



{**PROGRAMMING**}



**UNITED
MODELING
LANGUAGE**

Урок №2

Діаграми в UML,
діаграми варіантів
використання

Зміст

1. Концепція діаграми в UML.....	3
2. Короткий огляд існуючих видів діаграм	4
2.1. Діаграма варіантів використання (use case diagram)	4
2.2. Діаграма класів (class diagram)	5
2.3. Діаграма станів (statechart diagram).....	5
2.4. Діаграма активності (activity diagram)	6
2.5. Діаграма послідовності (sequence diagram).....	7
2.6. Діаграма кооперації (collaboration diagram)	7
2.7. Діаграма компонентів (component diagram).....	8
2.8. Діаграма розгорткування (deployment diagram)...	8
3. Інструменти для створення діаграм.....	10

1. Концепція діаграми в UML

Отже, після першого уроку ви володієте певним обсягом інформації про мову UML. Продовжимо наше знайомство та розберемо поняття діаграми. Що таке діаграма? Чи знайоме вам це поняття? «Так, звичайно!», — вигукнете ви. Напевно, ви зможете навести приклади діаграм з різних галузей знань (математики, економіки, фізики, хімії...). Наприклад, діаграми Фейнмана, діаграма стану і багато інших. Той чи інший вид діаграми використовується для демонстрації деякого факту, взаємодії між об'єктами, принципів роботи. Які цілі ставляться перед діаграмами в UML? Діаграми в мові UML використовуються для відображення зв'язків, станів, сценаріїв використання та поведінки додатків, і т. д. Тобто, фактично, мета набору діаграм — описати деякий додаток (комплекс програм) з різних сторін. У цьому уроці ми проведемо огляд типів UML діаграм, проаналізуємо засоби для побудови UML діаграм.

2. Короткий огляд існуючих видів діаграм

Метою даного розділу є короткий огляд існуючих видів UML діаграм. Важливо відразу розуміти, що у кожного виду діаграм є своя особлива мета, заради якої цей вид діаграми і був розроблений.

2.1. Діаграма варіантів використання (use case diagram)

Діаграма варіантів використання (діаграма прецедентів) — це вид діаграми, призначений для того, щоб дати можливість обговорювати, аналізувати поведінку і функціональність програми (програмного проекту, системи) різним зацікавленим особам (замовник, кінцевий користувач, програміст, тестувальник).

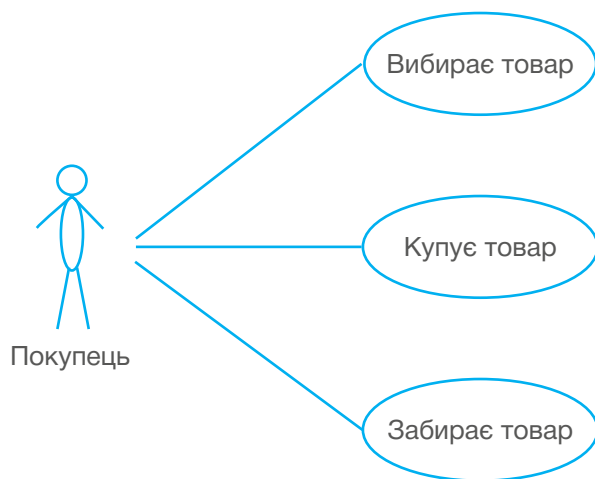


Рисунок 1.

Цей вид діаграми дуже простий і зрозумілий будь-якій людині через застосування найпростіших позначок. Фактично, цей вид діаграм відображає сценарій використання програми (*use case*) у звичних нам термінах. Наведемо приклад типової діаграми (рис. 1).

2.2. Діаграма класів (class diagram)

Діаграма класів — діаграма, яка визначає структуру програми (програмного проєкту, системи). Мета такої діаграми показати класи, їх властивості (атрибути, методи), зв'язок між класами, інтерфейси тощо. Наведемо приклад типової діаграми (рис. 2).

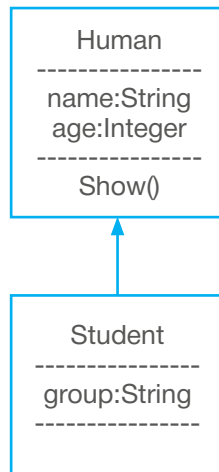


Рисунок 2.

2.3. Діаграма станів (statechart diagram)

Що таке стан? Напевно, ви знаєте відповідь на таке очевидне питання. Наприклад: «Як твій стан?» — це одне з найпоширеніших питань, яке ставлять хворій людині.

Природно, що стан, зазвичай, не константна величина, і він змінюється залежно від різних чинників. У випадку хвороби стан може покращитись після прийому ліків. В UML діаграма станів показує, як об'єкт переходить із одного стану до іншого. Причому, відображається процес зміни станів лише конкретного об'єкта деякого класу внаслідок реакції на подію. Наведемо приклад типової діаграми (рис. 3).

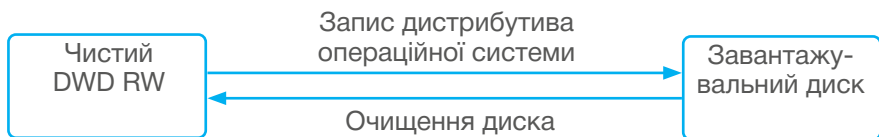


Рисунок 3.

2.4. Діаграма активності (activity diagram)

В рамках курсу «Програмування на мові C» ви познайомилися з поняттям блок-схеми. Основна мета блок-схеми полягає у наочному зображенні алгоритму розв'язання певної задачі.

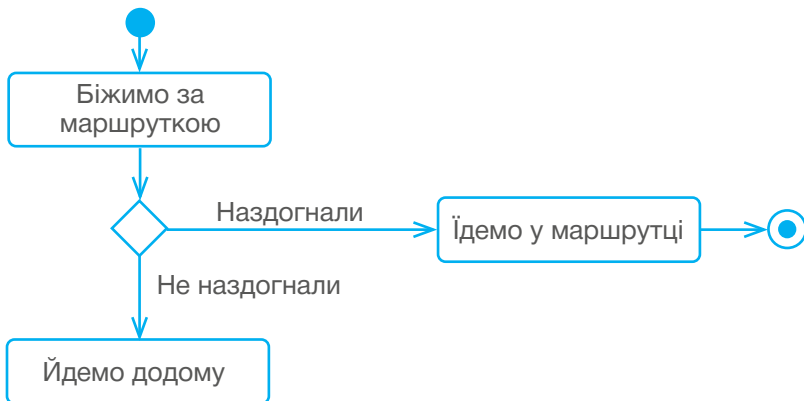


Рисунок 4.

Діаграма активності дуже подібна до вже знайомих вам блок-схем. Цей вид діаграм є окремим випадком діаграм стану. Як і блок-схеми, вони застосовуються для візуалізації роботи алгоритму, яким запрограмований певний клас. Наведемо приклад типової діаграми (рис. 4).

2.5. Діаграма послідовності (sequence diagram)

Цей вид діаграм використовується для відображення упорядкованої в часі взаємодії об'єктів. На ній відображаються взаємодіючі об'єкти, а також послідовність повідомлень, якими обмінюються. Наведемо приклад типової діаграми (рис. 5).

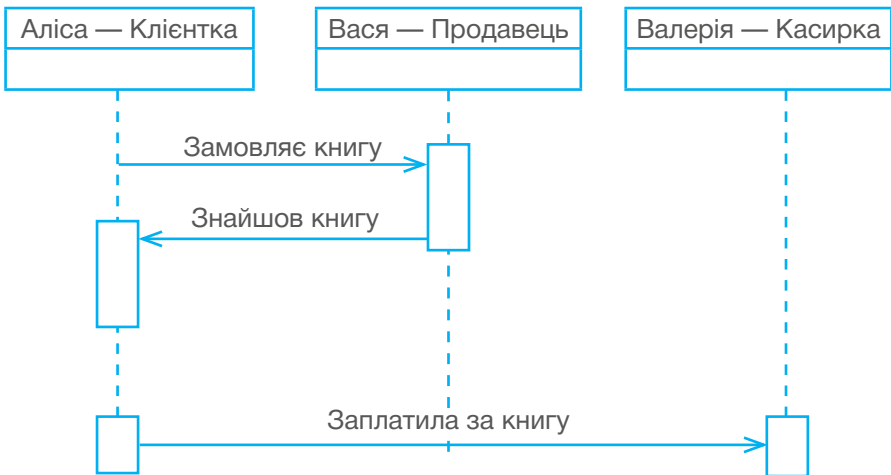


Рисунок 5.

2.6. Діаграма кооперації (collaboration diagram)

Діаграма кооперації (діаграма взаємодії) — відображає потік повідомлень між об'єктами, що становлять систему, а також демонструє зв'язки між ними. Ця діаграма є

аналогією до діаграми послідовності. Наведемо приклад типової діаграми (рис. 6).

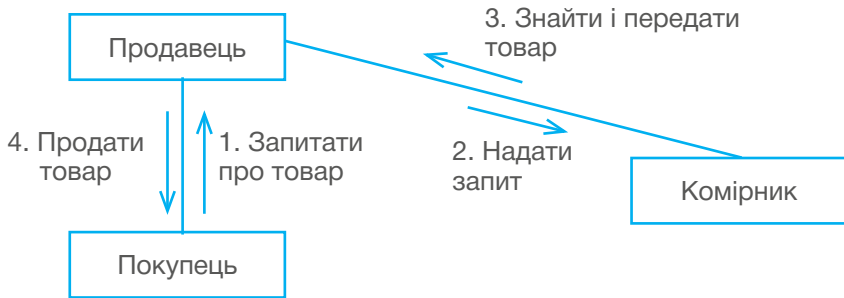


Рисунок 6.

2.7. Діаграма компонентів (component diagram)

За допомогою цього виду діаграм відображають розбиття програмного проєкту (системи) на структурні компоненти і зв'язки між ними. У якості компонентів можуть бути: бібліотеки, файли, модулі і т. д. Наведемо приклад типової діаграми (рис. 7).

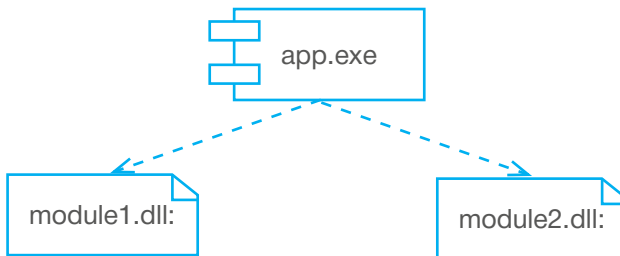


Рисунок 7.

2.8. Діаграма розгортання (deployment diagram)

З назви цієї діаграми вже зрозуміло навіщо вона потрібна. Коли говорять про розгортання, зазвичай

описують, що потрібно виконати для того, щоб додаток програми міг працювати. Діаграма розгорткування відображає конфігурацію вузлів, що проводять обробку інформації, та опис компонентів, розміщених у вузлах.

Наведемо приклад типової діаграми (рис. 8).

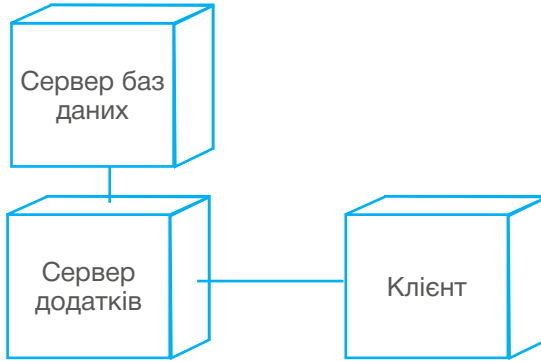


Рисунок 8.

Це був лише короткий огляд діаграм. З кожною із них ви ознайомитесь більш детально у наступних уроках.

3. Інструменти для створення діаграм

Для створення діаграм можна скористатися різними пакетами. Серед них є як платні, так і безкоштовні рішення. Наприклад, компанія Microsoft надає чудовий продукт Visio, який можна використовувати для розробки діаграм різних типів, зокрема діаграм UML.

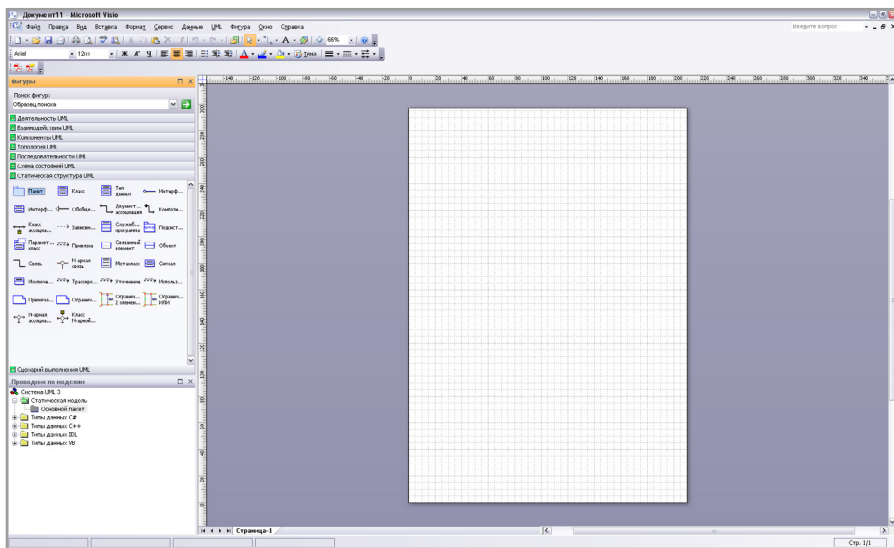


Рисунок 9.

Існує також велика кількість OpenSource проєктів, які можуть полегшити проєктування. Наприклад, такі проєкти як Ar-goUML, StarUML.

Також варто згадати про культовий засіб розробки Rational Rose. Цей продукт шанується серед розробників.

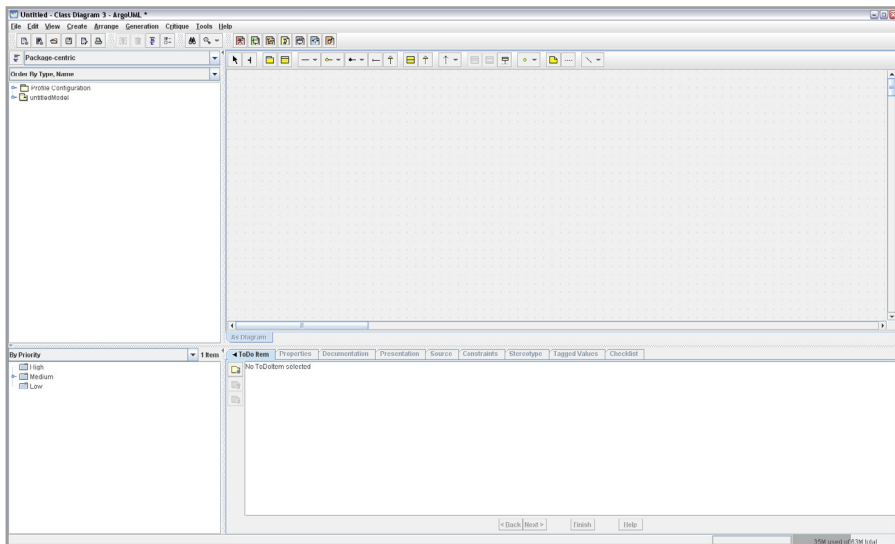


Рисунок 10.



Урок №2

Діаграми в UML, діаграми варіантів використання

© STEP IT Academy, www.itstep.org

Усі права на фото-, аудіо- і відеотвори, що охороняються авторським правом і фрагменти яких використані в матеріалі, належать їх законним власникам. Фрагменти творів використовуються в ілюстративних цілях в обсязі, виправданому поставленим завданням, у рамках учбового процесу і в учбових цілях, відповідно до законодавства про вільне використання твору без згоди його автора (або іншої особи, яка має авторське право на цей твір). Обсяг і спосіб цитованих творів відповідає прийнятим нормам, не завдає збитку нормальному використанню об'єктів авторського права і не обмежує законні інтереси автора і правовласників. Цитовані фрагменти творів на момент використання не можуть бути замінені альтернативними аналогами, що не охороняються авторським правом, і відповідають критеріям добросовісного використання і чесного використання.

Усі права захищені. Повне або часткове копіювання матеріалів заборонене. Узгодження використання творів або їх фрагментів здійснюється з авторами і правовласниками. Погоджене використання матеріалів можливе тільки якщо вказано джерело.

Відповідальність за несанкціоноване копіювання і комерційне використання матеріалів визначається чинним законодавством.