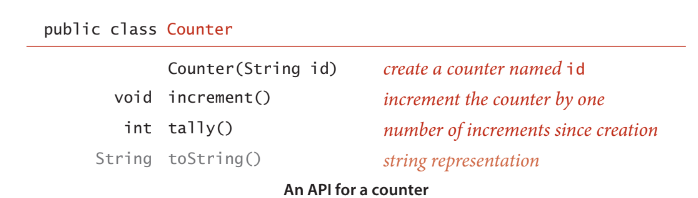
**1.2 Data Abstraction thuộc Chapter 1:**

1. Giới thiệu về Trừu tượng hóa dữ liệu (Data Abstraction)

Trừu tượng hóa dữ liệu là một khái niệm cơ bản trong lập trình hướng đối tượng, cho phép chúng ta định nghĩa các kiểu dữ liệu tùy chỉnh (custom data types) để quản lý các dữ liệu phức tạp. Với trừu tượng hóa dữ liệu, lập trình viên chỉ cần quan tâm đến cách sử dụng dữ liệu mà không cần biết cách dữ liệu được lưu trữ hoặc xử lý bên trong.

* Trừu tượng hóa dữ liệu tách biệt cách lưu trữ và cách thao tác dữ liệu.
* Lớp (Class) là một mô hình định nghĩa đối tượng bao gồm thuộc tính và hành vi.
* Đối tượng (Object) là một thể hiện của lớp, đại diện cho một đối tượng cụ thể trong chương trình.



Việc trừu tượng hóa giúp chia bài toán phức tạp thành các phần nhỏ hơn, quản lý dữ liệu dễ dàng hơn và cải thiện tính tổ chức của mã nguồn.

2. Kiểu dữ liệu trừu tượng (ADT)

Kiểu dữ liệu trừu tượng (ADT) là một kiểu dữ liệu mà các thao tác trên dữ liệu được xác định mà không cần quan tâm đến chi tiết cài đặt. Người dùng chỉ cần biết các thao tác có thể thực hiện và kết quả trả về, mà không cần hiểu cách triển khai.

Một số thao tác cơ bản có thể bao gồm việc tạo đối tượng, thực hiện các phép tính hoặc thao tác với dữ liệu. Các chi tiết về việc lưu trữ và xử lý bên trong được ẩn đi, giúp giảm sự phức tạp trong quá trình phát triển phần mềm.

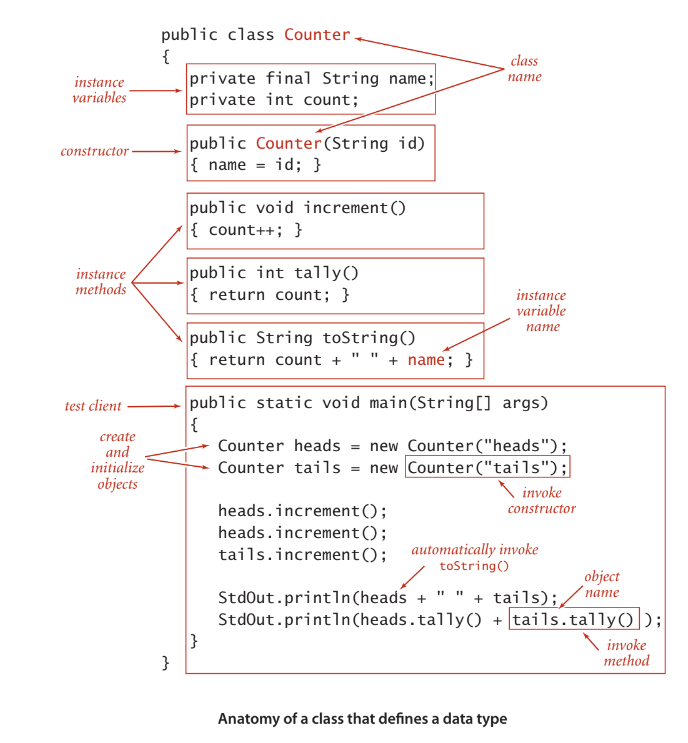


Lợi ích của ADT:

* Tái sử dụng mã: Có thể được sử dụng lại trong nhiều ứng dụng khác nhau.
* Dễ bảo trì: Thay đổi cách lưu trữ mà không ảnh hưởng đến chương trình.
* Dễ đọc và dễ hiểu: Tổ chức mã nguồn rõ ràng, dễ dàng hiểu và làm việc.

3. Lớp và đối tượng trong Java

Trong lập trình Java, lớp (class) được sử dụng để định nghĩa kiểu dữ liệu trừu tượng. Lớp chứa các thuộc tính (biến) và phương thức (hàm), giúp quản lý và thao tác dữ liệu. Các đối tượng là các phiên bản của lớp và có thể thao tác trực tiếp với các dữ liệu được định nghĩa trong lớp.

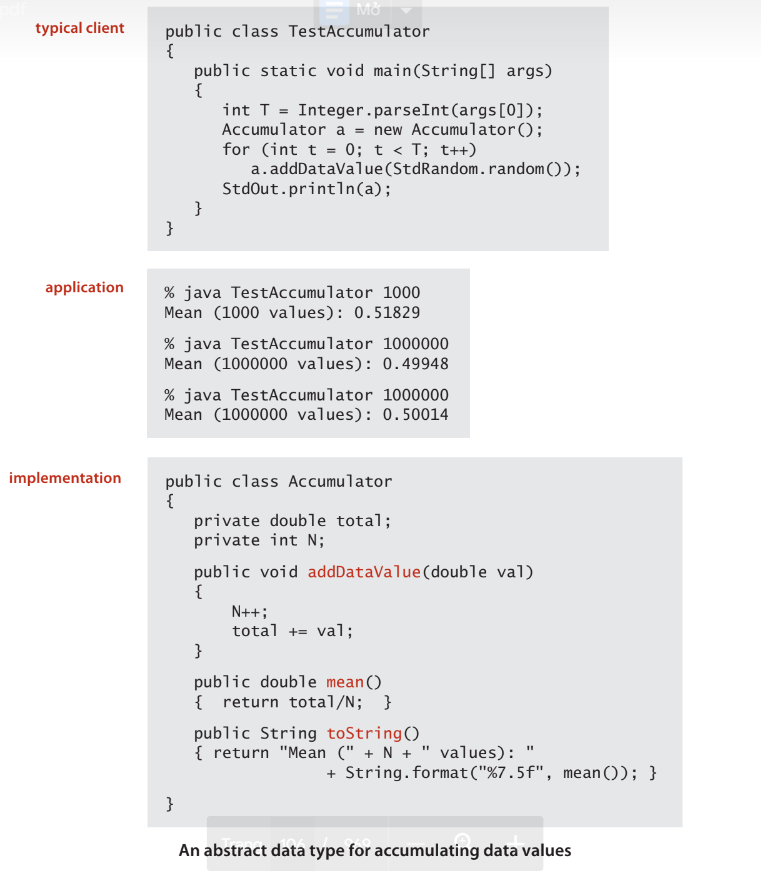


4. Đóng gói dữ liệu (Encapsulation)

Đóng gói dữ liệu là nguyên tắc quan trọng của trừu tượng hóa dữ liệu. Các dữ liệu và chi tiết triển khai bên trong một lớp được ẩn khỏi bên ngoài, chỉ có thể truy cập thông qua các phương thức cụ thể được cung cấp bởi lớp. Điều này giúp bảo vệ dữ liệu khỏi các thay đổi không mong muốn và tăng tính an toàn của chương trình.

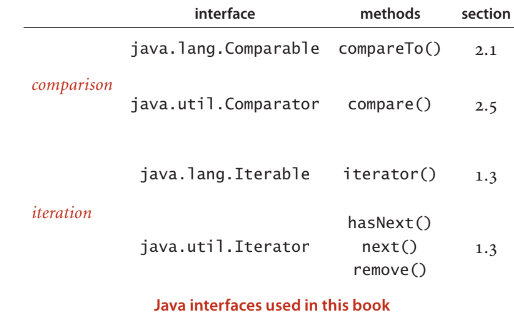
* Biến private: Chỉ cho phép truy cập bên trong lớp.
* Phương thức public: Cho phép truy cập dữ liệu một cách có kiểm soát thông qua các phương thức cụ thể.

Việc đóng gói dữ liệu giúp chương trình ổn định hơn và dễ dàng bảo trì khi có sự thay đổi.



5. Hàm khởi tạo (Constructors)

Constructor là một phương thức đặc biệt trong Java được sử dụng để khởi tạo đối tượng. Nó không có kiểu trả về và có cùng tên với lớp. Constructor giúp tạo ra các đối tượng và thiết lập các giá trị ban đầu cho các thuộc tính của đối tượng.



6. Hàm toString() và phương thức truy xuất dữ liệu

Lớp thường bao gồm các phương thức để truy xuất hoặc thay đổi giá trị của các thuộc tính. Phương thức toString() được sử dụng để biểu diễn đối tượng dưới dạng chuỗi, giúp việc in ra hoặc ghi log thông tin của đối tượng dễ dàng hơn. Phương thức này cũng có thể được ghi đè (override) để phù hợp với yêu cầu cụ thể của từng lớp.

7. Các phương thức tĩnh (Static Methods)

Phương thức tĩnh (static methods) là các phương thức không yêu cầu khởi tạo đối tượng để sử dụng. Phương thức này thuộc về lớp và có thể được gọi trực tiếp từ tên lớp. Chúng thường được sử dụng cho các chức năng chung, không liên quan đến các đối tượng cụ thể.

8. Tính đóng gói và kế thừa

Kế thừa (Inheritance) là một tính năng của lập trình hướng đối tượng cho phép một lớp mới kế thừa các thuộc tính và phương thức của một lớp hiện có. Điều này giúp tái sử dụng mã nguồn và tránh lặp lại các phần mã.

Kế thừa tạo điều kiện cho việc mở rộng và phát triển hệ thống một cách linh hoạt, bởi các lớp con có thể kế thừa các thuộc tính và hành vi của lớp cha và thêm vào các đặc tính riêng biệt của chúng.

Kết luận

Phần 1.2: Data Abstraction trong cuốn sách "Algorithms, Fourth Edition" cung cấp nền tảng quan trọng về trừu tượng hóa dữ liệu, từ cách tạo lớp và đối tượng đến việc sử dụng đóng gói và kế thừa. Hiểu rõ những khái niệm này giúp lập trình viên xây dựng các chương trình rõ ràng, dễ bảo trì và mở rộng. Trừu tượng hóa dữ liệu là một công cụ quan trọng giúp quản lý các hệ thống phức tạp và phát triển phần mềm hiệu quả.