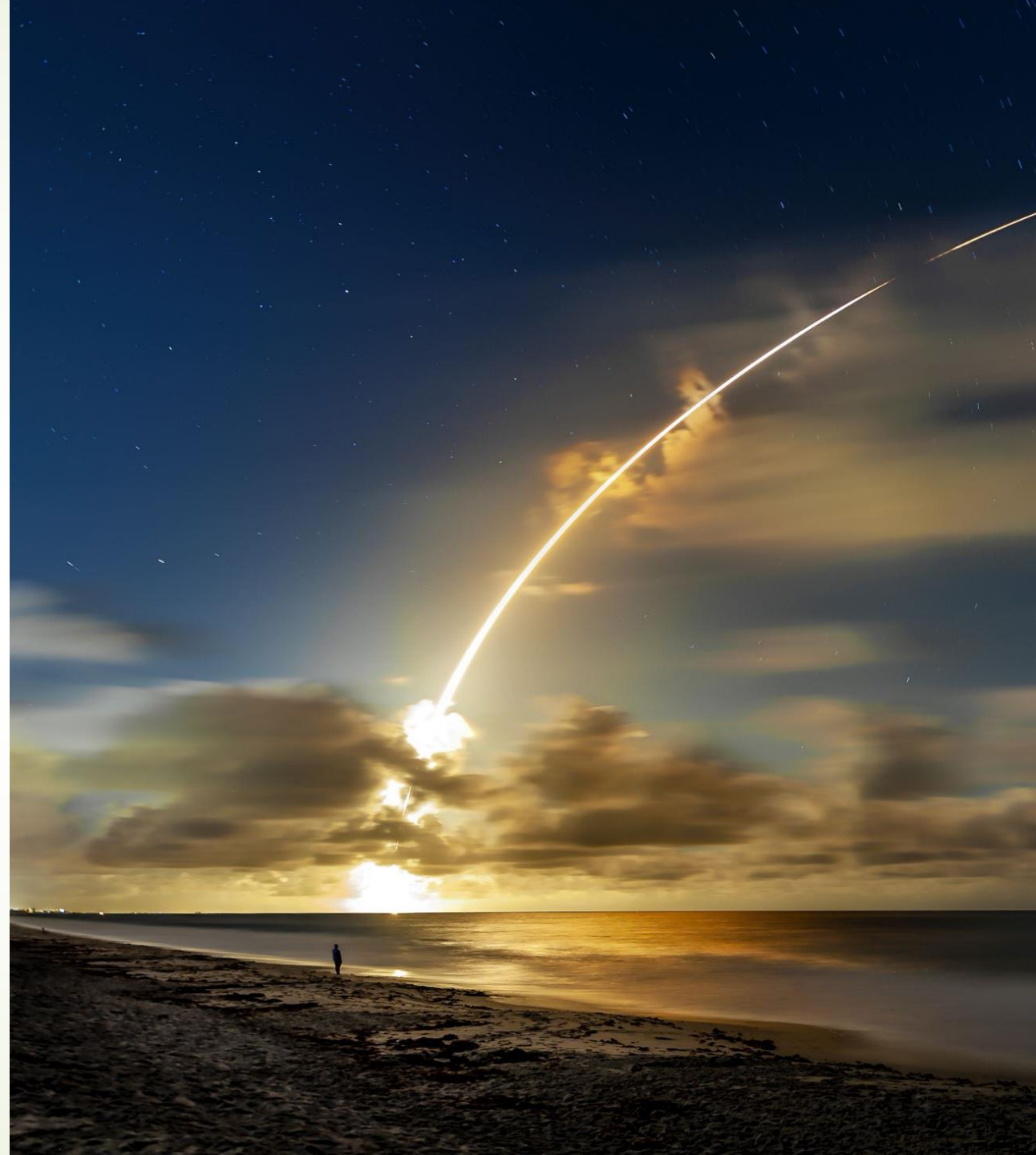


Bài 5: Thao tác ép kiểu và làm tròn số

- ✓ Khái niệm ép kiểu
- ✓ Lý do phải ép kiểu
- ✓ Có những cách ép kiểu nào
- ✓ Làm tròn số
- ✓ Ví dụ minh họa



Khái niệm ép kiểu

- Ép kiểu là hành động đưa giá trị của một kiểu sang một kiểu khác. Kiểu ở đây là các kiểu dữ liệu hợp lệ trong ngôn ngữ Java như các kiểu nguyên thủy, tham chiếu.
- Hành động ép kiểu có thể thể hiện tường minh: chỉ rõ kiểu đích. Hoặc ngầm định: không chỉ rõ kiểu đích.
- Ví dụ:
 - Ép kiểu từ int sang float
 - Ép kiểu float sang int
 - Ép kiểu double sang float
 - Ép kiểu double sang long

Lý do cần ép kiểu

- Trong một ngữ cảnh nào đó, ta sẽ cần một kiểu dữ liệu cụ thể để thao tác được chính xác. Để đảm bảo điều này có thể thực hiện thành công ta cần phải ép kiểu.
- Ví dụ:
 - Để có giá trị chính xác của phép chia $2/3$ ta phải ép kiểu tử hoặc mẫu số sang kiểu số thực float hoặc double.
 - Để lấy phần nguyên của giá trị 3.14159 ta ép kiểu nó sang số nguyên kiểu int chẳng hạn.
 - Khi muốn các giá trị nguyên thủy có thể thao tác như các đối tượng, ta ép kiểu sang kiểu lớp bao của nó. Ví dụ: `Integer intObj = 200;`

Các cách ép kiểu

- Có nhiều cách ép kiểu khác nhau:
 - Ép sang cùng kiểu
 - Ép sang các kiểu rộng hơn
 - Ép sang các kiểu hẹp hơn
 - Ép sang kiểu rộng hơn sau đó ép sang kiểu hẹp hơn
 - Ép kiểu đóng hộp
 - Ép kiểu mở hộp
- Các thao tác ép kiểu nâng cao khác và ép kiểu đối tượng sẽ học ở các bài sau

Ép sang cùng kiểu

- Đây là cách ép kiểu hay còn gọi chuyển đổi phổ thông nhất.
- Bạn có thể gán giá trị của các biến cùng kiểu cho nhau.
- Hành động này không xảy ra lỗi nào cả.
- Ép kiểu sang giá trị cùng kiểu mặc định là ép kiểu ngầm định.
- Ta vẫn có thể thể hiện ép kiểu cách này một cách tường minh nếu muốn.
- Khi ép kiểu theo cách này, dữ liệu gốc được bảo toàn.
- Ví dụ:

```
int a = 200;  
int b = a; // ok - ép kiểu ngầm định  
int c = (int)b; // ok - ép kiểu tường minh
```


Ép sang kiểu rộng hơn

- Là ép kiểu từ kiểu kích thước nhỏ hơn sang kiểu kích thước lớn hơn hoặc ép từ kiểu con sang kiểu cha.
- Các thao tác ép kiểu theo cách này sẽ được thực hiện ngầm định.
- Ép kiểu cách này dữ liệu gốc sẽ bảo toàn nếu:
 - Ép từ kiểu nguyên nhỏ hơn sang kiểu nguyên lớn hơn
 - Ép từ byte, short, char sang kiểu số thực
 - Ép từ int sang double
 - Từ float sang double ở biểu thức strictfp
- Ví dụ:

```
int a = 1234567890;  
float f = a;  
System.out.println(a - (int)f);  
// Kết quả -46
```

Ép sang kiểu hẹp hơn

- Là ép kiểu từ kiểu kích thước lớn hơn sang kiểu kích thước nhỏ hơn.
- Các thao tác ép kiểu theo cách này phải được thực hiện tường minh.
- Ép kiểu cách này sẽ bị mất dữ liệu. Lý do là kiểu nhỏ hơn chỉ có thể chứa được n bit thấp trong tổng số các bit của kiểu rộng hơn.

int	00000000000000011100000000000001
short	10000000000000001
Sau khi ép	10000000000000001

- Ví dụ:

```
float f = 1.25f;  
int newValue = (int)f;  
System.out.println(newValue);  
// kết quả: 1
```

Ép sang kiểu rộng hơn rồi sang kiểu hẹp hơn

- Trường hợp này xảy ra khi ép kiểu byte sang char. Các bước thực hiện là:
 - Đầu tiên byte sẽ được ép kiểu rộng hơn sang int
 - Sau đó từ int ép sang kiểu hẹp hơn là char

Ép kiểu đóng hộp

- Là cách ép kiểu đưa giá trị nguyên thủy sang giá trị đối tượng của kiểu lớp bao tương ứng của nó.
- Ví dụ:
 - Ép từ boolean sang Boolean
 - Ép từ int sang Integer
 - Ép từ float sang Float
- Ví dụ:

```
int x = 200; // x là kiểu nguyên thủy
Integer xObject = x; // xObject là kiểu tham chiếu

System.out.println(xObject.toString()); // ok
System.out.println(xObject.compareTo(200)); // ok

System.out.println(x.toString()); // error
System.out.println(x.compareTo(200)); // error
```

Ép kiểu mở hộp

- Là cách ép kiểu đưa giá trị kiểu lớp bao tương ứng về kiểu nguyên thủy tương ứng.
- Việc chuyển đổi giữa ép kiểu đóng hộp và mở hộp thường xuyên được thực hiện trong Java một cách ngầm định hoặc tường minh.
- Ví dụ:
 - Ép từ Boolean sang boolean
 - Ép từ Integer sang int

➤ Ví dụ:

```
Integer xObject = 500; // xObject là kiểu tham chiếu
int x = xObject.intValue(); // x là kiểu nguyên thủy
// tương đương:
x = xObject;

Double dObject = 123.456;
double d = dObject.doubleValue();
// tương đương:
d = dObject;
```

Làm tròn số

- Thường xảy ra với các giá trị số thực. Các cách làm tròn số:
 - Dùng phương thức `Math.ceil()` để làm tròn lên số nguyên gần nhất.
 - Dùng phương thức `Math.floor()` để làm tròn xuống số nguyên gần nhất.
 - Dùng phương thức `Math.round()` để làm tròn tự động đến số nguyên gần nhất. Quy ước làm tròn lên nếu phần thập phân ≥ 0.5 và ngược lại làm tròn xuống.
 - Làm tròn đến k chữ số ở phần thập phân với định dạng `%n.kf`. Giá trị n là tổng số kí tự dành để in ra kết quả.
 - Sử dụng `DecimalFormat`, trong đó số chữ số cần làm tròn được đánh dấu bởi # và phân tách phần nguyên, thập phân bằng dấu .

Làm tròn số

➤ Ví dụ:

```
double x = 2;  
double y = 3;  
double result = x / y; // kết quả 0.6666666(6)  
System.out.println(Math.ceil(result)); // kq = 1  
System.out.println(Math.floor(result)); // kq = 0  
System.out.println(Math.round(result)); // kq = 1  
// sử dụng DecimalFormat  
NumberFormat numberFormat = new DecimalFormat("###.#####");  
System.out.println(numberFormat.format(result));  
// sử dụng printf và chuỗi chuyển đổi định dạng:  
System.out.printf("%10.9f", result);
```

```
1.0  
0.0  
1  
0.666666667  
0.666666667
```

Nội dung tiếp theo

**Lớp String của
Java**