

ДОДАТОК А  
Графічний матеріал атестаційної роботи

Харківський національний університет радіоелектроніки  
Кафедра ЕОМ

# Методи проектування програмного та інформаційного забезпечення об'єктно-орієнтованих баз даних

Атестаційна робота  
Другий (магістерський) рівень

Автор:  
Гонтарева Д. В.,  
студ. гр. СПм-19-1

Керівник:  
Філімончук Т. В.,  
доц. каф. ЕОМ



## МЕТА АТЕСТАЦІЙНОЇ РОБОТИ І ПОСТАНОВКА ЗАДАЧІ

Метою атестаційної роботи є дослідження методів проектування та інформаційного забезпечення об'єктно-орієнтованих баз даних, з'ясування того, як застосування об'єктно-орієнтованої бази даних вплине на процес розробки програмного забезпечення, які переваги та недоліки будуть виявлені.

У ході атестаційної роботи необхідно виконати наступні задачі:

- визначити основні принципи об'єктно-орієнтованої парадигми;
- розглянути популярні об'єктно-орієнтовані системи управління базами даних;
- розглянути основні етапи проектування об'єктно-орієнтованої бази даних;
- розробити модель проектування програмного та інформаційного забезпечення ООБД;
- виконати проектування фізичної моделі об'єктно-орієнтованої бази даних;
- порівняти проблеми, які виникають в реляційних та об'єктно-орієнтованих базах даних.

## ОСНОВНІ ПРИНЦИПИ ОБ'ЄКТНО-ОРІЄНТОВАНОГО ПРОГРАМУВАННЯ

Парадигма програмування – це сукупність ідей та понять, що визначають стиль написання комп'ютерних програм.

Об'єктно-орієнтоване програмування – одна з парадигм програмування, яка розглядає програму як множину «об'єктів», що взаємодіють між собою.



3

## ОБ'ЄКТНО-ОРІЄНТОВАНІ СИСТЕМИ УПРАВЛІННЯ БАЗАМИ ДАНИХ

СУБД – комплекс програм, що дозволяють створити базу даних і маніпулювати даними (вставляти, оновлювати, видаляти та обирати). Система забезпечує безпеку, надійність зберігання і цілісність даних, а також надає засоби для адміністрування БД.

Проект ORION здійснювався фірмою МСС під керівництвом В. Кіма. Основними функціональними компонентами системи є:

- підсистеми управління пам'яттю;
- підсистеми управління об'єктами;
- підсистеми управління транзакціями.

Проект O2 розроблявся французькою компанією Altair, яка була утворена спеціально для цілей проектування і реалізації об'єктно-орієнтованої СУБД. Основними компонентами системи є:

- інтерпретатор запитів;
- підсистема управління схемою;
- підсистема управління об'єктами;
- підсистема управління дисками.

4

## ЕТАПИ ПРОЕКТУВАННЯ ОБ'ЄКТНО-ОРІЄНТОВАНОЇ БАЗИ ДАНИХ

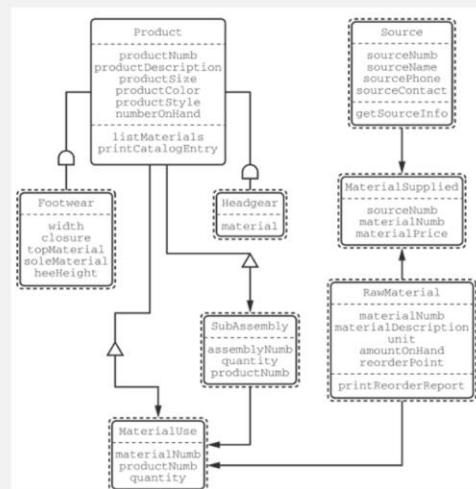
Проектування об'єктно-орієнтованої бази даних можна розділити на чотири етапи:

- аналіз предметної області (починаючи проектування ООБД, в першу чергу потрібно вивчити предметну область, в якій буде виконуватися робота БД, врахувати основні процеси, вимоги та побажання замовника та майбутніх користувачів);
- створення концептуальної моделі (конструювання моделі, що не залежить від будь-яких фізичних особливостей реалізації);
- логічне проектування (конструювання інформаційної моделі на основі конкретно виділених, існуючих типів даних, але без урахування специфіки СУБД та інших фізичних особливостей реалізації.);
- створення фізичної моделі.

5

## НОТАЦІЯ КООДА-ЙОРДАНА

- клас відображається заокругленим прямокутником;
- прямокутники класів, з яких створюються об'єкти, обводяться пунктиром. У прямокутника, що представляє абстрактний клас, є лише одна суцільна рамка;
- ім'я класу пишеться у верхній частині прямокутника, який його позначає;
- в середній частині прямокутника розташовуються атрибути, які визначені в класі;
- в нижній частині прямокутника розташовуються методи, які встановлені в класі;
- символ  $\square$  представляє зв'язок типу «є» (клас Footwear є виробом, так само як і клас Headgear);
- символ  $\triangle$  відповідає зв'язку «ціле частина» (клас SubAssembly – це частина виробу);
- лінія без стрілки позначає наявність одного примірника або відсутність жодного;
- лінія зі стрілкою позначає наявність одного та більше примірників.

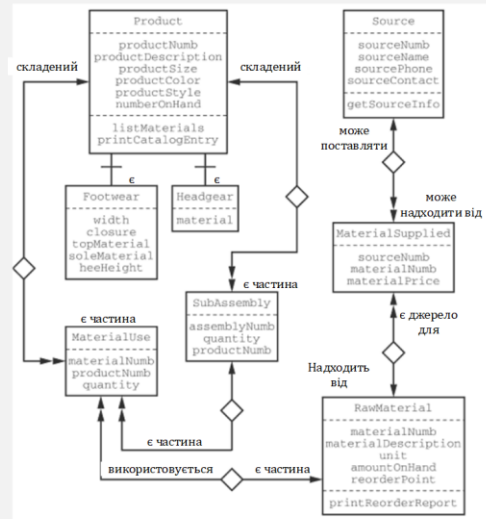


6

## НОТАЦІЯ ШЛЕЄРА-МЕЛЛОРА

- класи в нотатції зображуються прямокутниками, що містять такі ж секції для імені, атрибутів та методів, що і в нотатції Коада-Йордана;
- зв'язок типу «є» представлено вертикальною лінією, яка перекреслена горизонтальним штрихом;
- зв'язки в базі даних зображені ромбиками, з яких виходять лінії зі стрілками;
- одиначна стрілка означає нуль або один екземпляр, а подвійна стрілка – один або більше. З кожним зв'язком асоційована мітка, яка пояснює її семантику.

Нотатція Шлеєра-Меллора дає більше інформації про зв'язки в базі даних, ніж нотатція Коада-Йордана. Однак в ній немає явної підтримки зв'язку «ціле частина». До того ж вона теж не дозволяє розрізнити обов'язкові та факультативні зв'язки і не має простих засобів для подання агрегатів.



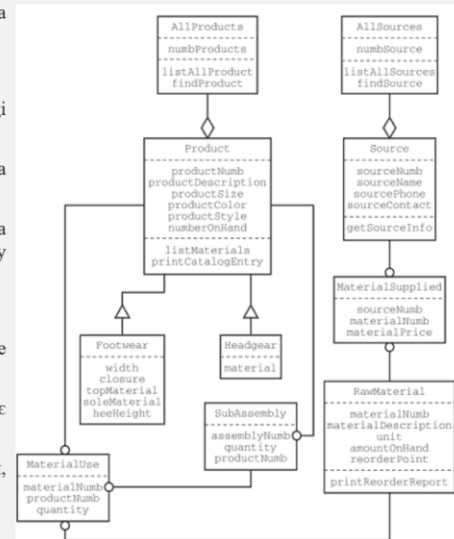
7

## НОТАЦІЯ РАМБО (OMT)

Класи позначаються прямокутниками, в які вписується ім'я, атрибути та методи класу.

Види зв'язків (асоціації) позначаються символами на кінцях стрілок:

- лінія без стрілок представляє «рівно один» та використовується на кінці «один» обов'язкового зв'язку «один до багатьох»;
- лінія з незафарбованим колом на кінці означає «нуль або один». Вона застосовується на кінці «один» необов'язкового зв'язку «один до багатьох»;
- лінією з зафарбованим колом на кінці зображується «нуль або більше». Вона вживається на кінці «багатьох» зв'язку «один до багатьох» або на будь-якому кінці зв'язку «багато до багатьох»;
- лінія з незафарбованим ромбом на кінці показує обов'язкове агрегування;
- лінія з незафарбованими ромбом та колом на кінці означає необов'язкове агрегування;
- лінія з незафарбованим ромбом та зафарбованим колом на кінці показує можливе агрегування;
- якщо кратність зв'язку фіксована та заздалегідь відома, то її можна вказати, помістивши над лінією ціле число.



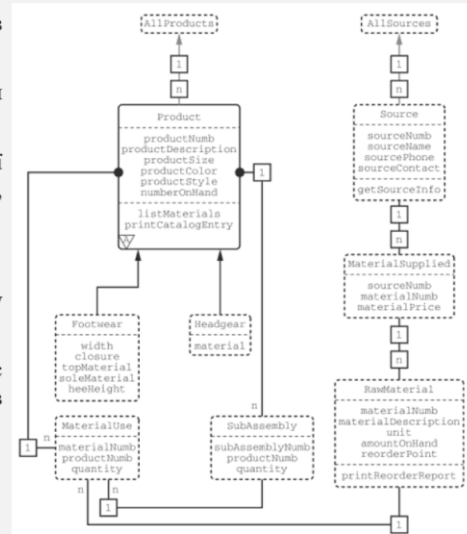
8

## НОТАЦІЯ БУЧА

Класи відображаються закругленими прямокутниками з трьома звичайними секціями: ім'я, атрибути та методи.

Асоціації між класами маркуються різноманітними позначеннями на лініях, що з'єднують їх:

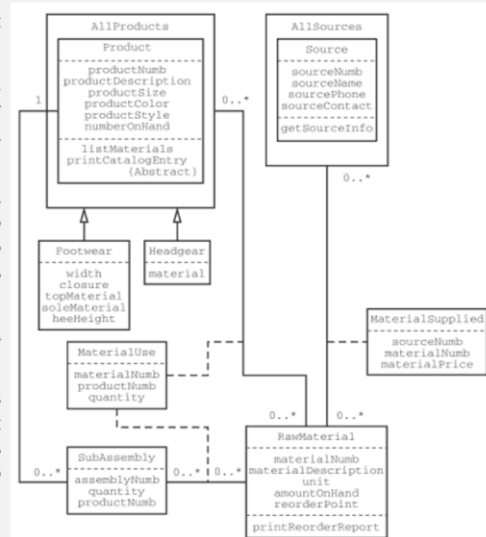
- лінія без символу на кінці позначає асоціацію, кратність якої задається числами. На рисунку всі асоціації мають кратність 1:n, тобто є обов'язковими зв'язками «один до багатьох»;
- лінія з одиночної стрілкою на кінці зображує наслідування;
- лінія з зафарбованим колом на кінці показує асоціацію типу «є»;
- біла лінія з однією стрілкою позначає метаклас, тобто клас класів. Такий зв'язок застосовується для асоціювання класів Product і Source з класами, які їх агрегують.



9

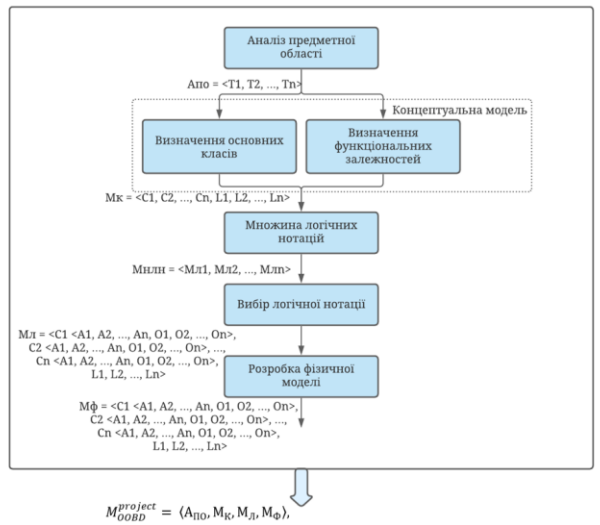
## НОТАЦІЯ UML (UNIFIED MODELING LANGUAGE)

- регулярний клас відображається прямокутником, розділеним на три секції: ім'я, атрибути, методи;
- клас-агрегат зображено прямокутником, в який записано його ім'я і прямокутник класу, об'єкти якого агрегуються. Наприклад, на рисунку класи Product та Source знаходяться відповідно всередині агрегуючих класів AllProducts і AllSources;
- асоціації зображуються лініями, на кінцях яких немає ніяких символів. При цьому кратність один записується просто як «1». Якщо кратність може бути нуль або один, то це виглядає як «0..1». Якщо кратність дорівнює нулю або більше, то пишеться «0..\*», а кратність один або більше позначається як «1..\*»;
- наслідування представлено лінією з незафарбованою стрілкою, яка направлена в бік батьківського класу;
- замість терміну «композиційна сутність» реляційної бази даних в ООБД застосовується термін клас-асоціація. Такі класи приєднуються до відповідного зв'язку пунктирною лінією. На рисунку кожен з класів MaterialSupplied та MaterialUse прикріплений, щонайменше, до одного зв'язку «більшість до більшості» пунктирною лінією.



10

## ФУНКЦІОНАЛЬНА СХЕМА ОБ'ЄКТНО-ОРІЄНТОВАНОЇ БАЗИ ДАНИХ



Функціональна схема – це схема, що роз'яснює певні процеси, які відбуваються у певних функціональних частинах виробу (устаткування) чи у виробі (устаткуванні) в цілому.

Функціональна схема містить інформацію про способи реалізації пристроєм заданих функцій. За такою схемою можна визначити, як здійснюються перетворення і які для цього необхідні функціональні елементи. Кожен функціональний елемент містить лише ті входи та виходи, які необхідні для його коректної роботи. Дана схема розробляється на основі структурної схеми для кожного блоку, в результаті з окремих функціональних елементів складається загальна функціональна схема об'єкту.

Також функціональні схеми можуть застосовуватися у програмуванні для візуалізації алгоритмів та спрощення обчислення їх складності, однак у цій сфері форма створення – довільна (точніше, обирається та, яка зручна автору).

## ПРИКЛАД ФІЗИЧНОЇ МОДЕЛІ ОБ'ЄКТНО-ОРІЄНТОВАНОЇ БАЗИ ДАНИХ

```
class Student : Person
{
    attribute string studentIdNumber; //Номер студентського квитка
    attribute string gradeBookNumber; //Номер залікової книжки
    relationship set <Parent> student_of
        inverse Parent :: parent_of;
    relationship Teacher curator
        inverse Teacher :: student;
    void setStudentIdNumber(in string iStudentIdNumber)
    raises(invalidStudentNumber);
    void setGradeBookNumber(in string iGradeBookNumber)
    raises(invalidGradeBookNumber);
    boolean addCurator(in Teacher iTeacher)
    raises (noSuchTeacher);
    string getStudentIdNumber();
    string getGradeBookNumber();
};

class Bachelor extends Student (extent bachelors)
{
    attribute integer currentGrade; //Теперішній середній бал
    void setCurrentGrade (in string iCurrentGrade);
    integer getCurrentGrade();
};

class Master extends Student (extent masters)
{
    attribute integer practiceGrade; //Оцінка за практику
    attribute string topicOfSertificationWork; //Тема роботи
    void setPracticeGrade (in string iPracticeGrade);
    void setTopicOfSertificationWork (in string iTopicOfSertificationWork);
    integer getPracticeGrade();
    string getTopicOfSertificationWork();
};
```

## ПІДСУМКИ

Першою формалізованою та загальновізнаною моделлю даних була реляційна модель Коада. До ролі стандарту для зображення логічних моделей даних найближче наблизилась уніфікована мова моделювання (Unified Modeling Language, UML). У її нотації об'єднані багато елементів, які розглядалися в багатьох методиках.

Одна з причин, за допомогою якої реляційні бази даних отримали велику популярність, полягає в існуванні стандартної мови маніпулювання даними (SQL). Через це в стандарті проекту ООБД визначені дві мови: Object Definition Language (ODL) це – мова визначення об'єктів, та Object Query Language (OQL) – мова запити об'єктів. В ООБД разом вони будуть грати ту саму роль, яка відведена SQL в реляційних базах.

Підводячи підсумки, можна відзначити, що об'єктно-орієнтовані бази даних дуже зручні в реалізації інформаційних структур практично будь-якої компанії або підприємства. Головною задачею стає виділення основних сутностей та їх атрибутів, а також сутнісних відносин, що вимагає ознайомлення з об'єктною областю застосування БД. Також необхідно пам'ятати, що на етапах логічного та фізичного проектування всі рішення необхідно погоджувати з замовником.

Якщо є необхідність зберегти поведінку або дані дуже складні, тобто складаються з декількох примітивних типів даних, – слід розглянути можливість використання об'єктно орієнтованої бази даних. В іншому випадку, в більшості випадків переважніше використовувати реляційну базу даних.

## ВИСНОВКИ

У ході атестаційної роботи були виконані наступні задачі:

- визначені основні принципи об'єктно-орієнтованої парадигми;
- розглянуті популярні об'єктно-орієнтовані системи управління базами даних;
- розглянуті основні етапи проектування об'єктно-орієнтованої бази даних;
- розроблена модель проектування програмного та інформаційного забезпечення ООБД;
- виконано проектування фізичної моделі об'єктно-орієнтованої бази даних;
- порівняно проблеми, які виникають в реляційних та об'єктно-орієнтованих базах даних.