|  |  |
| --- | --- |
| **1. Từ khóa static dùng để làm gì (mục đích)?** | - Từ khóa static sử dụng để khai báo các thuộc tính và phương thức của lớp (khác với thuộc tính và phương thức của đối tượng)  - Các thành phần static trực thuộc lớp , thay vì trực thuộc đối tượng  - Biến static còn được gọi là biến của lớp (class variable)  - Phương thức static còn được gọi là phương thức của lớp (class method)  - Có thể truy xuất các thành phần static bằng các sử dụng lớp hoặc đối tượng  - Không cần khởi tạo đối tượng vẫn có thể sử dụng các thành phần static  - Mục đích:  + Tiếc kiệm bộ nhớ  + Truy cập mà không cần tạo đối tượng |
| **2. Cách sử dụng biến static , method static , khối static , static class (tìm hiểu thêm )** | Biến static :  - Khai báo: static <kiểu\_dữ\_liệu><tên\_biến>;  - Sử dụng khi :  + Khi một giá trị là chung cho tất cả các đối tượng  + Khi bạn muốn truy cập nhanh mà không cần tọa object  Method static :  - Là phương thức không cần tạo đối tượng đẻ sử dụng . Nó thuộc về class , không phải object .  Cú pháp : modifer static data \_type method\_name(){  // body  }  Sử dụng khi :  + Khi hàm không phụ thuộc vào dữ liệu đối tượng  Khối static:  + Là một khối lệnh đặc biệt , dùng để khởi tạo dữ liệu tĩnh(static)  + Khối static được chạy duy nhất một lần , khi lớp được nạp (load) vào bộ nhớ , trước cả main()  + Cách khai báo : static{  // các câu lệnh  }  - Sử dụng khi :  + Khi bạn cần khởi tạo biến static phức tạp  + Khi bạn cần chạy đoạn code chỉ một lần lúc chương trình bắt đầu  Static class :  - Trong java , chỉ có lớp lồng nhau (nested class) mới có thể được khai báo là static.  - Static nested class là một lớp được khai báo beeb trong một lớp khác và có từ khóa static  - Nó không phụ thuộc vào object bên ngoài  - Kêt luận :  + Static class trong java bằng lớp con static nằm trong lớp khác  + Dùng khi bạn muốn tách logic độc lập , không cần ràng buộc với object của lớp ngoài  + Giúp tổ chức code gọn gàng hơn , nhất là khi viết hoặc cấu trúc dữ liệu |
| **3. Ràng buộc static (cách sử dụng)?** | - Phương thức static chỉ có thể gọi các phương thức static khác  - Phương thức static chỉ có thể truy xuất các biến static  - Phương thức static không thể sử dụng từ khóa this hoặc super  - Có thể khởi tạo biến static thông qua khối khởi tạo static |
| **4. Các loại biến trong java? Các biến được lưu ở đâu trong vùng nhớ ?** | |  |  | | --- | --- | | **Loại biến** | **Lưu ở vùng nhớ** | | Biến cục bộ (Local variable) | Stack | | Biến đối tượng (instance variable) | Heap | | Biến lớp (Final variable) | Method Area | | Final variable | Tùy loại  Final cục bộ → Stack Final static → Method Area | |
| **5. Access modifier là gì , được sử dụng ở đâu?** | Access modifier là các từ khóa được sử dụng để quy định mức độ truy cập đến lớp và các thành phần của lớp  Access modifier được sử dụng ở :  + Class(bên ngoài)  + Inner class (class bên trong class)  + Biến (fields)  + Phương thức (methods)  + Contructor |
| **6. Các loại access modifier, phân biệt ?** | - Public: có thể truy cập bất cứ đâu  - Private : các phương thức và thuộc tính chỉ được phép truy xuất trong cùng một lớp  - Protected : các phương thức và thuộc tính được phép truy xuất trong cùng một lớp và ở các lớp con (kế thừa)  - Default : Nếu không có access modifier thì mức default sẽ được áp dụng . Lớp và các thành phần của lớp được truy xuất ở những nơi trong cùng một package |
| **7. Trình bày tính bao đóng , mục đích ? làm sao có thể tạo được tính bao đóng ?** | - Tính bao đóng là cơ chế giấu dữ liệu bên trong đối tượng , không cho bên ngoài truy cập trực tiếp mà phải thông qua các phươn thức (getter và setter)  Mục đích:  + Bảo vệ dư liệu  + Kiểm soát truy cập  + Dễ bảo trì code  + Tăng tính bảo mật  + Ẩn chi tiết triển khai  Tạo tính bao đóng :  Private + getter/setter  1. khai báo biến là private  2. Tạo getter/setter public để truy cập gián tiếp |