1. Trình bày cách đặt tên trong Java cho biến (variable), hằng (constant), phương thức (method), lớp (class)

2. Giải thích các cách dùng của toán tử tăng (++), giảm (--)

3. (Đọc thêm) Tìm hiểu về toán tử gán: +=, -=, \*=, /= (Tham khảo phần 2.13 sách giáo trình)

Bài Làm

1

Biến Variable:

* Tên biến phải bắt đầu bằng một chữ cái (a-z hoặc A-Z), hoặc dấu gạch dưới (\_).

Ví dụ: ‘int age’,’String firstName’, ‘double \_balance’.

* Sau ký tự đầu tiên, bạn có thể sử dụng bất kỳ chữ cái, số hoặc dấu gạch dưới nào.

ví dụ: ‘int studentAge’, ‘double account\_balance’, ‘String user\_name’, ‘int var123’.

* Không được sử dụng các từ khóa của Java (reserved keywords) như ‘int’, ‘if’, ‘else’, ‘for’,... làm tên biến.
* Sử dụng các từ viết thường cho tên biến và tuân thủ quy tắc camelCase (chữ cái đầu viết thường và các từ tiếp theo viết hoa).

Ví dụ: ‘int studentAge’, ‘String userName’, ‘double accountBalance’.

Hằng Số (Constant):

* Tên hằng số nên được viết hoa và sử dụng dấu gạch dưới (\_) để phân tách các từ.

ví dụ: ‘final int MAX\_AGE’. ‘final doule PI\_VALUE’.

* Nếu tên hằng số gồm nhiều từ, hãy sử dụng dấu gạch dưới (\_) hoặc viết hoa các chữ cái đầu của từ.

Ví dụ: ‘final int MAXIMUM\_SCORE’,’final int MIN\_SCORE’.

Phương Thức (Method):

* Tên phương thức nên bắt đầu bằng một chữ cái (a-z hoặc A-Z).

Ví dụ: ‘void calculateTotal()’, ‘String getFullName()’.

* Sau ký tự đầu tiên, bạn có thể sử dụng bất kỳ chữ cái, số hoặc dấu gạch dưới nào.

Ví dụ: ‘void calculateTatalAmount()’, ‘int getAgeInYears()’.

* Không sử dụng các từ khóa của Java Làm lên phương thức.

Lớp(Class):

* Tên lớp nên bắt đầu bằng một chữ cái viết hoa và theo sau là chữ cái và số.

Ví dụ: ‘Class Person’, ‘Class StudentDetails’ , ‘Class Mycalculator’.

* Sử dụng quy tắc CamelCase cho tên lớp, Nghĩa là viết hoa chữ cái đầu mỗi từ.

Ví dụ: ‘class StudentRecord’, ‘class AddressInformation’.

2

* Toán Tử tăng (‘++’):
* Tiền tố (‘++i’): Tăng giá trị của biến ‘i’ lên một đơn vị trước khi sử dụng giá trị mới trong biểu thức khác.

VD: int i = 5;

int result = ++i; // i trở thành 6, và result cũng là 6

* Hậu tố (‘i++’): Sử dụng giá trị hiện tại của biến ‘i’ trong biểu thức, sau đó tăng giá trị của ‘t’ lên một đơn vị.

VD: int i = 5;

int result = i++; // result là 5, sau đó i trở thành 6

* toán tử giảm (‘--’):
* tiền tố(‘--i’): Giảm giá trị của biến ‘i’ đi một đơn vị trước khi sử dụng giá trị mới trong biểu thức khác.

VD: int i = 5;

int result = -- i; // i trở thành 4, và result cũng là 4

* hậu tố(‘i- -’): sử dụng giá trị hiện tại của biến ‘i’ trong biểu thức, sau đó giảm giá của ‘i’ đi một đơn vị.

VD i = 5;

int result = i- -; // result là 5, sau đó i trở thành 4