

Bài tập 2: Tính chu vi, diện tích hình tròn (Java)



Phân tích bài toán

Mục tiêu:

Sử dụng `final` để tạo hằng số `so_PI` (không thể thay đổi giá trị).
Thực hiện phép toán số học để tính chu vi, diện tích.
Dùng `Math.pow()` để tính lũy thừa (bán kính bình phương).
In kết quả với `printf` làm tròn 2 chữ số thập phân.

Công thức:

Chu vi = $2 \times \pi \times r$

Diện tích = $\pi \times r^2$

```
Welcome X HeSeNhanSu.java 1 X HinhTron.java 1 X
D: > Java > BT_B2 > J HinhTron.java > HinhTron
1 public class HinhTron {
2     Run | Debug
3     public static void main(String[] args) {
4         // ===== KHAI BAO HANG SO VA BIEN =====
5
6         // Hang so SO_PI: dung tu khoa final -> KHONG the thay doi gia tri
7         final double SO_PI = 3.14159;
8
9         // Ban kinh hinh tron
10        double banKinh = 5.5;
11
12        // ===== TINH TOAN =====
13
14        // Chu vi = 2 * PI * banKinh
15        double chuVi = 2 * SO_PI * banKinh;
16
17        // Dien tich = PI * banKinh^2
18        // Cach 1: Dung phép nhân
19        double dienTich1 = SO_PI * banKinh * banKinh;
20
21        // Cach 2: Dung Math.pow()
22        double dienTich2 = SO_PI * Math.pow(banKinh, 2);
23
24        // ===== IN KET QUA =====
25
26        System.out.println(x: "===== TINH TOAN HINH TRON =====");
27        System.out.printf(format: "Hang so PI          : %.5f\n", SO_PI);
28        System.out.printf(format: "Ban kinh           : %.2f\n", banKinh);
29        System.out.println(x: "-----");
30        System.out.printf(format: "Chu vi hinh tron   : %.2f\n", chuVi);
31        System.out.printf(format: "Dien tich (cach 1): %.2f\n", dienTich1);
32        System.out.printf(format: "Dien tich (cach 2): %.2f\n", dienTich2);
33    }
34 }
```

Giải thích chi tiết

1. Từ khóa `final` - Tạo hằng số

```
final double SO_PI = 3.14159;
```

`final` = **hằng số**, giá trị không thể thay đổi sau khi gán.

Quy ước đặt tên hằng số: **VIẾT HOA_TOÀN_BỘ** (UPPER_SNAKE_CASE).

Nếu cố tình thay đổi → **lỗi biên dịch**.

Ví dụ lỗi khi cố thay đổi:

```
final double SO_PI = 3.14159; SO_PI = 3.14; // ❌ LỖI: cannot assign a value  
to final variable SO_PI
```

2. Toán tử số học trong Java

Toán tử	Ý nghĩa	Ví dụ
<code>+</code>	Cộng	<code>a + b</code>
<code>-</code>	Trừ	<code>a - b</code>
<code>*</code>	Nhân	<code>a * b</code>
<code>/</code>	Chia	<code>a / b</code>
<code>%</code>	Chia lấy dư	<code>a % b</code>

3. Hai cách tính bình phương

Cách 1: Dùng phép nhân (nhANH hơn, đơn giản hơn)

```
double dienTich = SO_PI * banKinh * banKinh;
```

Cách 2: Dùng `Math.pow()` (linh hoạt cho lũy thừa bậc cao)

```
double dienTich = SO_PI * Math.pow(banKinh, 2);
```

Cú pháp `Math.pow()`:

```
Math.pow(cơ_số, số_mũ) // Trả về kiểu double
```

Ví dụ:

```
Math.pow(5.5, 2) = 30.25
```

```
Math.pow(2, 10) = 1024.0
```

```
Math.pow(9, 0.5) = 3.0 (căn bậc 2)
```

💡 **Lưu ý:** `Math.pow()` luôn trả về `double`, kể cả khi đầu vào là số nguyên.



Bảng so sánh 2 cách tính lũy thừa

Tiêu chí	<code>a * a</code>	<code>Math.pow(a, 2)</code>
Tốc độ	Nhanh hơn	Chậm hơn (gọi hàm)
Độ đơn giản	Đơn giản	Rõ ràng ý nghĩa
Linh hoạt	Chỉ bình phương	Mọi số mũ
Kiểu trả về	Theo biến gốc	Luôn <code>double</code>

Khuyến nghị:

Bình phương đơn giản → dùng `a * a`

Lũy thừa bậc cao (a^3 , a^4 , ...) hoặc số mũ thực → dùng `Math.pow()`