

Bài 18: Tổng quan về phương thức

- ✓ Khái niệm
- ✓ Cú pháp tổng quát
- ✓ Kiểu trả về của phương thức
- ✓ Ví dụ minh họa
- ✓ Bài tập thực hành

Khái niệm

- Phương thức là tập hợp của các câu lệnh cùng thực hiện trọn vẹn 1 chức năng và được đặt dưới 1 cái tên chung
- Mỗi phương thức chỉ thực hiện duy nhất 1 nhiệm vụ. Hay còn gọi là tác vụ, hành động, chức năng, task
- Do đó phương thức thường dùng để thể hiện hành động của một đối tượng nào đó
- Không thực hiện nhiều hơn 1 nhiệm vụ trong 1 phương thức
- Không tạo các phương thức quá dài, trên 40 dòng

Ví dụ

➤ Sau đây là một số ví dụ về phương thức:

```
public class Lesson18 {  
    public static void main(String[] args) {  
        System.out.println("Hello World!");  
    }  
}
```

➤ Phương thức do người dùng tự định nghĩa:

```
public static boolean isPrime(int n) {  
    if (n < 2) {  
        return false;  
    }  
    for (int i = 2; i <= Math.sqrt(n); i++) {  
        if (n % i == 0) {  
            return false;  
        }  
    }  
    return true;  
}
```

Cú pháp định nghĩa

- Cú pháp tổng quát liên quan phương thức gồm cú pháp định nghĩa phương thức và cú pháp gọi phương thức
- Cú pháp định nghĩa phương thức:

```
access non-access type name(params) {  
    // statements  
}
```

- Trong đó:
 - Phần access là access modifier, là mức độ hay khả năng được nhìn thấy từ bên ngoài. Sẽ trình bày cụ thể trong phần sau của khóa học. Bây giờ ta mặc định phần này là public: ở đâu cũng thấy và dùng đc

Cú pháp định nghĩa

➤ Trong đó:

- Phần non-access có thể có hoặc không, phần này có nhiệm vụ làm rõ tính chất của một phương thức. Cụ thể sẽ trình bày trong các bài học sau. Mặc định bây giờ phần này là static: cho phép truy cập không cần đối tượng
- Phần type là kiểu của phương thức, có thể là bất kì kiểu dữ liệu hợp lệ nào trong Java
- Phần name là tên phương thức. Tên phương thức là các hành động, nhiệm vụ mà phương thức thực hiện. Tên phương thức là động, cụm động từ, bắt đầu với chữ cái thường. Nếu tên có nhiều từ thì các từ viết liền và viết hoa chữ cái đầu từ kể từ từ kế tiếp. Tên cần đặt sao cho thể hiện được hành động mà phương thức đảm nhiệm

Cú pháp định nghĩa

➤ Trong đó:

- Sau tên là phần ngoặc tròn, đây là dấu hiệu nhận diện 1 phương thức. Trong ngoặc tròn là danh sách tham số
- Phần params là danh sách tham số, làm nhiệm vụ nhận dữ liệu đầu vào khi phương thức được gọi. 1 phương thức có thể có 0, 1 hoặc nhiều tham số. Nếu có nhiều tham số thì chúng sẽ phân tách nhau bởi dấu phẩy. Mỗi tham số là 1 biến hoặc hằng số có tên, kiểu cụ thể
- Phần thân phương thức nằm trong cặp {} chứa các lệnh cùng thực hiện chức năng mà phương thức đảm nhiệm
- Nếu phương thức trả về giá trị, kiểu của phương thức khác void và trong thân phương thức có lệnh `return giá_trị`; Nếu không trả về, kiểu phương thức là void và không có `return giá_trị`; có thể có `return`;

Ví dụ định nghĩa phương thức

➤ Sau đây là một số phương thức:

```
// phương thức trả về kiểu boolean, nhận 1 tham số
public static boolean isPrime(int n) {
    if (n < 2) {
        return false; // trả về false
    }
    for (int i = 2; i <= Math.sqrt(n); i++) {
        if (n % i == 0) {
            return false; // trả về false
        }
    }
    return true; // trả về true
}

// phương thức không trả về, không có tham số
public static void move() {
    System.out.println("Tôi đang bò trên mặt đất");
}
```

Cú pháp gọi phương thức

➤ Cú pháp thực hiện lời gọi phương thức:

```
name (params);  
// nếu muốn lưu lại giá trị trả về từ phương thức:  
variable = name (params);
```

➤ Trong đó:

- Lời gọi cần cung cấp đầy đủ tên phương thức, danh sách, thứ tự và số lượng các tham số
- Phần name là tên phương thức cần gọi
- Đi sau name là cặp ngoặc tròn, kết thúc lời gọi bằng dấu chấm phẩy
- Trong ngoặc tròn là danh sách, thứ tự đối số
- Số lượng đối số phải tương ứng số lượng tham số

Cú pháp gọi phương thức

➤ Cú pháp thực hiện lời gọi phương thức:

```
name (params);  
// nếu muốn lưu lại giá trị trả về từ phương thức:  
variable = name (params);
```

➤ Tham số vs đối số:

- Biến trong ngoặc tròn của định nghĩa phương thức gọi là tham số
- Tham số chỉ có thể là các biến, các hằng
- Thứ trong ngoặc tròn khi thực hiện lời gọi phương thức gọi là đối số
- Đối số có thể là biến, hằng, giá trị cụ thể, một biểu thức hoặc đối tượng nào đó

Ví dụ về lời gọi phương thức

➤ Sau đây là một số lời gọi phương thức:

```
public static void main(String[] args) {  
    // gọi phương thức khởi tạo của lớp scanner  
    var input = new Scanner(System.in); // gán giá trị có đc  
    cho input  
    // gọi phương thức đọc vào giá trị số nguyên:  
    int i = input.nextInt(); // gán giá trị nhận đc cho i  
    // gọi phương thức hiển thị dữ liệu  
    System.out.println("Hello World!"); // chỉ hiển thị k gán  
    cho biến nào  
    // gọi phương thức kiểm tra số ng.tố, gán giá trị cho biến  
    prime  
    boolean prime = isPrime(500); // truyền vào đối số là 500  
    // gọi phương thức move:  
    move(); // không truyền gì vào  
}
```

Kiểu trả về của phương thức

- Ta sẽ làm rõ khái niệm trả về và không trả về:
 - Một phương thức gọi là trả về nếu nó return một giá trị nào đó. Trong thân của phương thức luôn có dòng *return giá_trị;*
 - Phương thức trả về có kiểu khác void. Ví dụ kiểu int, float, String... trong các phương thức nextInt(), nextFloat() và nextLine() của Scanner
 - Sử dụng phương thức trả về khi muốn đánh giá tiêu chí nào đó hoặc giá trị có được khi thực hiện phương thức sẽ là đầu vào hoặc được tiếp tục sử dụng về sau bởi các thành phần khác trong chương trình
 - Về bản chất, phương thức trả về gửi trả cho nơi gọi nó một kết quả sau khi phương thức hoàn thành nhiệm vụ. Từ đó ta sẽ biết được trạng thái của việc thực hiện phương thức. Như đi thi cần biết kết quả thi vậy.

Kiểu trả về của phương thức

- Ta sẽ làm rõ khái niệm trả về và không trả về:
 - Ví dụ như khi ta đọc vào một số, tính tổng các chữ số, kiểm tra một số có phải số đẹp hay không, kiểm tra x có phải số chẵn hay không
 - Phương thức không trả về có kiểu là void. Nó có thể có return nhưng sau return không có giá trị mà là ;
 - Sử dụng phương thức không trả về khi kết quả thực hiện của phương thức là không quan trọng hoặc không cần báo trạng thái thực hiện cho ai cả hoặc là kết quả của phương thức đó không được dùng vào mục đích nào khác nữa
 - Ví dụ phương thức hiển thị kết quả, hiển thị thông báo, main

Các ví dụ:

- Ví dụ 1: Viết phương thức kiểm tra số nguyên n có phải số nguyên tố hay không.
- Ví dụ 2: Viết phương thức kiểm tra n có phải số đẹp hay không.
- Viết thêm các phương thức hỗ trợ khác nếu cần như hiển thị kết quả, thông báo, lời nhắc chẳng hạn.

Nội dung tiếp theo

Nạp chồng phương thức