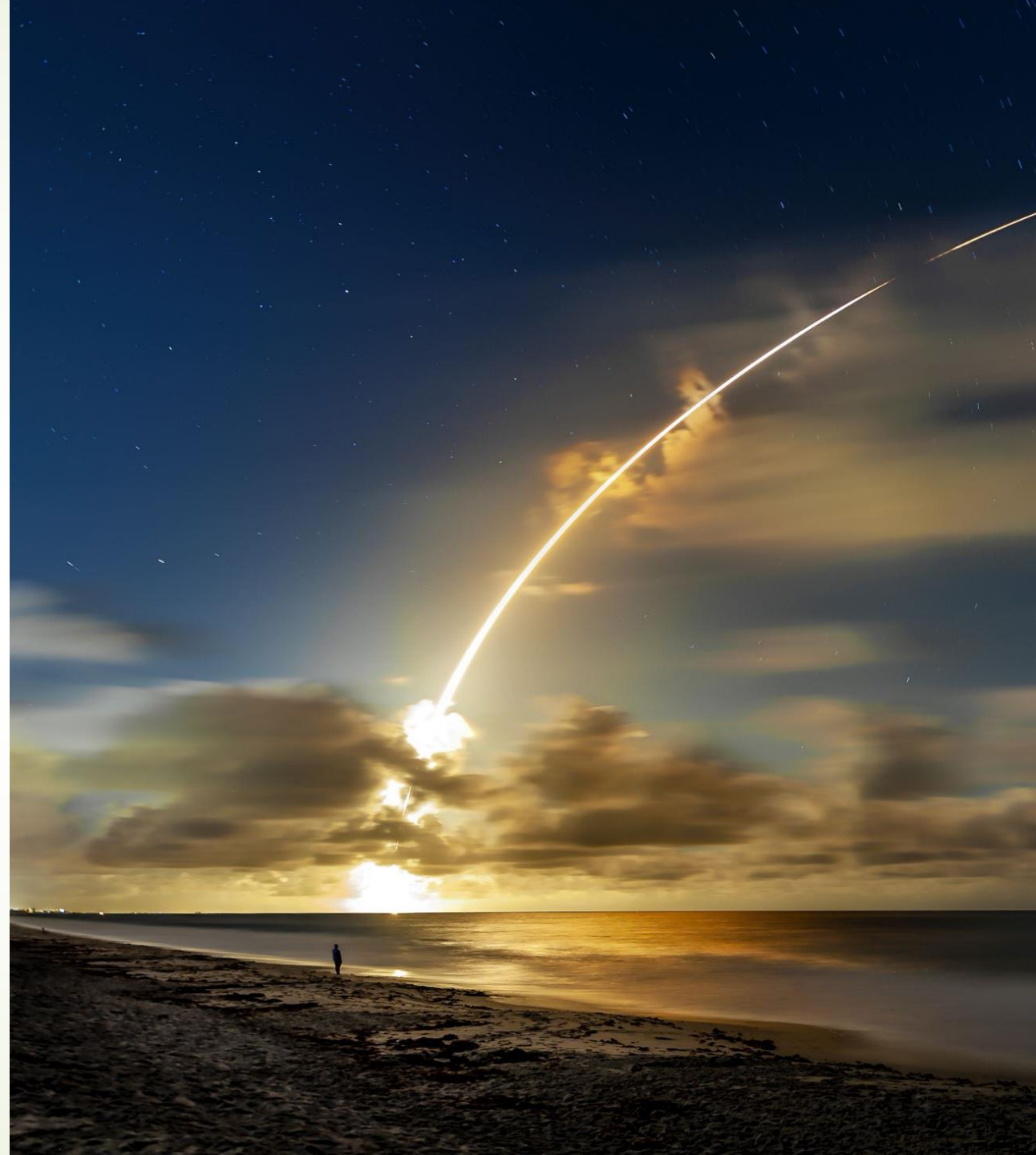


## Bài 4: Các toán tử trong Java

- ✓ Khái niệm và quy ước
- ✓ Các toán tử số học
- ✓ Các toán tử gán
- ✓ Các toán tử so sánh
- ✓ Các toán tử logic



# Khái niệm và quy ước

- Toán tử hiệu đơn giản là các phép toán
- Toán tử 1 ngôi là toán tử chỉ cần 1 toán hạng để hoàn thành biểu thức
- Toán tử 2 ngôi là toán tử cần có 2 toán hạng để hoàn thành biểu thức
- Nếu phép toán có nhiều hơn 1 kí tự thì ta viết liền. Ví dụ:  $\geq$  thay vì  $> =$
- Luôn phân tách các toán tử hai ngôi với toán hạng của nó bằng 1 dấu cách cho chương trình dễ đọc

# Các toán tử số học

- Là các toán tử  $+$ ,  $-$ ,  $*$ ,  $/$ ,  $\%$ ,  $++$ ,  $--$
- Toán tử  $+$ ,  $-$ :
  - Nếu được sử dụng như toán tử 1 ngôi thì phải đặt trước toán hạng
  - Thứ tự ưu tiên từ trái qua phải
  - Các toán hạng của nó phải là kiểu số hoặc kiểu lớp bao của các kiểu số
  - Khi thực hiện  $+$ ,  $-$  có thể xảy ra tràn số nếu giá trị nhận được quá khả năng lưu trữ của kiểu tương ứng
- Nếu toán hạng có chứa kiểu float hoặc double thì:
  - Nếu 1 trong hai toán hạng là NaN thì kết quả là NaN
  - Tổng của hai giá trị vô cực trái dấu là NaN
  - Tổng của hai giá trị vô cực cùng dấu là giá trị vô cực cùng dấu

## Các toán tử +, -

- Nếu toán hạng có chứa kiểu float hoặc double thì:
  - Tổng của một giá trị hữu hạn và vô cực là vô cực
  - Tổng của +0 và -0 là +0
  - Tổng của giá trị khác 0 và 0 là giá trị khác 0
  - Tổng của hai giá trị cùng độ lớn nhưng trái dấu là +0
- Trường hợp phép + với String:
  - Nếu một toán hạng trong biểu thức là String thì kết quả là String
  - Phép + còn gọi là phép nối chuỗi
  - Ví dụ "hello" + 5 + 6 cho kết quả là "hello56" không phải "hello11"

## Các toán tử $*$ , $/$ , $\%$

- Thứ tự ưu tiên thực hiện biểu thức từ trái qua phải
- Đây là các toán tử hai ngôi, các toán hạng của nó phải là kiểu số hoặc lớp bao của kiểu số
- Kết quả của phép toán sẽ được lưu vào kiểu lớn nhất trong các kiểu tham gia vào biểu thức
- Nếu các toán hạng nguyên và thực hiện biểu thức xảy ra tràn số thì giá trị nhận được có thể khác dấu với giá trị đúng của nó
- Nếu phép nhân mà có toán hạng kiểu số thực:
  - Nếu 1 toán hạng là NaN thì kết quả là NaN
  - Nếu kết quả khác NaN thì dấu của kết quả quy ước như nhân hai số trong toán học
  - Nhân NaN với vô cực được NaN



## Các toán tử $*$ , $/$ , $\%$

- Nếu phép nhân mà có toán hạng kiểu số thực:
  - Nhân giá trị hữu hạn và vô cực được vô cực có dấu xét theo quy tắc nhân
  - Nếu các toán hạng khác NaN và vô cực thì thực hiện như quy tắc nhân thông thường
- Phép chia  $/$ :
  - Nếu tử số và mẫu số là kiểu nguyên thì kết quả sẽ là kiểu nguyên
  - Nếu tử và mẫu đều là số nguyên và mẫu  $= 0$  thì xảy ra ngoại lệ chia cho 0
- Phép chia với ít nhất 1 toán hạng là kiểu thực
  - Nếu 1 toán hạng là NaN kết quả là NaN
  - Nếu kết quả khác NaN thì dấu được xác định theo quy tắc nhân
  - Chia vô cực cho nhau được NaN
  - Chia vô cực cho giá trị giới hạn được vô cực có dấu xác định theo quy tắc nhân

# Các toán tử $*$ , $/$ , $%$

- Phép chia với ít nhất 1 toán hạng là kiểu thực
  - Chia 0 cho 0 được NaN. Nếu tử số là 0, mẫu khác 0 thì dấu xác định theo quy tắc nhân
  - Chia giá trị khác 0 cho 0 được vô cực có dấu xác định theo quy tắc nhân
  - Các trường hợp khác tính toán bình thường và kết quả làm tròn chữ số cuối cùng đến giá trị gần nhất
- Phép chia cũng có thể gây tràn số và cho kết quả không mong muốn
- Việc nhân hay chia có thể gây mất thông tin, tràn số, chia cho 0 nhưng không xảy ra ngoại lệ

# Các toán tử \*, /, %

- Phép chia lấy dư %:
  - Là lấy phần còn lại sau khi chia nguyên tử số cho mẫu
  - Áp dụng cho cả số nguyên và số thực
  - Lưu ý mẫu số khác 0 để không xảy ra ngoại lệ
  - Dấu của kết quả chỉ phụ thuộc vào dấu của tử số
- Khi thao tác với số thực:
  - Nếu 1 toán hạng là NaN thì kết quả là NaN
  - Nếu kết quả khác NaN thì dấu chỉ phụ thuộc dấu của tử số
  - Nếu tử số là vô cực hoặc mẫu là 0 hoặc cả hai thì kết quả là NaN
  - Nếu mẫu là vô cực và tử là hữu hạn thì kết quả là tử số
  - Nếu tử số bằng 0 và mẫu khác 0 thì kết quả là tử số



## Các toán tử ++, --

- Là toán tử 1 ngôi thực hiện việc tăng/giảm giá trị của 1 số đi 1 đơn vị
- Có thể đặt trước hoặc sau toán hạng đều được
- Toán hạng của các toán tử này phải là các kiểu số hoặc lớp bao của các kiểu số
- Kiểu của kết quả ++, -- cùng kiểu với kiểu của biến
- Các phép toán này chỉ ảnh hưởng đến giá trị của biến
- Các hằng số không thể áp dụng các toán tử trên

# Các toán tử gán

- Là các toán tử hai ngôi dùng để gán giá trị tương thích cho một biến cụ thể nào đó
- Các kiểu rộng hơn luôn có thể chứa giá trị của kiểu hẹp hơn. Thứ tự này trong Java là: byte->short->int->long->float->double
- Nếu muốn gán giá trị của kiểu rộng hơn cho kiểu hẹp hơn, ví dụ float->int thì ta phải ép kiểu tường minh và chấp nhận mất giá trị
- Thứ tự ưu tiên được thực hiện từ phải qua trái. Tức là nếu  $a = b = c$  thì đầu tiên gán  $b = c$  sau đó mới đến  $a = b$
- Toán tử gán có thể kết hợp với các toán tử số học và toán tử thao tác trên bit(không học ở khóa học này)
- Toán hạng trái của phép gán luôn là các biến thường, nếu là hằng số chỉ gán đc 1 lần đầu tiên và duy nhất

# Các toán tử gán

- Toán hạng phải của phép gán có thể là biến, hằng, giá trị cụ thể, biểu thức,...
- Kết quả của phép gán làm thay đổi giá trị của toán hạng trái. Nếu trong quá trình gán mà xảy ra lỗi thì giá trị của toán hạng trái không đổi
- Sau đây là cú pháp viết tắt của việc kết hợp phép gán và toán tử số học:

Biểu thức	Tương đương	Kết quả
$a = a + 5$	$a += 5$	$a = 10$
$a = a - b$	$a -= b$	$a = -1$
$a = a * b$	$a *= b$	$a = 30$
$a = a / b$	$a /= b$	$a = 0$
$a = a \% b$	$a \% = b$	$a = 5$

# Các toán so sánh

- Là các toán tử hai ngôi dùng để so sánh mối quan hệ giữa hai toán hạng
- Ta thường chỉ so sánh các toán hạng kiểu số hoặc kiểu lớp bao của các kiểu số
- Kết quả phép so sánh là kiểu Boolean theo đúng mối quan hệ của hai toán hạng
- Việc so sánh hai số thực luôn cho kết quả chính xác
  - Nếu một trong hai toán hạng là NaN thì kết quả là false
  - Tất cả các giá trị khác theo thứ tự: âm vô cực < giá trị hữu hạn < dương vô cực
  - -0.0 tương đương 0.0 nhưng phương thức min, max của lớp Math coi  $-0.0 < 0.0$
- Với toán tử == và !=:
  - Thứ tự ưu tiên từ trái qua phải
  - Hai toán hạng phải cùng kiểu để đảm bảo không bị lỗi

# Các toán so sánh

- Với toán tử `==` và `!=`:
  - Thứ tự ưu tiên từ trái qua phải
  - Hai toán hạng phải cùng kiểu để đảm bảo không bị lỗi
  - Phép `==` cho kết quả true khi và chỉ khi hai biểu thức trái phải tương đương
  - Phép `!=` cho kết quả ngược với `==`
  - Biểu thức `a != b` tương đương `!(a == b)`
- Với số thực:
  - Nếu 1 trong các toán hạng là NaN thì kết quả của `==` là false trong khi `!=` là true
  - Có thể dùng `Float.isNaN` hoặc `Double.isNaN` để kiểm tra giá trị của biến nào đó có phải NaN không
  - `-0` và `0` là tương đương, tức `-0.0 == 0.0` cho kết quả true



# Các toán tử logic

## ➤ Toán tử &&:

- Thứ tự ưu tiên thực hiện biểu thức từ trái qua phải
- Là toán tử hai ngôi với hai toán hạng trái phải cùng là kiểu Boolean hoặc boolean
- Toán hạng phải chỉ được đánh giá nếu toán hạng trái true
- Kết quả thu được luôn ở kiểu boolean

## ➤ Toán tử ||:

- Thứ tự ưu tiên thực hiện biểu thức từ trái qua phải
- Là toán tử hai ngôi với hai toán hạng trái phải bắt buộc là kiểu boolean
- Toán hạng phải chỉ được đánh giá khi toán hạng trái false
- Kết quả true/false

# Các toán tử logic

## ➤ Toán tử phủ định !:

- Luôn đứng trước toán hạng kiểu Boolean hoặc boolean
- Kết quả luôn ở kiểu boolean
- Nếu  $x = \text{true}$  thì  $!x == \text{false}$
- Nếu  $x = \text{false}$  thì  $!x == \text{true}$

## ➤ Bảng giá trị chân lý:

a	b	a && b	a    b	!a
0	0	0	0	1
0	1	0	1	1
1	0	0	1	0
1	1	1	1	0

# Nội dung tiếp theo

**Ép kiểu và làm tròn số**