|  |  |
| --- | --- |
| Print | In ra màn hình nhưng con trỏ sẽ không xuống dòng mà bắt đầu từ nó kết thúc câu lệnh đó. |
| Println | In ra màn hình nhưng con trỏ sẽ xuống dòng và bắt đầu từ dòng mới đó. |
| Các kiểu dữ liệu | |  |  | | --- | --- | | byte | Dùng để lưu dữ liệu kiểu số nguyên có kích thước một byte (8 bít). Phạm vi biểu diễn giá trị từ -128 đến 127. Giá trị mặc định là 0. | | char | Dùng để lưu dữ liệu kiểu kí tự hoặc số nguyên không âm có kích thước 2 byte (16 bít). Phạm vi biểu diễn giá trị từ 0 đến u\ffff. Giá trị mặc định là 0. | | boolean | Dùng để lưu dữ liệu chỉ có hai trạng thái đúng hoặc sai (độ lớn chỉ có 1 bít). Phạm vi biểu diễn giá trị là {“True”, “False”}. Giá trị mặc định là False. | | short | Dùng để lưu dữ liệu có kiểu số nguyên, kích cỡ 2 byte (16 bít). Phạm vi biểu diễn giá trị từ - 32768 đến 32767. Giá trị mặc định là 0. | | int | Dùng để lưu dữ liệu có kiểu số nguyên, kích cỡ 4 byte (32 bít). Phạm vi biểu diễn giá trị từ -2,147,483,648 đến 2,147,483,647. Giá trị mặc định là 0. | | long | Dùng để lưu dữ liệu có kiểu số nguyên có kích thước lên đến 8 byte. Giá trị mặc định là 0L. | | float | Dùng để lưu dữ liệu có kiểu số thực, kích cỡ 4 byte (32 bít). Giá trị mặc định là 0.0F. | | double | Dùng để lưu dữ liệu có kiểu số thực có kích thước lên đến 8 byte. Giá trị mặc định là 0.00D | |
| next() | - Không thể đọc những từ có khoảng trắng trong đó.  - Kết thúc việc đọc đầu vào sau khi nhận được khoảng trắng. |
| nextLine() | - Có thể đọc những từ có khoảng trắng trong đó.  - Kết thúc việc đọc đầu vào sau khi nhận được '\ n' hoặc nhấn Enter. |
| parseInt() | - Phương thức parseInt() là một phương thức của lớp Integer trong gói java.lang được sử dụng để chuyển String sang Int trong Java hoặc số thập phân thay bằng các kiểu dữ liệu vào interger ( số nguyên ). |
| Scanner | 1. Để nhập dữ liệu từ bàn phím trong Java, chúng ta sử dụng lớp Scanner 2. Lớp Scanner, là một lớp nằm trong gói java.util, được sử dụng để đọc dữ liệu đầu vào từ các nguồn khác nhau như luồng đầu vào, người dùng, tệp. 3. Đầu tiên phải import thử viện vào: import java.util.Scanner; 4. Sau đó ta gọi và khởi tạo đối tượng từ lớp Scanner: Scanner sc = new Scanner(System.in); 5. Mời người dùng nhập: System.out.prinln(“Mời bạn nhập Tên: ”); 6. Khai báo 1 biến để lưu giá trị cho người dùng nhập vào: String name = sc.nextLine(): 7. In ra màn hình: System.out.printf(“Chào bạn %s”,name); |
| Arrays.ToString | - Phương thức toString() của mảng được sử dụng để hiển thị tất cả các phần tử của một mảng. Phương thức này trả về chuỗi của tất cả các phần tử của một mảng. Chuỗi này bao gồm tất cả các phần tử được bao quanh trong “[]”. Tất cả các phần tử được phân tách bằng “,” (Dấu phẩy và khoảng cách). |
| Arrays.deepToString | Phương thức Arrays.deepToString () được sử dụng để nhận được biểu diễn chuỗi của các mảng đa chiều. Phương thức này trả về nội dung sâu của mảng được chỉ định. Nếu mảng được chỉ định chứa các mảng khác như là các phần tử thì nó sẽ trả lại nội dung của các mảng đó. |
| Math.sprt(value) | - Dùng để tính căn bậc 2 của một giá trị nào đó. |
| Biến Static  ( static trong method của đối tượng ) | Đôi khi, ta muốn một lớp có những biến dùng chung cho tất cả các đối tượng thuộc lớp đó. Ta gọi các biến dùng chung này là biến của lớp (class variable), hay gọi tắt là biến lớp. Chúng không gắn với bất cứ một đối tượng nào mà chỉ gắn với lớp đối tượng. Chúng được dùng chung cho tất cả các đối tượng trong lớp đó.  \* có nghĩa là biến static đặt một biến dùng chung cho tất cả các đối tượng trong class đó, không cần phải khai báo từng đối tượng một có biến đó, để rút gọn code và dung lượng. |
| toString() | - dùng để in ra thông tin cần in của một đối tượng, nếu không có phương thức toString() thì java sẽ hiểu là in ra địa chỉ bộ nhớ của đối tượng đó, còn nếu có thì sẽ in ra thông tin của đối tượng |
| Getter và setter | -Việc sử dụng Setter và Getter cần thiết trong việc kiểm soát những thuộc tính quan trọng mà ta thường được sử dụng và yêu cầu giá trị chính xác. Khi dùng getter và setter thì thuộc tính nên để private.  - getter : để lấy giá trị của thuộc tính đối tượng ra để xem.    - setter : để sửa đổi giá trị của thuộc tính của đối tượng ra sửa lại. |
| Các biến trong đối tượng |  |
| Static trong method | - để nó có thể tồn tại độc lập với các đối tượng và có thể được gọi thẳng từ lớp mà không cần đến một tham chiếu đối tượng nào.  - Không được gắn với một đối tượng nào, nên các phương thức static của một lớp chạy mà không biết một chút gì về bất cứ đối tượng cụ thể nào của lớp đó. |
|  |  |