних елементів, які позначають різні частини системи програмного забезпечення. Елементи UML використовуються для побудови діаграм, які відповідають певній частині системи або точці зору на систему. У Umbrello UML Modeller реалізовано підтримку таких типів діаграм:

- *Діаграма випадків використання* показує дієвих осіб (людей або інших користувачів системи), випадки використання (сценарії використання системи) та їх взаємодію
- *Діаграми класів*, на яких буде показано класи та зв'язки між ними
- *Діаграми послідовності*, на яких показано об'єкти і послідовність методів, якими ці об'єкти викликають інші об'єкти.
- *Діаграми співпраці*, на яких буде показано об'єкти та їх взаємозв'язок з наголосом на об'єкти, які беруть участь у обміні повідомленнями
- *Діаграми стану*, на яких буде показано стани, зміну станів і події у об'єкті або частині системи
- *Діаграми діяльності*, на яких буде показано дії та зміни однієї дії іншою, які є наслідком подій, що сталися у певній частині системи
- *Діаграми компонентів*, на яких буде показано програмні компоненти високого рівня (на зразок KParts або Java Beans).



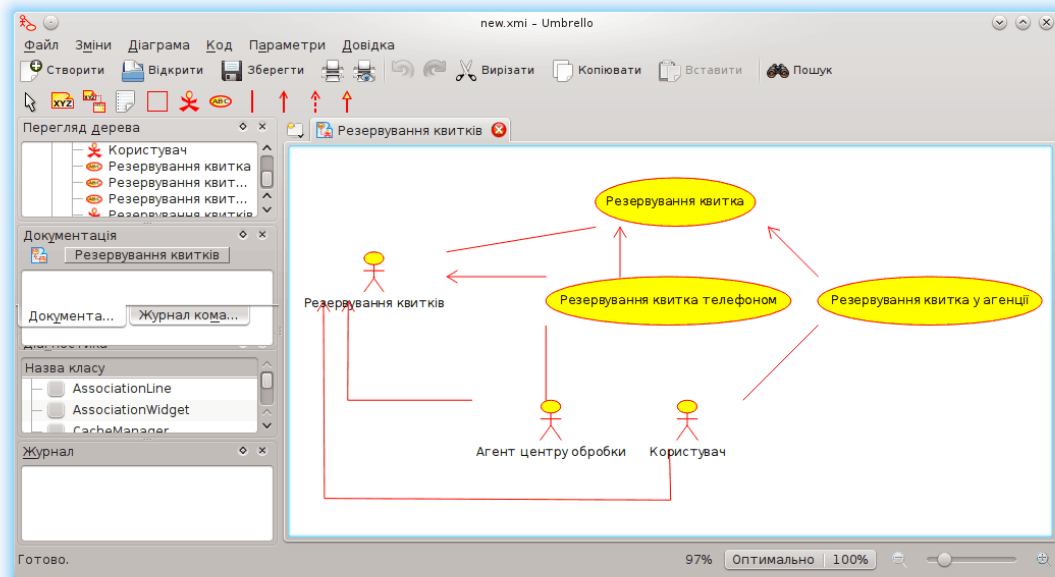
- *Діаграми впровадження*, на яких буде показано екземпляри компонентів та їх взаємодію.
- *Діаграми взаємозв'язку сутностей*, на яких буде показано дані, взаємозв'язки і умови обмеження зв'язків між даними.

## 2.2 Елементи UML

### 2.2.1 Діаграма випадків використання

Діаграми випадків використання описують взаємозв'язки і залежності між групою *випадків використання* і акторами, що беруть участь у процесі.

Важливо зауважити, що діаграми випадків використання не призначено для показу компонування, вони не можуть описати внутрішню структуру системи. Діаграми випадків використання призначено для полегшення обміну інформацією між майбутніми користувачами системи і замовником, вони особливо корисні для визначення переліку можливостей, які повинна мати система. За діаграмами випадків використання можна сказати, *що* система має робити, але не те, *як* вона досягає потрібних результатів, для останнього ці діаграми просто не придатні.



Показ у Umbrello UML Modeller діаграми випадків використання

#### 2.2.1.1 Випадок використання

*Випадок використання* визначає, з точки зору акторів (користувачів), групу дій у системі, які призводять до конкретного видимого результату.

Випадки використання є описом типових елементів взаємодії користувачів системи з самою системою. Вони відповідають зовнішньому інтерфейсу системи і визначають форму вимог до того, що має робити система (зауважте, лише «що», а не «як»).

Під час роботи з випадками використання важливо пам'ятати декілька простих правил:

- Кожен випадок використання має бути пов'язано принаймні з одним актором
- У кожного з випадків використання має бути ініціатор (тобто актор)

- Кожен з випадків використання має призводити до відповідного результату (результату з «комерційним значенням»)

Випадки використання можуть мати зв'язки з іншими випадками використання. Ось три найпоширеніших зв'язки між випадками використання:

- «*включення*», яке вказує на те, що випадок використання відбувається *всередині* іншого випадку використання
- «*розширення*», яке означає, що у певних випадках або у певній точці (яку називають точкою розширення) випадок використання буде розширено іншим випадком використання.
- *Узагальнення*, за якого випадок використання успадковує характеристики випадку використання «вищого рангу», при цьому можливе перевизначення деяких з характеристик у спосіб, подібний до успадкування між класами.

### 2.2.1.2 Актор

Актор — це зовнішній чинник (поза межами системи), який взаємодіє з системою шляхом участі (і часто ініціювання) у випадку використання. Акторами, на практиці, можуть бути звичайні люди (наприклад, користувачі системи), інші комп'ютерні системи або зовнішні події.

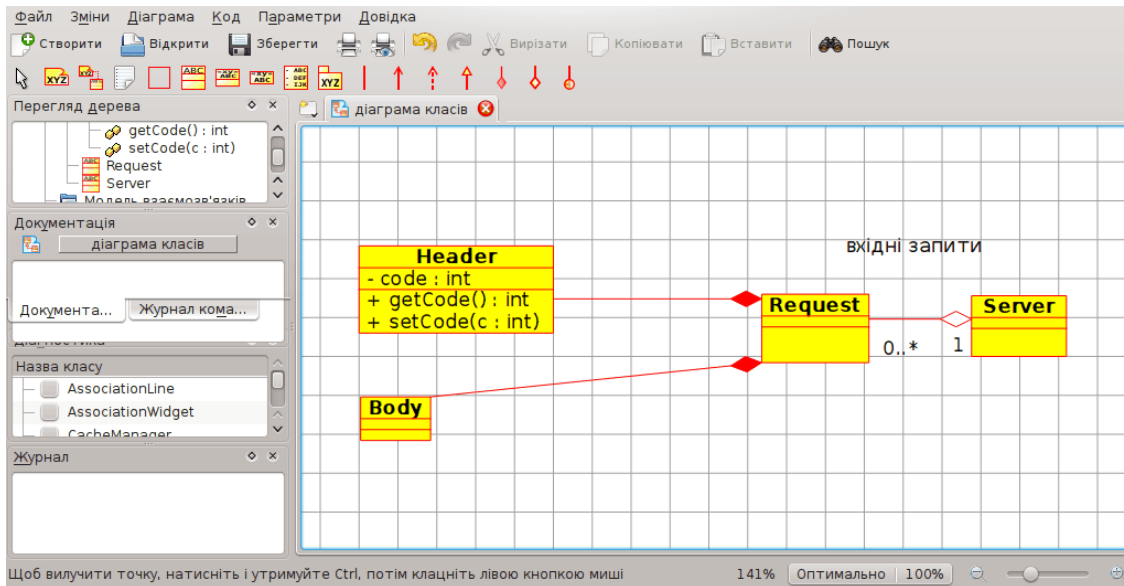
Акторам відповідають не *реальні* люди або системи, а лише їх *ролі*. Це означає, що коли особа у різний спосіб взаємодіє з системою (виконуючи різні ролі), їй відповідають декілька акторів. Наприклад, особа, яка виконує підтримку користувачів телефоном і приймає замовлення від користувачів до системи, може бути показано актором «Персонал служби підтримки» і актором «Відповідальний за продажі».

### 2.2.1.3 Опис випадків використання

Описи випадків використання — це текстові примітки до випадків використання. Зазвичай, вони мають форму нотаток або документа, який певним чином пов'язано з випадком використання, і який пояснює процеси або дії, які відбуваються під час випадку використання.

## 2.2.2 Діаграма класів

На діаграмах класів буде показано різноманітні класи, які утворюють систему і їх взаємозв'язки. Діаграми класів називають «статичними діаграмами», оскільки на них показано класи разом з методами і атрибутами, а також статичний взаємозв'язок між ними: те, яким класам «відомо» про існування яких класів, і те, які класи «є частиною» інших класів, — але не показано методи, які при цьому викликаються.

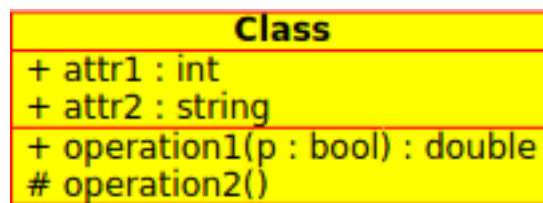


Показ Umbrello UML Modeller діаграми класів

### 2.2.2.1 Клас

Клас визначає атрибути і методи набору об'єктів. Всі об'єкти цього класу (екземпляри цього класу) мають спільну поведінку і однаковий набір атрибутів (кожен з об'єктів має свій власний набір значень). Іноді замість назви «клас» використовують назву «тип», але, слід зауважити, що ці назви описують різні речі: тип є загальнішим визначенням.

У UML класи позначаються прямокутниками з назвою класу, у цих прямокутниках у вигляді двох «відсіків» може бути показано атрибути і операції класу.



Наочне представлення класу у UML

#### 2.2.2.1.1 Атрибути

У UML атрибути показуються щонайменше назвою, також може бути показано їх тип, початкове значення і інші властивості. Крім того, атрибути може бути показано з областю видимості атрибута:

- + відповідає публічним (*public*) атрибутам
- # відповідає захищеним (*protected*) атрибутам
- - відповідає приватним (*private*) атрибутам

#### 2.2.2.1.2 Операції

Операції (методи) також показуються принаймні назвою, крім того, може бути показано їх параметри і типи значень, які буде повернуто. Операції, як і атрибути, може бути показано з областю видимості:

- + відповідає *публічним* (*public*) операціям
- # відповідає *захисеним* (*protected*) операціям
- - відповідає *приватним* (*private*) операціям

### 2.2.2.1.3 Шаблони

Серед класів можуть бути шаблони, значення, які використовуються для невизначеного класу або типу. Тип шаблону визначається під час ініціалізації класу (тобто, під час створення об'єкта). Шаблони існують у сучасній мові програмування C++, їх буде введено у Java 1.5, де вони матимуть назву *Generic*.

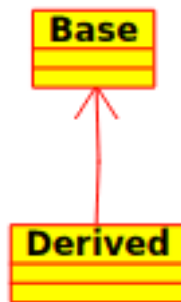
### 2.2.2.2 Асоціації класів

Класи можна співвіднести (пов'язати) один з одним у декілька способів:

#### 2.2.2.2.1 Узагальнення

Наслідування є однією з фундаментальних основ об'єктно-орієнтованого програмування, у якому клас «отримує» всі атрибути і операції класу, нащадком якого він є, і може перевизначати або змінювати деякі з них, а також додавати власні атрибути і операції.

У UML пов'язування *Узагальнення* між двома класами розташовує їх у вузлах ієрархії, яка відповідає концепції успадкування класу-нащадка від базового класу. У UML узагальнення буде показано у вигляді лінії, яка поєднує два класи, зі стрілкою, яку спрямовано від базового класу.



Наочний показ узагальнення у UML

#### 2.2.2.2.2 Асоціації

Асоціація означає взаємозв'язок між класами, вона є базовим семантичним елементом і структурою для багатьох типів «з'єднань» між об'єктами.

Асоціації є тим механізмом, який надає об'єктам змогу обмінюватися даними між собою. Асоціація описує з'єднання між різними класами (з'єднання між дійсними об'єктами називається об'єктним з'єднанням, або *зв'язком*).

Асоціації можуть виконувати роль, яка визначає призначення асоціації і може бути одно- чи двосторонньою (другий варіант означає, що у межах зв'язку кожен з об'єктів може надсилати повідомлення іншому, перший же — варіанту, коли лише один з об'єктів знає про існування іншого). Крім того, кожен з кінців асоціації має значення численності, яке визначає кількість об'єктів на відповідному кінці асоціації, які можуть мати зв'язок з одним з об'єктів на іншому кінці асоціації.

У UML асоціації позначаються лініями, що з'єднують класи, які беруть участь у зв'язку, крім того, може бути показано роль і численність кожного з учасників зв'язку. Численність буде

показано у вигляді діапазону [мін..макс] невід'ємних чисел, зірочка (\*) на боці максимального значення позначає нескінченність.



Наочний показ асоціації у UML

### 2.2.2.2.3 Агрегація

Агрегації є особливим типом асоціацій, за якого два класи, які беруть участь у зв'язку не є рівнозначними, вони мають зв'язок типу «ціле-частина». За допомогою агрегації можна описати, яким чином клас, який грає роль цілого, складається з інших класів, які грають роль частин. У агрегаціях клас, який грає роль цілого, завжди має численність рівну одиниці.

У UML агрегації буде показано асоціаціями, у яких з боку цілої частини буде намальовано ромб.

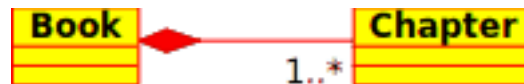


Наочний показ зв'язку агрегації у UML

### 2.2.2.2.4 Композиція

Композиції — це асоціації, які відповідають *дуже сильній* агрегації. Це означає, що у композиціях ми також маємо справу з співвідношеннями ціле-частина, але тут зв'язок є настільки сильним, що частини не можуть існувати без цілого. Вони існують лише у межах цілого, після знищення цілого буде знищено і його частини.

У UML композиції буде показано як асоціації з зафарбованим ромбом з боку цілого.



### 2.2.2.3 Інші елементи діаграми класів

Окрім класів на діаграмах класів можуть міститися і деякі інші елементи.

#### 2.2.2.3.1 Інтерфейси

Інтерфейси — це абстрактні класи, тобто з них не можна напряму створювати екземпляри. У інтерфейсах можуть міститися операції, але не атрибути. Класи можуть бути нащадками інтерфейсів (за допомогою асоціації реалізації), а з цих діаграм можна потім створювати сутності.

#### 2.2.2.3.2 Типи даних

Типи даних — це базові елементи, з яких типово будується мова програмування. Типовими прикладами є цілі числа і булеві значення. Вони не можуть мати зв'язків з класами, але класи можуть мати зв'язки з ними.

#### 2.2.2.3.3 Переліки

Переліки є простими списками значень. Типовим прикладом є перелік днів тижня. Пункти переліків називаються літералами переліків. Подібно до типів даних, переліки не можуть мати зв'язків з класами, але класи можуть мати зв'язки з переліками.

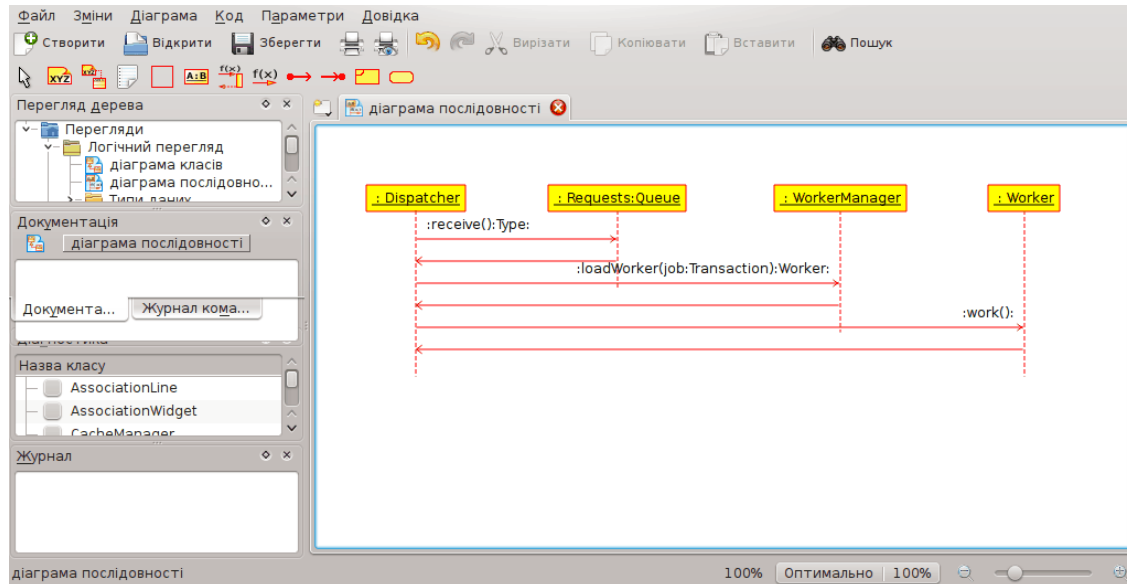
### 2.2.2.3.4 Пакунки

Пакункам відповідають простори назв у мовах програмування. На діаграмі пакунки використовуються для позначення частин системи, у яких міститься декілька класів, може навіть сотні класів.

## 2.2.3 Діаграми послідовностей

На діаграмах послідовностей буде показано обмін повідомленнями (тобто виклик методів) між декількома об'єктами у окремій обмеженій часом ситуації. Об'єкти є екземплярами класів. Основний наголос на діаграмах послідовностей робиться на порядок і моментах часу, у які повідомлення надсилаються об'єктам.

На діаграмах послідовностей об'єкти буде показано вертикальними штриховими лініями з назвою об'єкта над ними. Вісь часу також має вертикальний напрямок, її спрямовано вниз, повідомлення, які надсилаються від одного об'єкта до іншого, буде позначено стрілками з назвами операції і параметрів.



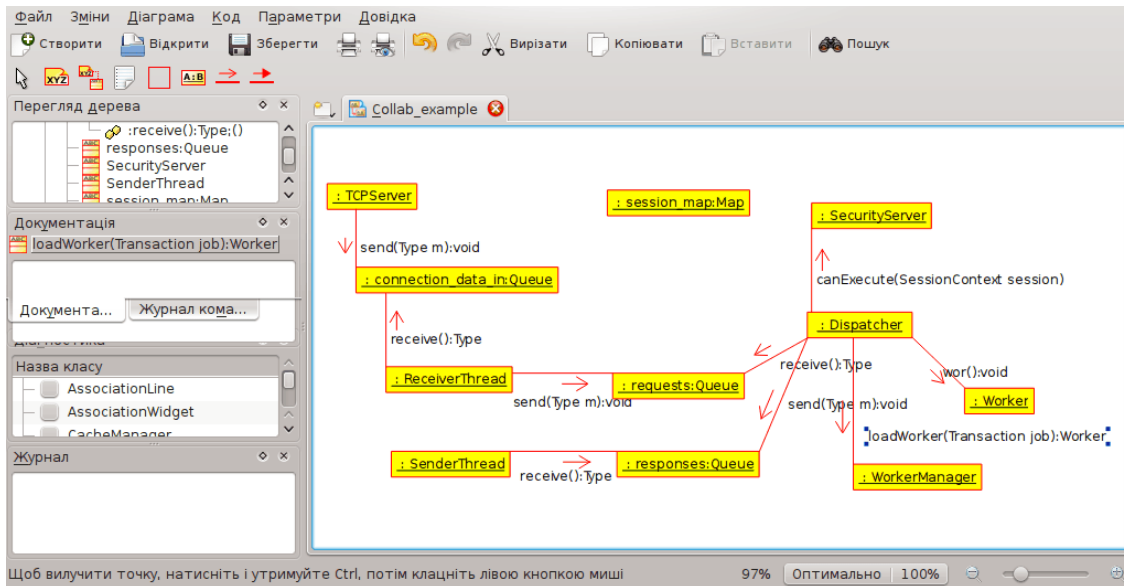
Показ діаграми послідовностей у Umbrello UML Modeller

Повідомлення можуть бути або синхронними, звичайного типу повідомленнями, за виклику яких керування передається викликаному об'єкту до завершення виконання методу, або асинхронними, за виклику яких керування передається назад напряму об'єкту, який здійснивав виклик. За використання синхронного повідомлення збоку від викликаного об'єкта буде показано вертикальний блок, який показуватиме перебіг виконання програми.

### 2.2.4 Діаграми співпраці

На діаграмах співпраці 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 співпраці основний наголос робиться на взаємодії між об'єктами та її топології на передньому плані.

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



Показ діаграми співпраці у Umbrello UML Modeller

## 2.2.5 Діаграма станів

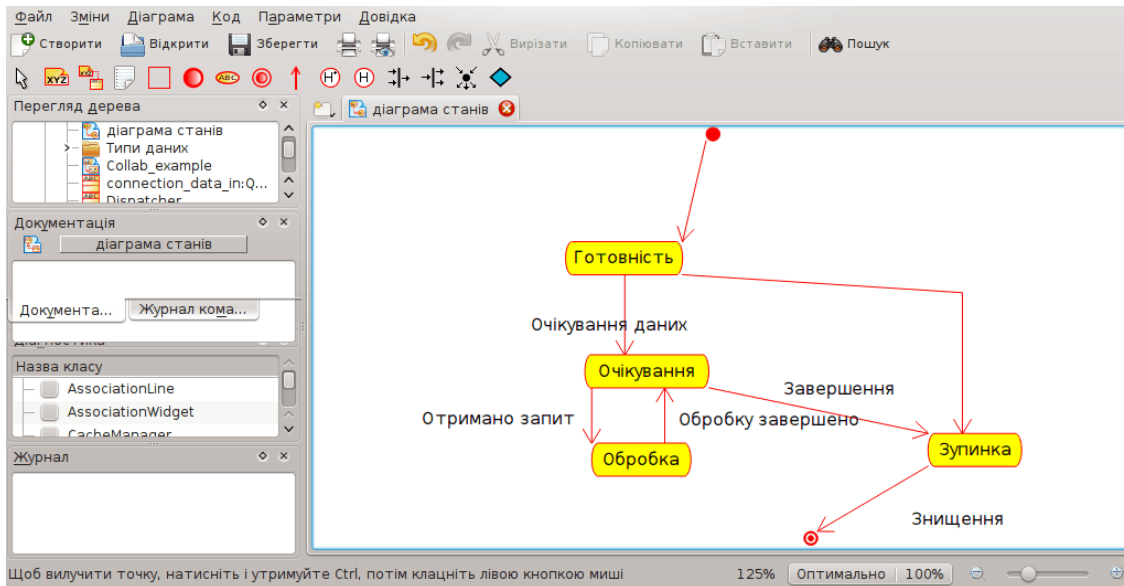
На діаграмах станів зображають різні стани об'єкта під час його існування і стимули, які призводять до переходу об'єкта з одного стану у інший.

На діаграмах стану об'єкти розглядаються як *машини станів* або скінченні автомати, які можуть перебувати у одному зі станів скінченного набору станів, і які можуть змінювати цей стан через вплив одного зі стимулів зі скінченного набору стимулів. Наприклад, об'єкт типу *Сервер мережі* може перебувати у одному з таких станів протягом існування:

- Готовність
- Очікування
- Робота
- Зупинка

а подіями, які можуть спричинити зміну стану об'єкта можуть бути

- Створення об'єкта
- Об'єкт отримує повідомлення «очікувати»
- Клієнт надсилає запит на з'єднання мережею
- Клієнт перериває запит
- Запит виконано і перервано
- Об'єкт отримує повідомлення «зупинка»
- тощо



Показ діаграми станів у Umbrello UML Modeller

### 2.2.5.1 Стан

Будівельними цеглинками діаграм станів є стани. Стан належить лише одному класу і відповідає переліку значень атрибутів, які може приймати клас. У UML стан описує внутрішній стан об'єкта одного з окремих класів

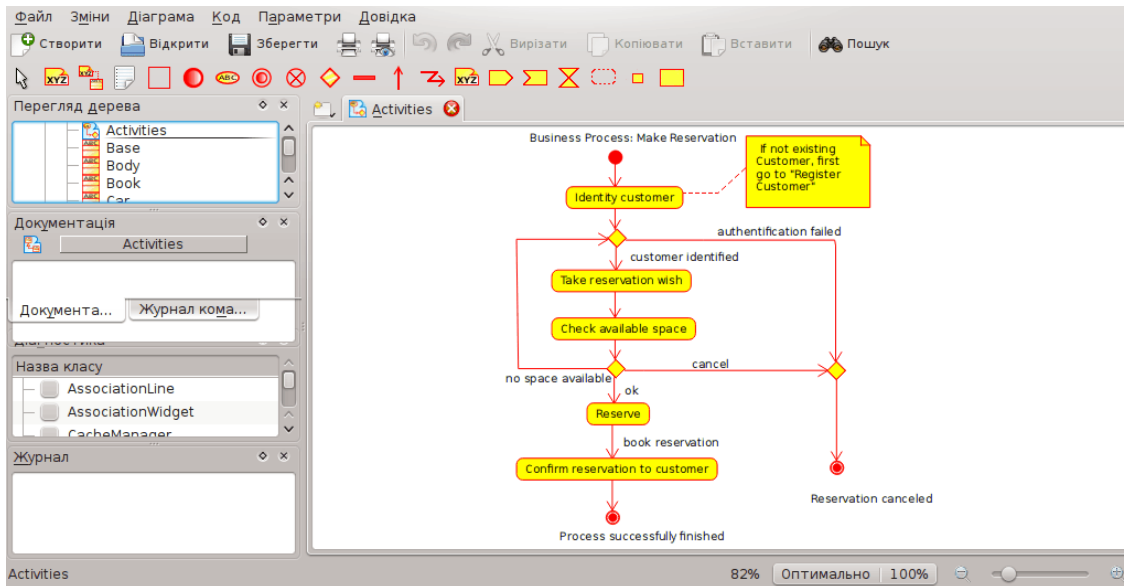
Зауважте, що не кожен змінюваний атрибут об'єкта має бути показано станом, станам відповідають лише ті зміни, які значно впливають на виконання об'єктом завдань.

Існує два особливих типи станів: початок і кінець. Їх особливість полягає у тому, що не існує жодної події, яка може спричинити повернення об'єкта до його початкового стану, так само, не існує жодної події, яка б могла повернути об'єкт зі стану кінця, тільки-но він його досягне.

### 2.2.6 Діаграма діяльності

На діаграмі діяльності буде показано послідовність актів дій системи на основі Діяльностей. Діаграми діяльності є особливою формою діаграм стану, на яких містяться лише (або головним чином) діяльності.





Показ діаграми діяльності у Umbrello UML Modeller

Діаграми діяльності подібні до процедурних діаграм потоку, але відрізняються від них тим, що діяльності точно прив'язано до об'єктів.

Діаграми діяльності завжди пов'язано з класом, операцією або випадком використання.

На діаграмах діяльності може бути показано як послідовні, так і паралельні діяльності. Паралельне виконання буде показано за допомогою піктограм Розділити/Чекати, для діяльностей, які виконуються паралельно, неважливим є порядок їх обробки (їх може бути виконано одночасно або одну за одну).

### 2.2.6.1 Діяльність

Діяльність є окремим кроком у процесі. Одній діяльності відповідає окремий стан у системі з внутрішньою діяльністю і, принаймні, одна вихідна транзакція. Крім того, діяльності можуть мати декілька вихідних транзакцій, якщо умови цих транзакцій є різними.

Діяльності можуть формувати ієрархічні структури, це означає, що діяльність може бути складено з декількох «менших» діяльностей, у цьому випадку вхідні і вихідні транзакції мають відповідати вхідним і вихідним транзакціям докладної діаграми.

### 2.2.7 Допоміжні елементи

У UML є декілька елементів, які не мають реального семантичного змісту для моделі, але допомагають прояснити частини діаграми. Цими елементами є

- Рядки тексту
- Текстові нотатки і якорі
- Блоки

Рядки тексту можуть знадобитися, якщо до діаграми слід додати коротку текстову інформацію. Вони є довільно розташованим тестом і не мають значення для самої моделі.

Нотатками можна скористатися для додавання докладніших відомостей щодо об'єкта або певної ситуації. У них є велика перевага у тому, що нотатки можна пов'язати з елементами UML, щоб було видно, що нотатка «стосується» певного об'єкта або ситуації.

Блоки є довільно розташованими прямокутниками, які можна використовувати для групування об'єктів діаграми, яке зробить діаграму зрозумілішою. Вони не мають логічного навантаження у межах моделі.

## 2.2.8 Діаграми компонентів

На діаграмах компонентів буде показано компоненти програмного забезпечення (або технології компонентів, такі як KParts, компоненти CORBA або Java Beans, або просто розділи системи, які чітко відрізняються один від одного), а також елементи, з яких вони складаються, такі як файли з початковими кодами, програмні бібліотеки або таблиці реляційних баз даних.

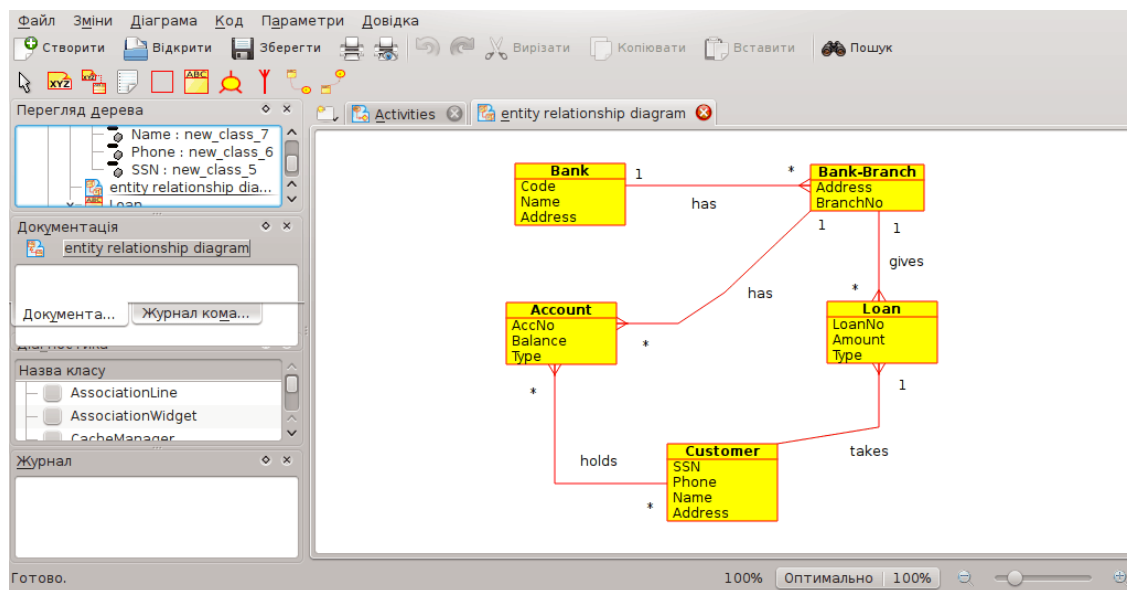
Компоненти можуть мати інтерфейси (тобто абстрактні класи з операціями), які надають змогу створювати асоціації між компонентами.

## 2.2.9 Діаграми впровадження

На діаграмах впровадження буде показано екземпляри компонентів та їх асоціації. На них буде показано вузли, які є фізичними ресурсами, типово, окремими комп'ютерами. Крім того, на них показують інтерфейси і об'єкти (екземпляри класів).

## 2.2.10 Діаграми взаємозв'язків сутностей

На діаграмах взаємозв'язку сутностей (діаграмах ВС (ER)) показують концептуальний дизайн програм для роботи з базами даних. На них показують різноманітні сутності (концепти) у інформаційній системі і існуючі взаємозв'язки і обмеження між ними. Розширення діаграм взаємозв'язку сутностей називають «Розширеними діаграмами взаємозв'язку сутностей» або «Покращеними діаграмами взаємозв'язку сутностей» (EER), їх використовують для інтеграції методик компонування орієнтованих на об'єкти у діаграми ВС.



Показ діаграми взаємозв'язків сутностей у Umbrello UML Modeller

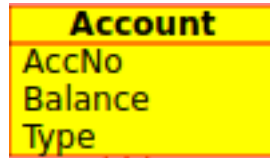
### 2.2.10.1 Сутність

*Сутністю* є будь-яке з понять реального світу, яке має окреме існування. Нею може бути об'єкт фізичної природи (наприклад, комп'ютер або робот), нею може бути і об'єкт зі концептуальним існуванням (університетський курс). Кожна з сутностей має набір атрибутів, які описують властивості сутності.

*Зауваження:* не існує стандартів позначень діаграм ВС (ER). У різних працях з цього питання використовують різні позначення. Поняття і позначення для діаграм у Umbrello UML

Modeller запозичено з книги: *Elmasri R. and Navathe S. (2004). Fundamentals of Database Systems 4th edn. Addison Wesley*

На діаграмі BC(ER) сутності позначаються прямокутниками з назвою у верхній частині, на них також може бути показано атрибути сутності у іншому «відсіку» прямокутника.



Наочний показ сутності на діаграмі взаємозв'язку сутностей

### 2.2.10.1.1 Атрибути сутності

На діаграмах взаємозв'язку сутностей атрибути сутностей показуються назвами у окремій ділянці сутності, якій вони належать.

### 2.2.10.1.2 Обмеження

Обмеження на діаграмах взаємозв'язку сутностей визначають обмеження на дані у інформаційній схемі.

У Umbrello UML Modeller підтримуються чотири типи обмежень:

- *Головний ключ:* Набір атрибутів, оголошених як *головний ключ* є унікальним для сутності. У сутності має існувати лише один головний ключ, а жоден з складових атрибутів цього ключа не повинен дорівнювати NULL.
- *Унікальний ключ:* Набір атрибутів, оголошених як *унікальний ключ* є унікальним для сутності. У сутності може бути декілька унікальних обмежень. Складові атрибути ключа можуть приймати значення NULL. Унікальні ключі і головні ключі однозначно визначають рядок у таблиці (сутність).
- *Сторонній ключ:* Сторонній ключ є довідковим обмеженням між двома таблицями. За стороннім ключем визначається стовпчик або набір стовпчиків у одній (тій, для якої потрібна довідка) таблиці, яка стосується стовпчика або набору стовпчиків у іншій (еталонній) таблиці. Стовпчики у еталонній таблиці повинні мати форму головного ключа і унікального ключа.
- *Обмеження перевірки:* Обмеження перевірки (також відоме як обмеження перевірки таблиці) є умовою, яка визначає коректність даних під час додавання або оновлення запису у таблиці реляційної бази даних. Обмеження перевірки застосовується до кожного з рядків таблиці. Обмеження має бути предикативним. Воно може стосуватися одного або декількох стовпчиків таблиці.

Приклад: вартість  $\geq 0$

## 2.2.11 Концепції розширеної діаграми взаємозв'язку сутностей

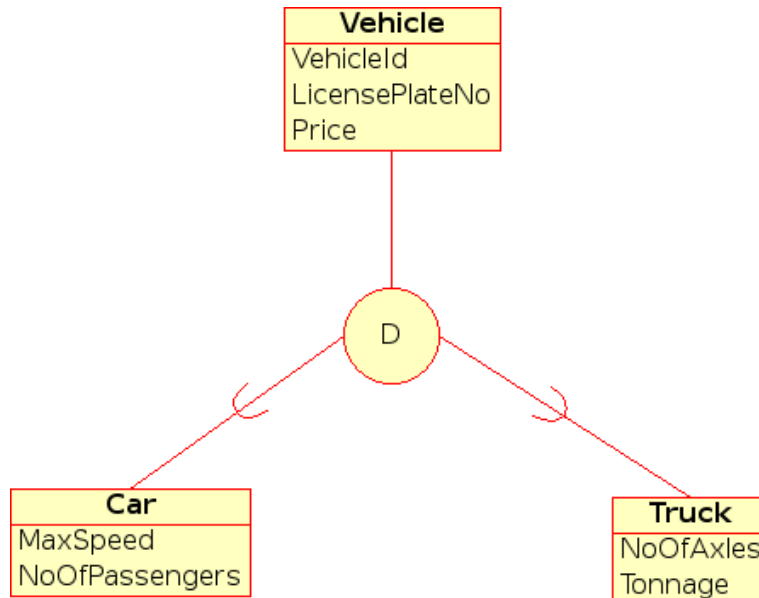
### 2.2.11.1 Спеціалізація

Спеціалізація — це один зі способів формування нових сутностей за допомогою сутностей, які вже було визначено. Нові сутності, відомі як сутності-нащадки, успадковують атрибути сутностей, які вже існували, і які називають базовими. Спеціалізація призначена для спрощення повторного використання даних з невеликою модифікацією або взагалі без модифікації.

У Umbrello UML Modeller можна визначати несумісну і перекриту спеціалізацію

### 2.2.11.1.1 Несумісна спеціалізація

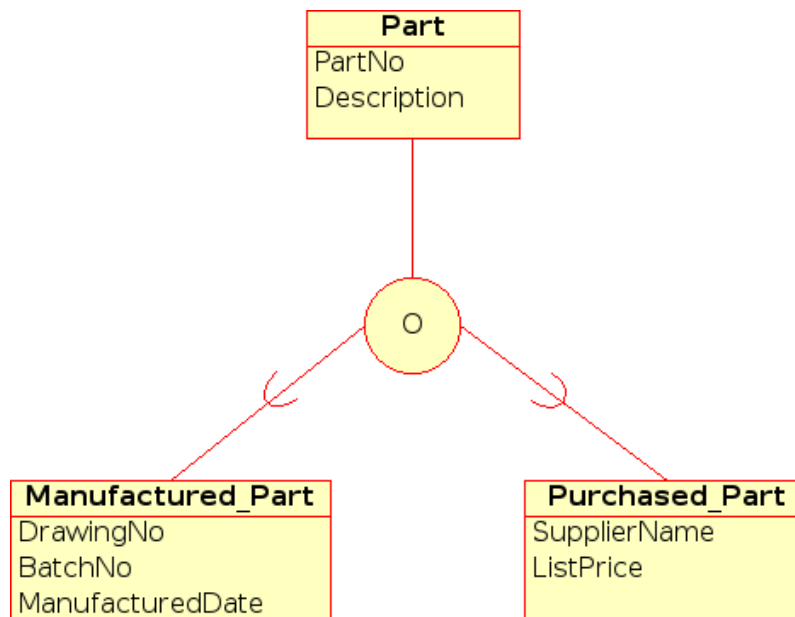
Несумісна спеціалізація вказує, що підкласи спеціалізації не повинні перетинатися. Це означає, що сутність може бути членом не більше, ніж однієї з сутностей-нащадків спеціалізації.



Наочний показ несумісної спеціалізації на діаграмі EER

### 2.2.11.1.2 Перекрита спеціалізація

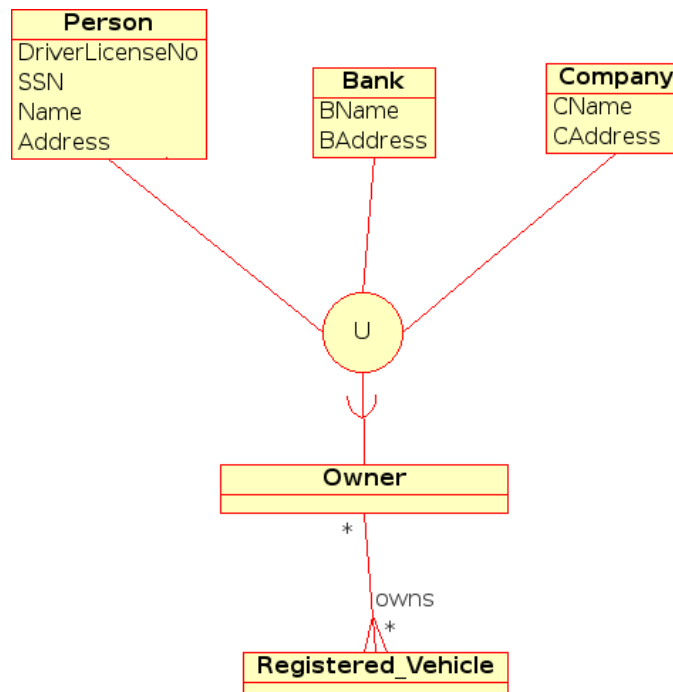
Якщо сутності-нащадки не обмежено вимогою відсутності перетинів, такий набір сутностей належить до перекритої спеціалізації. Це означає, що однакові сутності реального світу можуть бути членами декількох сутностей-нащадків спеціалізації/



Наочний показ перекритої спеціалізації на діаграмі EER

### 2.2.11.1.3 Категорія

Говорять, що сутність-нащадок є *Категорією*, якщо їй відповідає збірка об'єктів, яка є підмножиною об'єднання різних типів сутностей. Категорії беруть участь у моделюванні, якщо виникає потреба у окремому співвідношенні суперкласі/підкласі з декількома суперкласами, де суперкласу відповідають різні типи сутностей. (Дуже схоже на кратне успадкування у об'єктно-орієнтованому програмуванні).



Наочний показ категорії на діаграмі EER

## Розділ 3

# Як працювати з Umbrello UML Modeller

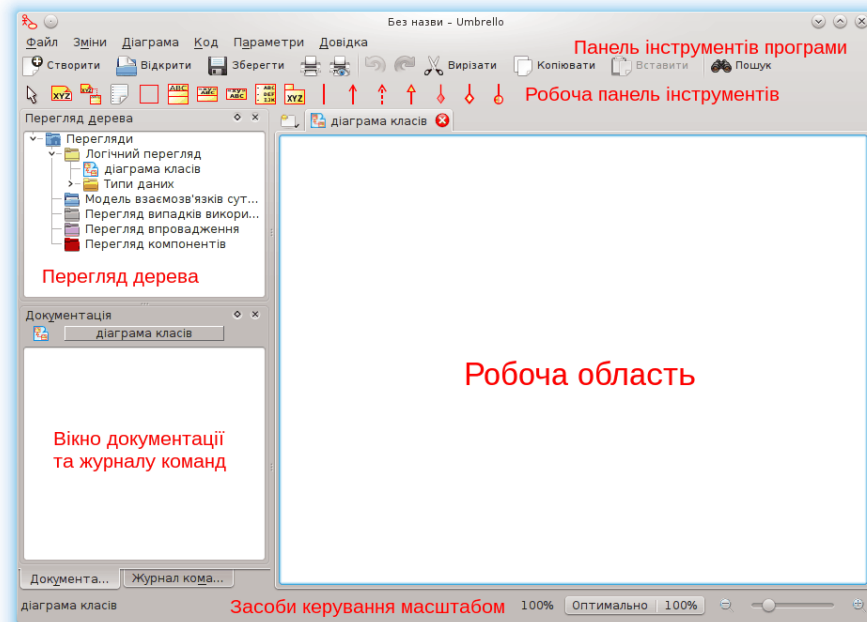
У цій главі ви познайомитеся з інтерфейсом користувача Umbrello UML Modeller та ознайомитеся з відомостями, які потрібні будуть вам для того, щоб почати моделювання. Доступ до усіх дій у Umbrello UML Modeller можна отримати за допомогою меню і панелей інструментів, але у Umbrello UML Modeller передбачено і широке використання контекстних меню, які можна викликати клацанням правою кнопкою миші. Ви можете навести вказівник майже на будь-який елемент у робочій області або перегляді деревом Umbrello UML Modeller і клацнути правою кнопкою, щоб відкрити контекстне меню з пунктами для найпоширеніших дій, які можна застосувати до окремого елемента, над яким ви працюєте. На початку деякі з користувачів можуть відчувати певні незручності, оскільки вони звикли до роботи за допомогою меню та панелей інструментів, але після того, як вони при звичаються до клацання правою кнопкою миші, такий спосіб роботи значно пришвидшить розробку.

### 3.1 Інтерфейс користувача

Головне вікно Umbrello UML Modeller поділено на три області, які призначено для спостереження за загальним виглядом усієї системи і пришвидшення доступу до різних діаграм під час роботи над вашою моделлю.

Ці області носять такі назви:

- Перегляд дерева
- Робоча область
- Вікно документації та журналу команд



*Інтерфейс користувача Umbrello UML Modeller*

### 3.1.1 Перегляд дерева

Перегляд дерева, зазвичай, розташовано у верхній лівій частині вікна програми, у цій області буде показано список всіх діаграм, класів, акторів і випадків використання, з яких складається ваша модель. За допомогою перегляду дерева ви зможете спостерігати за переліком елементів, з яких складається ваша модель. Крім того, за допомогою перегляду деревом ви зможете швидко перемикатися між різними діаграмами вашої моделі і вставляти елементи з вашої моделі у поточну діаграму.

Якщо ви працюєте над моделлю, у якій набагато більше, ніж декілька класів і діаграм, перегляд дерева може допомогти вам бачити загальну картину шляхом впорядкування елементів вашої моделі за теками. Ви можете створювати теки вибором відповідного пункту з контекстного меню (його можна відкрити наведенням вказівника на позначку однієї з тек у перегляді дерева з наступним клацанням правою кнопкою миші), ви також можете впорядковувати ваші елементи пересуванням їх до відповідної теки (перетягування зі скиданням)

### 3.1.2 Вікно документації та журналу команд

Вікно документації та журналу команд — це маленьке вікно, розташоване у лівому нижньому куті вікна Umbrello UML Modeller. У цій області буде показано коротенький перегляд документації щодо поточного вибраного елемента та журнал команд вашого поточного сеансу. Вікно документації має порівняно невеликі розміри, тому що його призначено для того, щоб ви могли одним оком зазирнути до документації елемента та журналу команд без зайвої витрати місця на екрані. Якщо вам потрібно переглянути детальнішу документацію, ви завжди можете відкрити вікно властивостей елемента.

### 3.1.3 Робоча область

Робоча область є основним вікном у Umbrello UML Modeller, саме у ньому і виконуються дії програми. Ви можете скористатися робочою областю для редагування і перегляду діаграм вашої моделі. У робочій області буде показано поточну активну діаграму. У поточній версії у робочій області може бути показано лише одну діаграму одночасно.

## 3.2 Створення, завантаження і збереження моделей

Першим, що ви маєте зробити, щоб розпочати використання Umbrello UML Modeller, є створення моделі, над якою ви працюватимете. Одразу після запуску Umbrello UML Modeller завжди завантажуватиме останню використану модель або створюватиме нову порожню модель (її параметри залежатимуть від набору налаштувань, визначених у діалоговому вікні налаштування). Такі параметри надають вам змогу одразу перейти до роботи.

### 3.2.1 Нова модель

Якщо вам потрібно буде під час роботи створити нову модель, ви можете зробити це за допомогою пункту **Створити** з меню **Файл** або натисканням на кнопку **Створити** на панелі інструментів програми. Якщо ви працювали над моделлю, у якій залишилися незбережені зміни, Umbrello UML Modeller запитає вас, чи слід зберігати зміни, до завантаження нової моделі.

### 3.2.2 Збереження моделі

Ви можете будь-якого моменту зберегти вашу модель за допомогою пункту **Зберегти** з меню **Файл** або натискання кнопки **Зберегти** на панелі інструментів програми. Якщо вам потрібно зберегти вашу модель під новою назвою, ви можете скористатися пунктом **Зберегти як** з меню **Файл**.

Крім того, щоб полегшити вам користування програмою, Umbrello UML Modeller може автоматично зберігати вашу роботу через певні проміжки часу. Увімкнути або вимкнути таке автоматичне зберігання, а також встановити проміжок часу між послідовними зберіганнями можна у меню **Параметри** Umbrello UML Modeller

### 3.2.3 Завантажити модель

Для завантаження вже існуючої моделі ви можете скористатися пунктом **Відкрити** меню **Файл** або кнопкою **Відкрити** на панелі інструментів. Завантаження нещодавно використаних моделей можна також здійснити за допомогою підменю **Відкрити недавні** меню **Файл**, у цьому підменю ви знайдете список моделей, якими ви користуєтеся найчастіше.

Umbrello UML Modeller може одночасно працювати лише з однією моделлю, отже, якщо ви накажете програмі завантажити модель, а вашу поточну модель було змінено з часу її останнього збереження, Umbrello UML Modeller запитає у вас про те, чи слід зберігати ці зміни, щоб запобігти втраті результатів роботи. Ви можете одночасно запустити декілька екземплярів Umbrello UML Modeller, а також виконувати дії з копіювання та вставки між цими двома екземплярами.

## 3.3 Редагування моделей

Загалом, у Umbrello UML Modeller існує два способи редагування елементів вашої моделі.

- Редагування елементів моделей напряму за допомогою перегляду дерева
- Редагування елементів моделей за допомогою діаграми

За допомогою контекстних меню різних пунктів перегляду дерева ви можете додавати, вилучати і змінювати майже всі елементи вашої моделі. Наведенням вказівника миші на позначки тек у перегляді дерева з наступним клацанням правою кнопкою миші відкриє перед вами меню, у якому ви знайдете пункти, призначені для створення діаграм різних типів, а також, залежно від того, чи належить тека до *Перегляду випадків використання* або до *Логічного перегляду*, Акторів, Випадків використання, Класів тощо.

Після додавання елементів до вашої моделі ви зможете змінювати елементи за допомогою їх діалогових вікон властивостей, доступ до яких можна отримати за допомогою пункту



*Властивості* з контекстного меню, яке буде показано після клацання правою кнопкою миші на пункті перегляду дерева.

Крім того, ви можете змінювати вашу модель за допомогою створення або зміни елементів на діаграмах. Докладніше цей процес описано у наступних розділах.

## 3.4 Додавання і видалення діаграм

Ваша модель UML складатиметься з набору елементів UML і зв'язків між ними. Але ви не зможете безпосередньо побачити вашу модель, для того, щоб її оглянути вам потрібні будуть *Діаграми*.

### 3.4.1 Створення діаграм

Щоб створити нову діаграму у вашій моделі просто оберіть тип діаграми, яка вам потрібна, з підменю **Створити** меню **Діаграма**, а потім вкажіть назву діаграми. Програма створить діаграму і зробить її активною, — ви негайно зможете побачити її у перегляді дерева.

Пам'ятайте, що у Umbrello UML Modeller широко використовуються контекстні меню: ви можете навести вказівник на теку у перегляді дерева, клацнути правою кнопкою миші і обрати потрібний тип діаграми з підменю **Створити** контекстного меню. Зауважте, що ви можете створювати діаграми випадків використання лише у теках перегляду випадків використання, а інші типи діаграм можна створювати лише у теках логічного перегляду.

### 3.4.2 Видалення діаграм

Якщо вам потрібно видалити діаграму з вашої моделі, для цього слід зробити її активною і обрати пункт **Видалити** з меню **Діаграма**. Такого ж результату можна досягти вибором пункту **Видалити** з контекстного меню перегляду дерева.

Оскільки видалення діаграми є відповідальною дією, яка може призвести до втрати результатів роботи, якщо її здійснено випадково, Umbrello UML Modeller проситиме вас підтвердити дію з видалення перед її виконанням. Після видалення діаграми і збереження файла скасувати наслідки дії з видалення буде неможливо.

### 3.4.3 Перейменування діаграм

Якщо ви бажаєте змінити назву існуючої діаграми, ви можете зробити це за допомогою пункту «Перейменувати» з контекстного меню діаграми у перегляді дерева.

Іншим способом перейменування діаграми є використання діалогового вікна властивостей, яке можна відкрити за допомогою пункту «Властивості» контекстного меню або подвійного клацання на позначці теки у перегляді дерева.

## 3.5 Редагування діаграм

Під час роботи з діаграмою Umbrello UML Modeller намагатиметься направляти вашу роботу застосуванням певних простих правил щодо того, які елементи можна використовувати на діаграмі обраного типу, а також щодо типів взаємозв'язку, які можуть між цими елементами існувати. Якщо ви є експертом з UML, ймовірно, ви навіть не помітите цього, але така робота програми допоможе новачкам у UML створювати діаграми у відповідності зі стандартами.

Після створення діаграми настане час її редагування. Ось тут ви й зауважите (майже непомітну для новачків) різницю між редагуванням вашої діаграми і редагуванням *моделі*. Як ви вже знаєте, діаграми є *переглядами* вашої моделі. Наприклад, якщо ви створили клас редагування діаграми класів, ви насправді редагували і діаграму, і вашу модель. Якщо ж ви змінили колір або інші параметри показу класу на діаграмі класів, ваші зміни стосуватимуться лише діаграми, — у моделі нічого не зміниться.

### 3.5.1 Як вставити елементи

Одною з перших операцій, яка знадобиться вам під час редагування нової діаграми, є вставка елементів до неї (класів, акторів, випадків використання, тощо). Загалом, існує два способи виконання цієї дії:

- Перетягування існуючих елементів до вашої моделі у перегляді дерева
- Створення нових елементів у вашій моделі і одночасне додавання цих елементів на вашу діаграму за допомогою одного з інструментів редагування на панелі інструментів.

Щоб вставити елементи, які вже існують у вашій моделі, просто перетягніть їх з перегляду дерева і скиньте на місце, де вони мають знаходитися на діаграмі. Крім того, ви завжди зможете пересунути елементи на вашій діаграмі за допомогою інструмента вибору.

Другим способом додавання елементів на вашу діаграму є використання інструментів редагування панелі інструментів (зауважте, що подібні дії додаватимуть елементи до вашої моделі).

Типово, панель інструментів розташовано у верхній частині вікна. Список інструменти, доступ до яких можна отримати за допомогою цієї панелі (кнопок, які ви на ній бачите), залежатиме від типу діаграми, над якою ви працюєте. Кнопку поточного вибраного інструмента буде активовано, перемкнутися на інструмент вибору можна натисканням клавіші **Esc**.

Якщо ви обрали інструмент редагування на робочій панелі інструментів (наприклад інструмент для додавання класів), вигляд вказівника миші буде змінено на хрестик, і ви зможете вставляти елементи до вашої моделі одноразовим клацанням лівою клавішею миші у полі вашої діаграми. Зауважте, що у елементів UML мають бути *унікальні назви*. Отже, якщо на одній з ваших діаграм є клас з назвою «ClassA», і ви використали інструмент додавання класу на іншій діаграмі, ви не зможете дати цьому класу назву «ClassA». Якщо це мають бути різні класи, вам слід надати їм різні назви. Якщо ви бажаєте додати на вашу діаграму *той самий* елемент, інструмент додавання класу вам не знадобиться. Замість цього, вам слід перетягнути і скинути клас з перегляду дерева.

### 3.5.2 Вилучення елементів

Ви можете вилучати будь-які елементи вибором пункту **Вилучити** з їх контекстного меню.

Знову ж таки, існує *значна* різниця між вилученням об'єкта з діаграми і вилученням об'єкта з вашої моделі: якщо ви вилучите об'єкт з діаграми, ви лише вилучите об'єкт з діаграми, — елемент залишиться частиною вашої моделі, і якщо у інших діаграмах використано цей елемент, вони ніяким чином не зміняться. Якщо ж, з іншого боку, ви вилучите елемент з перегляду дерева, ви вилучите елемент з вашої *моделі*. Оскільки у моделі більше не існуватиме відповідного елемента, його буде автоматично вилучено з усіх діаграм, де його було використано.

### 3.5.3 Редагування елементів

Ви можете редагувати більшість елементів UML вашої моделі і діаграм за допомогою діалогового вікна властивостей елемента, де можна вибрати відповідні параметри. Щоб змінити властивості об'єкта, скористайтеся пунктом **Властивості** його контекстного меню (можна викликати клацанням правою кнопкою миші). У кожного з елементів діалогове вікно складатиметься з декількох сторінок, на яких ви зможете налаштувати параметри, що стосуються цього елемента. Для деяких елементів (наприклад якорів) ви можете визначити лише парочку параметрів, на зразок назви об'єкта і документації, для інших же елементів (наприклад класів) ви зможете змінити атрибути і операції, визначити вигляд елемента на діаграмі (повний підпис операції або просто назви операцій тощо) і навіть вказати кольори, якими слід показувати лінії і розфарбовувати відповідник класу на діаграмі.

Крім того, діалогове вікно властивостей для елементів UML можна відкрити подвійним клацанням на елементі, якщо ви використовуєте інструмент вибору (стрілочку).

Зауважте, що пункт властивостей можна обрати і з контекстного меню елементів у перегляді дерева. Таким чином можна змінювати властивості діаграм, наприклад визначити, чи слід показувати ґратку.

### 3.5.4 Редагування класів

Хоча аспекти редагування властивостей всіх об'єктів і було розглянуто у попередньому розділі, класи заслуговують окремого розділу, оскільки ці об'єкти є складнішим і мають більшу кількість параметрів, ніж більшість інших елементів UML.

У діалоговому вікні класу ви можете встановити будь-які властивості, від кольорів, якими буде намальовано позначку класу, до операцій і атрибутів, які матиме клас.

#### 3.5.4.1 Загальні параметри класу

Призначення сторінки загальних параметрів властивостей класу очевидне з назви цієї сторінки. Тут ви можете змінити назву класу, видимість, документацію тощо. Ця сторінка буде доступною завжди.

#### 3.5.4.2 Параметри атрибутів класу

На сторінці параметрів атрибутів ви зможете додати, змінити або вилучити атрибути (змінити) класу. За допомогою кнопок зі стрілками, розташованих у боковій частині сторінки, ви зможете пересувати атрибути вгору або вниз списком. Ця сторінка буде доступною завжди.

#### 3.5.4.3 Параметри операцій класу

Подібно до сторінки параметрів атрибутів, на сторінці параметрів операцій ви зможете додати, змінити або вилучити операції з вашого класу. Під час додавання або зміни операції ви вводите базові дані до діалогового вікна *Властивості операції*. Щоб додати параметри до вашої операції, вам слід натиснути кнопку **Новий параметр**, — буде відкрито діалогове вікно *Властивості параметра*. Ця сторінка буде доступною завжди.

#### 3.5.4.4 Параметри шаблонів класу

На цій сторінці ви зможете додати шаблони класів, які є невизначеними класами або типами даних. У Java 1.5 такі шаблони називаються «Generic».

#### 3.5.4.5 Сторінка асоціацій класів

На сторінці **Асоціації класу** буде показано всі асоціації класу на поточній діаграмі. Наведення вказівника миші на позначку асоціації з наступним подвійним клацанням лівою кнопкою миші відкриє діалогове вікно з її властивостями, у ньому, залежно від типу асоціації, ви зможете змінити деякі параметри, серед яких встановлення численності і назва ролі. Якщо параметри асоціації змінювати не можна, у діалоговому вікні властивостей асоціації буде лише показано параметри, змінювати ви зможете лише документацію щодо цієї асоціації.

Ця сторінка стане доступною, лише якщо ви відкриєте вікно властивостей класу з діаграми. Якщо ви відкриєте вікно властивостей класу за допомогою контекстного меню перегляду дерева, ця сторінка доступна не буде.

#### 3.5.4.6 Сторінка показу класу

На сторінці **Параметри показу** ви можете вказати параметри, які буде показано на діаграмі. Клас може бути показано як прямокутник з назвою класу у ньому (корисно, якщо на вашій діаграмі багато класів, або подробиці щодо класу вам на даний момент не цікаві) або повноцінно, з показом паунків, стереотипів, атрибутів і операцій з підписами і позначками видимості.

Ви можете обрати параметри на цій сторінці залежно від об'єму відомостей, які ви бажаєте бачити. Зміни, які ви зможете зробити за допомогою цієї сторінки, стосуватимуться лише

*параметрів показу* діаграми. Це означає, що «приховування» операції класу призведе лише до того, що їх не буде показано на діаграмі, але операція залишиться частиною вашої моделі. Ця сторінка буде доступною, лише якщо ви відкриєте вікно властивостей класу з діаграми. Якщо ви відкриєте вікно властивостей з перегляду дерева, цю сторінку показано не буде, оскільки у такому випадку параметри показу не матимуть сенсу.

#### 3.5.4.7 Сторінка стилю класу

На сторінці **Стиль віджетів** ви зможете налаштувати кольори, які буде використано для ліній і розфарбування віджета. Очевидно, що ці параметри матимуть сенс, лише якщо клас показано на діаграмі, якщо ви відкриєте вікно властивостей класу з перегляду дерева, цієї сторінки показано не буде.

### 3.5.5 Асоціації

Асоціації пов'язують один з одним два об'єкти UML. Зазвичай, асоціації визначаються для зв'язку між двома класами, але певні типи асоціацій можуть існувати і для випадків використання і акторів.

Щоб створити асоціацію, оберіть відповідний інструмент на робочій панелі інструментів (типово, Асоціація, Узагальнення, Агрегація тощо), наведіть вказівник миші на перший з елементів асоціації і клацніть лівою кнопкою миші, потім наведіть вказівник миші на другий елемент асоціації і ще раз клацніть лівою кнопкою миші. Зауважте, що слід виконати два окремих клацання, а *не* перетягувати один об'єкт на інший.

Якщо ви намагатиметеся створити асоціацію, яка суперечитиме специфікації UML, Umbrello UML Modeller заборонить вам створення асоціації, ви отримаєте повідомлення про помилку. Таке може статися, наприклад, якщо існує узагальнення від класу А до класу В, а ви намагаєтесь створити інше узагальнення від класу В до класу А.

Після наведення вказівника миші на асоціацію і клацання правою кнопкою миші буде показано контекстне меню з пунктами дій, які ви можете застосувати до асоціації. Якщо вам потрібно вилучити асоціацію, оберіть у контекстному меню пункт **Вилучити**. Крім того, ви можете обрати пункт меню **Властивості**, а потім, залежно від типу асоціації, змінити такі атрибути, як ролі і численність.

#### 3.5.5.1 Точки якорів

Типово, асоціації малюються як прямі лінії, що пов'язують між собою два об'єкти на діаграмі.

Ви можете додати точки якорів, у яких асоціації вигинатимуться, подвійним клацанням лівою кнопкою миші у потрібній точці лінії асоціації. Така дія додасть точку якоря (якщо буде виділено асоціацію, точку якоря буде показано синьою точкою), яку можна буде пересунути, щоб надати асоціації потрібної форми.

Якщо вам потрібно буде вилучити точку якоря, просто наведіть на неї вказівник миші і ще раз двічі клацніть лівою кнопкою миші

Пам'ятайте, що єдиним способом, у який можна змінити властивості асоціації, є контекстне меню. Якщо ви спробуєте відкрити вікно властивостей подвійним клацанням, як це робиться для інших об'єктів UML, замість вікна на діаграму буде додано точку якоря.

### 3.5.6 Нотатки, текст і блоки

Нотатки, рядки тексту і блоки є елементами, які можна додати до діаграми будь-якого типу. Вони не мають семантичного значення, але дуже корисні, якщо вам потрібно додати коментар або пояснення, яке зробить вашу діаграму зрозумілішою.

Щоб додати нотатку або рядок тексту, оберіть відповідний інструмент на робочій панелі інструментів, а потім наведіть вказівник миші на місце діаграми, куди ви бажаєте додати ваш коментар, і один раз клацніть лівою кнопкою миші. Змінити текст елемента можна за допомогою властивостей елемента у контекстному меню або, у випадку нотаток, подвійного клацання лівою кнопкою миші на елементі.

### 3.5.6.1 Якорі

Якорі використовують для прив'язки текстової нотатки до іншого елемента UML. Наприклад, зазвичай текстову нотатку використовують для пояснення або коментування класу або окремої асоціації, у таких випадках ви можете скористатися якорем, щоб підкреслити, що нотатка «стосується» певного елемента.

Щоб додати якорі між нотаткою і іншим елементом UML, скористайтеся інструментом якоря з робочої панелі інструментів. Спочатку вам слід клацнути лівою кнопкою миші на нотатці, а потім на елементі UML, з яким ви бажаєте пов'язати нотатку.

## Розділ 4

# Імпортування та створення коду

Umbrello UML Modeller є інструментом для моделювання за допомогою UML, а отже його основним призначенням є допомогти вам у *аналізі і розробці* ваших систем. Крім того, для створення містка між вашими розробками і їх *реалізаціями* Umbrello UML Modeller надає вам змогу створювати початкові коди програм різними мовами програмування. Якщо ви бажаєте почати використання UML у вже створеному проєкті C++, Umbrello UML Modeller може допомогти вам створити модель вашої системи за початковими кодами шляхом аналізу вашого початкового коду та імпортування знайдених у ньому класів.

### 4.1 Створення коду

Umbrello UML Modeller може створювати початкові коди різними мовами програмування на основі вашої моделі UML, це допоможе вам розпочати реалізацію вашого проєкту. Створений код складатиметься з оголошень класів з їх методами та атрибутами, отже вам залишиться лише «заповнити порожні місця» реалізацією дій, передбачених у ваших класах.

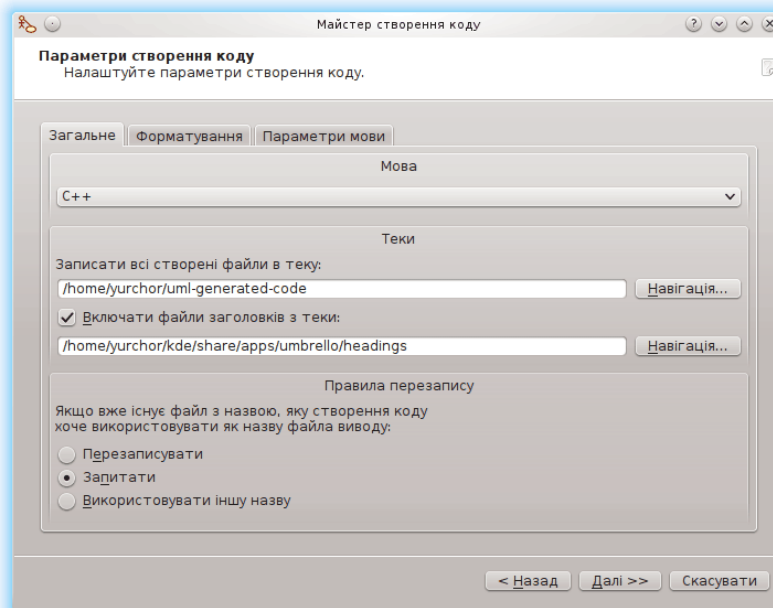
У версії Umbrello UML Modeller 2 реалізовано підтримку створення коду для мов програмування ActionScript, Ada, C++, C#, D, IDL, Java™, JavaScript, MySQL та Pascal.

#### 4.1.1 Як створити код

Для того, щоб створити за допомогою Umbrello UML Modeller початкові коди, вам спершу слід створити або завантажити модель, у якій міститиметься принаймні один клас. Коли ваша модель буде готова для створення коду, оберіть пункт **Майстер створення коду** з меню **Код**, щоб запустити майстер, за допомогою якого ви зможете пройти всіма кроками процесу створення коду.

Першим кроком буде вибір класів, для яких ви бажаєте створити початкові коди. Типово, буде обрано всі класи вашої моделі, ви зможете вилучити ті з них, для яких створювати код не потрібно, пересуванням відповідних записів до списку, розташованого у лівій частині вікна.

На наступному кроці майстер надасть вам змогу змінити параметри, які буде використано інструментом створення коду під час запису вашого коду. Можна змінити такі параметри:



Параметри створення коду у Umbrello UML Modeller

#### 4.1.1.1 Параметри створення

##### 4.1.1.1.1 Докладність коментарів

За допомогою параметра **Записувати коментарі документації**, навіть якщо елементи **порожні** можна наказати засобу створення коду записати коментарі у стилі `/** тру-ля-ля */`, навіть якщо блоки коментаря є порожніми. Якщо ви додавали документацію до ваших класів, методів або атрибутів вашої моделі, засіб створення коду запише ці коментарі у вигляді документації Doxygen безвідносно до того, чи позначите ви цей параметр, але, якщо ви позначите цей пункт Umbrello UML Modeller запише блоки коментарів для всіх класів, методів та атрибутів, навіть якщо у самій моделі документації не було, у останньому випадку вам слід буде виконати документування ваших класів безпосередньо у початкових кодах програми.

Позначення пункту **Записувати коментарі у розділи**, навіть якщо розділ порожній призведе до того, що Umbrello UML Modeller запише коментарі до коду програми для відокремлення різних розділів класу, наприклад «публічних (public) методів» або «Атрибутів» перед відповідними розділами. Якщо ви позначите цей пункт, Umbrello UML Modeller запише коментарі для всіх розділів класу, навіть якщо розділ є порожнім. Наприклад, програма запише коментар зі словами «protected methods» (захищені методи), навіть якщо у вашому класі немає захищених методів.

##### 4.1.1.1.2 Теки

**Записати всі створені файли в теку:** Тут вам слід обрати теку, до якої Umbrello UML Modeller має записати створені коди програми.

За допомогою пункту **Включати файли заголовків з теки**: ви можете вставити заголовки на початок кожного зі створених файлів. Файли заголовків можуть містити відомості про авторські права або ліцензування програми, а також змінні, які має бути обчислено під час створення. Щоб дізнатися, яким чином можна скористатися цими змінними для заміни вашого імені або поточної дати під час створення файлів, ви можете переглянути файли шаблонів заголовків, які постачаються разом з Umbrello UML Modeller.

#### 4.1.1.1.3 Правила перезапису

За допомогою цього пункту можна визначити поведінку Umbrello UML Modeller, якщо програма виявить, що файл, який вона бажає створити вже існує у теці призначення. Umbrello UML Modeller *не може змінювати вже існуючих файлів коду*, отже вам залишається обрати між перезаписом існуючого файла, пропуском створення цього окремого файла або наданням дозволу Umbrello UML Modeller на створення файла з іншою назвою, яку обере програма. Якщо ви використаєте варіант з іншою назвою, Umbrello UML Modeller додасть певний суфікс до назви файла.

#### 4.1.1.1.4 Мова

Типово, Umbrello UML Modeller створить код мовою програмування, яку ви обрали як активну мову, але у майстрі створення коду ви можете змінити цей вибір і обрати іншу мову.

#### 4.1.1.2 Створення за допомогою майстра створення

На третьому і останньому кроці майстер покаже вам стан процесу створення коду. Вам залишиться лише натиснути кнопку «Створити», щоб програма створила потрібні вам класи.

Пам'ятайте, що параметри, які ви оберете під час створення коду за допомогою майстра, будуть чинними лише під час поточного сеансу створення коду. Після наступного запуску майстра вам знову потрібно буде обирати значення цих параметрів (адресу вашої теки з заголовками, правила перезапису тощо). Встановити типові значення, які використовуватиме Umbrello UML Modeller можна у розділі **Створення коду** параметрів Umbrello UML Modeller, вікно яких можна відкрити за допомогою пункту меню **Параметри** → **Налаштувати Umbrello UML Modeller...**

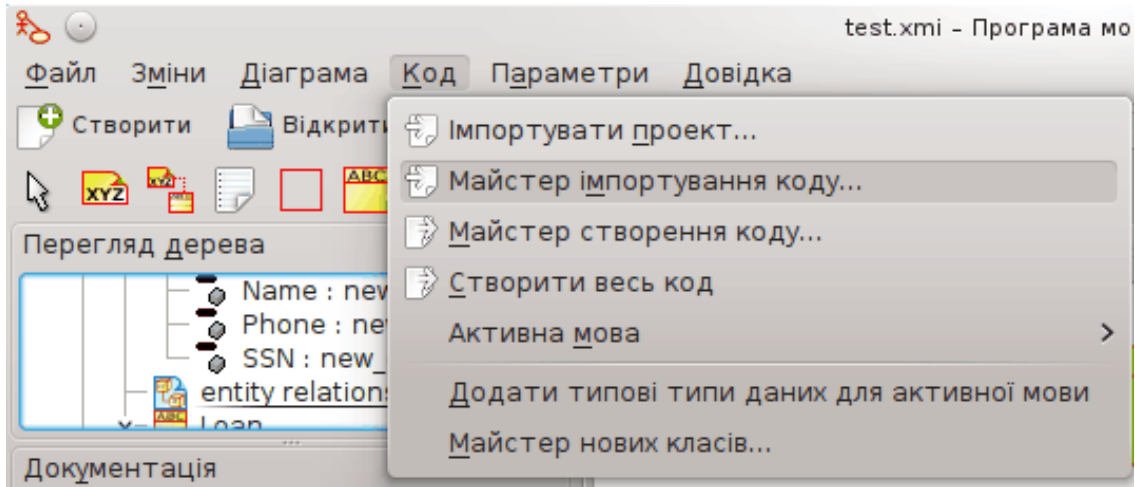
Якщо ви вказали належні значення параметрів створення коду і бажаєте одразу перейти до створення коду без використання майстра, ви можете обрати пункт **Створити весь код** з меню **Код**. Програма створить код для всіх класів у вашій моделі на основі поточних параметрів (зокрема значень теки виводу даних і правил перезапису, отже будьте обережні).

## 4.2 Імпортування коду

Umbrello UML Modeller може імпортувати код з вже створених вами проектів, щоб допомогти вам у побудові моделі ваших систем. У версії Umbrello UML Modeller 2 реалізовано підтримку коду мовами програмування ActionScript, Ada, C++, C#, D, IDL, Java™, Javascript, MySQL та Pascal.

Щоб імпортувати класи до моделі, скористайтеся пунктом **Майстер імпортування класів...** з меню **Код**. У діалоговому вікні вибору файлів оберіть файли, у яких містяться визначення класів C++, а потім натисніть кнопку **Далі >** та **Почати імпортування**. Вказані класи буде імпортовано, їхні записи стануть частиною вашої моделі, ви побачите їх у перегляді деревом. Пам'ятайте, що Umbrello UML Modeller не створюватиме ніякої діаграми для показу ваших класів, їх буде лише імпортовано для вашої моделі, а пізніше ви зможете скористатися ними для будь-якої бажаної діаграми.





Меню для імпортування вихідних кодів у Umbrello UML Modeller

## Розділ 5

# Інші можливості

### 5.1 Інші можливості Umbrello UML Modeller

У цій главі буде коротко розглянуто деякі інші можливості, які пропонує вам Umbrello UML Modeller.

#### 5.1.1 Копіювання об'єктів у вигляді зображень PNG

Окрім звичайних функціональних можливостей копіювання, вирізання і вставки, працездатності яких ви можете очікувати для об'єктів різних діаграм, Umbrello UML Modeller може копіювати об'єкти у вигляді зображень PNG так, щоб ви могли вставити ці зображення до будь-якого іншого типу документа. Для того, щоб скористатися цією можливістю, вам не потрібно виконувати ніяких додаткових налаштувань, просто оберіть об'єкт на діаграмі (Клас, Об'єкт дії тощо) і скопіюйте його (за допомогою комбінації клавіш **Ctrl-C** або пункту меню), а потім відкрийте документ Calligra Words (або будь-якої іншої програми, у якій існує можливість вставки зображень) і оберіть пункт меню **Вставити**. Це чудова можливість для експорту частин вашої діаграми у вигляді простих зображень.

#### 5.1.2 Експортування до зображення

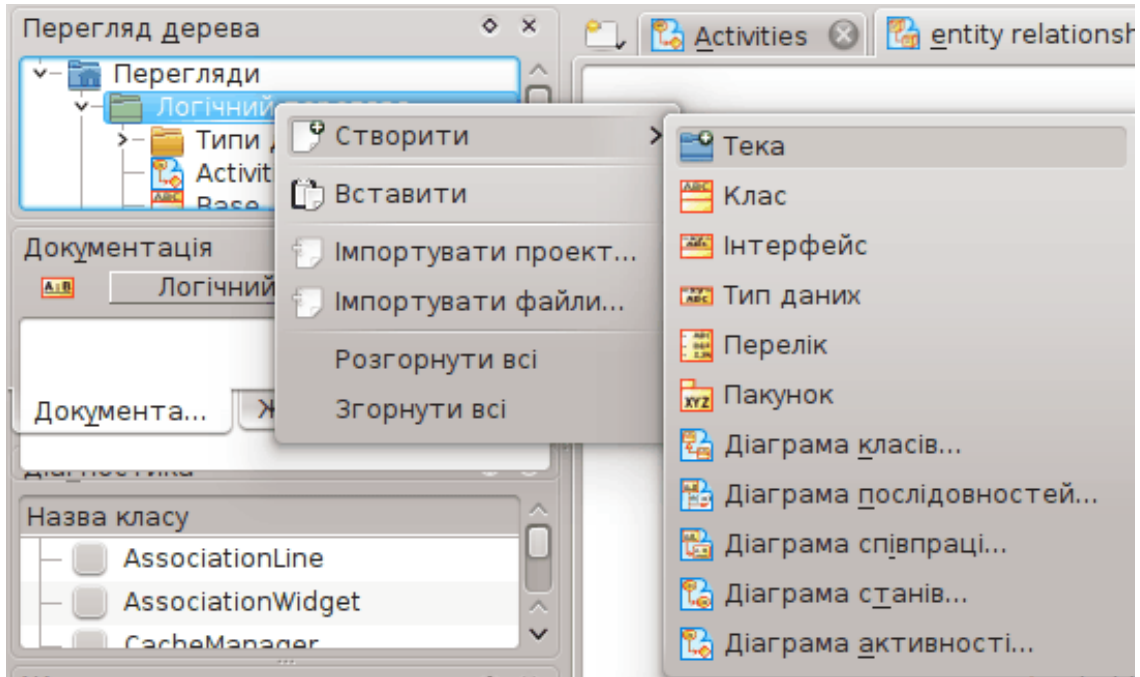
Крім того, ви можете експортувати як зображення цілу діаграму. Єдина річ, яку вам буде потрібно зробити, це обрати діаграму, яку ви бажаєте експортувати, а потім скористатися пунктом **Експортувати як зображення...** з меню **Діаграма**.

#### 5.1.3 Друк

Umbrello UML Modeller надає вам змогу друкувати окремі діаграми. Натисніть кнопку **Друкувати** на панелі інструментів програми або скористайтеся пунктом **Друкувати** з меню **Файл**. У відповідь програма відкриє стандартне діалогове вікно налаштування друку KDE, за його допомогою ви зможете надрукувати вашу діаграму.

#### 5.1.4 Логічні теки

Для покращення впорядкування вашої моделі, особливо, якщо це модель великого проекту, ви можете створити логічні теки на панелі перегляду деревом. Просто скористайтеся пунктом контекстного меню **Створити** → **Тека** типових тек на панелі перегляду деревом, щоб створити ці логічні теки. Ви також можете створювати підтеки і пересувати між ними об'єкти перетягуванням позначки об'єкта з однієї теки з наступним скиданням її до іншої теки.



*Впорядкування моделі за логічними теками у Umbrello UML Modeller*

## Розділ 6

# Автори та історія програми

Цей проект було розпочато Паулем Генсеном у рамках одного з його університетських проєктів. Першою назвою програми була назва UML Modeller. Розробка всіх частин програми здійснювалася Паулем до кінця 2001 року, коли було випущено версію 1.0 програми.

Версія 1.0 вже була достатньо функціональною, але після розгляду проєкту у університеті Пауля, до проєкту змогли долучитися інші розробники, які зробили значний внесок до UML Modeller, зокрема, перехід з бінарного формату файлів на файли XML, підтримку більшої кількості типів діаграм UML, створення коду та імпорт коду та багато інших удосконалень.

Пауль пішов з команди розробників влітку 2002 року, але оскільки програма є вільним і відкритим програмним забезпеченням, розробку було продовжено, програма вдосконалювалася і розвивалася, її підтримувала група розробників з різних частин світу. У вересні 2002 року назву проєкту було змінено з UML Modeller на Umbrello UML Modeller. Для зміни назви було декілька причин, найважливішою з них було те, що «uml» — у тому сенсі, у якому він відомий багатьом — був занадто загальним для назви, і це спричинило проблеми з додаванням програми до деяких дистрибутивів. Іншою важливою причиною було те, що розробникам здалося, що Umbrello є набагато кращою назвою.

Дискусія з приводу розробки Umbrello UML Modeller, а також обговорення напрямків, у яких програм розвиватиметься у наступних версіях, відкрита для будь-яких думок. Це обговорення триває у мережі Інтернет. Якщо ви маєте намір зробити внесок до проєкту, будь ласка, не вагайтеся і зв'яжіться з розробниками. Існує багато способів, у які ви можете допомогти розвитку Umbrello UML Modeller:

- Повідомляти про вади або пропонувати покращення
- Виправляти вади і додавати можливості
- Написати гарну документацію або переклад іншими мовами.
- Ну, і звичайно ж,... писати програму разом з нами!

Як ви бачите, існує декілька шляхів зробити свій внесок до програми. Всі ці шляхи є дуже важливими, автори запрошують всіх взяти участь у покращенні програми.

Зв'язатися з розробниками Umbrello UML Modeller можна за допомогою адреси електронної пошти [umbrello-devel@kde.org](mailto:umbrello-devel@kde.org).

## Розділ 7

# Авторські права

Авторські права на програму належать Paul Hensgen, ©2001

Авторські права на програму належать авторам Umbrello UML Modeller, 2002–2014.

Цей документ поширюється за умов дотримання [GNU Free Documentation License](#).

Ця програма поширюється за умов дотримання [GNU General Public License](#).