

<sup>1</sup>Шаров С.В., канд. пед. наук, доцент; <sup>2</sup>Шамардак О.А.

Мелітопольський державний педагогічний університет імені Богдана Хмельницького,

м. Мелітополь

Кафедра інформатики і кібернетики, <sup>1</sup>доцент, <sup>2</sup>студентка 4 курсу

## КОНЦЕПТУАЛЬНЕ МОДЕЛЮВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ЗА ДОПОМОГОЮ МОВИ UML

**Актуальність.** На сучасному етапі розвитку інформаційних технологій широкого застосування набули інформаційні системи, призначені для задоволення інформаційних потреб кінцевих користувачів. Процесу безпосередньої розробки інформаційної системи повинен передувати процес проектування окремих модулів системи та їх поведінки. Одним із засобів моделювання складу та функціональності інформаційної системи є уніфікована мова моделювання UML, зокрема діаграма класів.

**Метою статті** є огляд основних понять та загальна характеристика уніфікованої мови моделювання UML в аспекті побудови концептуальної моделі інформаційної системи за допомогою діаграми класів.

**Виклад основного матеріалу.** Як відомо, розробці потужного програмного засобу за допомогою конкретного інструментального середовища повинен передувати процес проектування його структури та функціональності. При цьому повинен забезпечуватися високий рівень і докладний опис логіки додатку, що задовольняє системним вимогам та можливим обмеженням. Крім того, у процесі створення та функціонування інформаційної системи потреби користувачів можуть змінюватися або уточнюватися, що призводить до більшого ускладнення розробки і супроводу таких систем.

Для уникнення або вирішення зазначених проблем рекомендується будувати певну модель, у якій будуть відбиті структурні та функціональні особливості майбутньої інформаційної системи. Взагалі, під моделлю розуміється об'єкт (або опис об'єкта) для заміщення однієї системи (оригіналу)

іншою системою для вивчення оригіналу або відтворення його будь-яких властивостей. Процес моделювання пов'язаний з розробкою систем комп'ютерного моделювання, які підтримують весь життєвий цикл моделі. Особливість комп'ютерних систем моделювання полягає в їх високій інтеграції та інтерактивності [4, с. 61].

Об'єктивні труднощі, що пов'язані із формальним описом предметної області та побудовою повних, функціонально-несуперечливих інформаційних моделей, сприяли появі програмно-технологічних засобів (CASE-засобів), використання яких сприятиме підвищенню ефективності розробки різноманітного програмного забезпечення. При цьому CASE-технологія визначається як сукупність методів проектування програмного забезпечення та набір інструментальних засобів, які дозволяють у наочній формі моделювати наочну область та аналізувати цю модель на всіх стадіях розробки та супроводження програмного забезпечення [3, с. 118].

Більшість існуючих CASE-засобів засновано на методологіях структурного та об'єктно-орієнтованого підходів. Слід зазначити, що для моделювання потужних програмних засобів, зокрема інформаційних систем, які використовують бази даних, рекомендується використовувати об'єктно-орієнтований підхід. Основна ідея об'єктно-орієнтованого аналізу і проектування полягає у розгляді предметної області та логічного вирішення завдань з точки зору об'єктів (понять або сутностей). У процесі об'єктно-орієнтованого аналізу основна увага приділяється визначенню та опису об'єктів в термінах предметної області. У процесі об'єктно-орієнтованого проектування визначаються логічні програмні об'єкти (атрибути та методи), які будуть реалізовані засобами певної об'єктно-орієнтованої мови програмування [2].

Об'єктно-орієнтовані методи мають наступні переваги: можливість збірки програмної системи з готових компонентів; багаторазове використання розроблених компонентів; можливість накопичення проектних рішень у вигляді бібліотек класів; простота внесення змін у проекти за рахунок інкапсуляції даних в об'єктах; швидка адаптація додатків до умов за рахунок використання

властивостей спадкоємства та поліморфізму; можливість організації паралельної роботи проєктувальників і програмістів [6, с. 43].

Універсальним засобом об'єктно-орієнтованого моделювання є уніфікована мова UML (Unified Modeling Language), розробка якої розпочалася з середини 90-х років минулого століття на базі декількох об'єктно-орієнтованих методів і нотацій опису інформаційних систем. Зараз мова UML є загальноприйнятим стандартом документування процесу створення інформаційних систем і програмного забезпечення. На сьогодні існує версія UML 2.0. До пакетів, які працюють з UML, можна віднести IBM Rational Rose, Telelogic TAU G2, Microsoft Visio, Enterprise Architect, Modelio та ін.

У межах мови UML усі уявлення про модель складної системи фіксуються у вигляді спеціальних графічних конструкцій, що одержали назву діаграм. У даному випадку під діаграмою (diagram) розуміється графічне представлення сукупності елементів моделі у формі зв'язного графа, вершинам і ребрам (дугам) якого приписується визначена семантика [7, с. 279].

UML 1.5 визначає дванадцять типів діаграм, розділених на три групи: чотири типи діаграм представляють статичну структуру додатка (діаграми класів, діаграми об'єктів, діаграми компонентів, діаграми розгортання); п'ять представляють поведінкові аспекти (діаграми прецедентів; діаграми послідовності; діаграми кооперації; діаграми станів; діаграми діяльності); три представляють фізичні аспекти функціонування (діаграми реалізації) [1].

Декілька слів слід сказати про поняття нотації, що використовується у всіх у різних діаграмах. Нотація – сукупність символів і правил їх застосування, що використовуються для представлення понять та зв'язків між ними [7, с. 279]. Слово «нотація» підкреслює, що UML – мова графічна, де моделі (діаграми) не записують, а малюють. В UML використовується чотири види елементів нотації: фігури (для позначення вузлів інфраструктури), лінії (для зв'язку між вузлами), значки та написи.

При проєктуванні об'єктно-орієнтованих систем діаграми класів є обов'язковими. Діаграма класів – це набір статичних, декларативних елементів моделі. Слід зазначити, що діаграми класів можуть застосовуватися і при

прямому (в процесі розробки нової системи), і при зворотному проектуванні – описі існуючих інформаційних систем [1].

При побудові діаграми класу використовуються наступні об'єкти: клас (має ім'я, мову програмування для трансляції програмного коду), атрибут (властивість класу, має ім'я, тип даних, може мати значення видимості), операція (поведінка класу, має ім'я та параметри) зв'язок між класами (має кратність, клас-source, клас-target). Дано визначення та коротко охарактеризуємо вищезазначені поняття.

Під класом у мові UML розуміється іменований опис сукупності об'єктів, що володіють однаковою структурою, семантикою, властивостями та відносинами з іншими об'єктами. Графічно клас зображується у вигляді прямокутника.

Атрибутом класу називається іменована властивість класу, що описує безліч значень, які можуть приймати екземпляри цієї властивості. Клас може мати будь-яке число атрибутів або не мати жодного. Імена атрибутів розташовуються у розділі класу, розташованому під ім'ям класу.

Операцією класу називається іменована абстракція того, що можна робити з об'єктом (класом). Клас може містити будь-яке число операцій або не містити жодної. Набір операцій є загальним для всіх об'єктів даного класу. Операції класу визначаються в розділі, розташованому нижче розділу з атрибутами [5].

У діаграмі класів можуть брати участь зв'язки трьох різних категорій: залежність (dependency), узагальнення (generalization) та асоціація (association). Зупинимося лише на третьому типі зв'язку. Асоціацією називається структурний зв'язок, який показує, що об'єкти одного класу певним чином пов'язані з об'єктами іншого або того ж самого класу.

Слід додати, що іноді в діаграмі класів потрібно відобразити той факт, що асоціація між двома класами має спеціальний вид «частина-ціле». Асоціація такого роду називається агрегатною. Графічно агрегатні асоціації зображуються у вигляді простої асоціації з незафарбованим ромбом на стороні класу-«цілого». Існують випадки, коли знищення «цілого» призводить до знищення всіх його

«частин». В такому випадку агрегатні асоціації, що володіють такою властивістю, називаються композитними. Графічно композиція зображується у вигляді простої асоціації, доповненої зафарбованим ромбом з боку «цілого».

З поняттям асоціації пов'язано поняття кратності ролі (multiplicity). Це характеристика, яка вказує, скільки об'єктів класу з даною роллю можуть брати участь у кожному примірнику асоціації. Найбільш поширеним способом завдання кратності ролі асоціації є вказівка конкретного числа або діапазону.

**Висновки.** Отже, уніфікована мова моделювання UML призначена для моделювання інформаційних систем та різноманітного програмного забезпечення. Діаграма класів, яка є однією з частин повної моделі інформаційної системи, дозволяє спроектувати об'єкти наочної області, їх характеристики та поведінку, а також зв'язки між даними об'єктами.

#### Література

1. Бабич А. Види діаграм UML: [Електроний ресурс] / А. Бабич. – Режим доступу: <http://www.intuit.ru/studies/courses/1007/229/lecture/5954>.
2. Белова А.А. Объектно-ориентированное моделирование: [Електроний ресурс] / А.А. Белова. – Режим доступу: <http://www3.msiu.ru/~belova/compmo/obj.pdf>.
3. Гайна Г.А. Основы проектирования баз данных: навч. посіб. / Г.А. Гайна. – К.: КНУБА, 2005. – 204 с.
4. Казиев В.М. Информационно-логическое и математическое моделирование самоорганизующихся социально-экономических систем // В.М. Казиев, К.В. Казиев. – Нальчик: Каб.-Балк. ун-т, 2003. – 232 с.
5. Кузнецов С. Проектирование реляционных баз данных с использованием семантических моделей: диаграммы классов языка UML [Електроний ресурс] / С. Кузнецов. – Режим доступу: <http://www.intuit.ru/studies/courses/74/74/lecture/2234>.
6. Павленко Л.А. Корпоративні інформаційні системи: Навч. посіб. / Л.А. Павленко. – Харків: ВД «ІНЖЕК», 2003. – 260 с.
7. Томашевський О.М. Інформаційні технології та моделювання бізнес-процесів: навч. посіб. / О.М. Томашевський, Г.Г. Цегелик, М.Б. Вітер, В.І. Дудук. – К.: Видавництво «Центр учбової літератури», 2012. – 296 с.