Java là gì?

Java là ngôn ngữ lập trình hướng đối tượng

Java là ngôn ngữ lập trình hướng đối tượng

Java là ngôn ngữ lập trình hướng đối tượng

Java la ngôn ngữ lập trình hướng đói tương

Java la ngon ngu lap trinh huong doi uong

Java là ngôn ngữ lập trình hướng đối tương

JDK, JRE, JVM

JDK.

Là công cụ hỗ trợ lập trình viên phát triển ứng dụng java

JDK . là công cụ hỗ trợ lập trình viên phát triển ứngd dụng java

JDK. Là công cụ hỗ trợ lập trình viên phát triển ứng dụng java

JDK. Là công cụ hỗ trợ lập trình viên phát triêtn ứng dụng java

JDK. La cong cu ho tro lap trinh vien phat trien ung dung java

JDK là công cụ hỗ trợ lập trình viên phát triên ứng dụng java

JDK. Là công cụ hỗ trợ lập trình viên phát trienr ứng dụng java

JRE.

JRE. Môi trường thực thi ứng dụng java

JRE. Môi trương thực thi ứng dụng java

JRE. Môi trương thực thi ứng dụng java

JRE. Môi trương thưc thi ưng dung java

JRE. Môi truong thuc thi ung dung java

JRE. Môi trường thực thi ứng dung java

JVM.

JVM. Máy ảo java thực thi mã byte code.class ra ngôn ngữ máy

JVM.máy ảo java thực thi mã byte code.class ra ngôn ngữ máy

JVM. Máy ảo java thực thi mã byte code.class ra ngôn ngữ máy

JVM. Máy ảo java thực thi mã byte code.class ra ngôn ngữ máy

JVM. Máy ảo java thực thi mã byte code.class ra ngôn ngư máy

JVM. Máy ảo java thực thi mã byte code.class ra ngôn ngữ máy

Thông dịch biên dịch khác nhau như thế nào?

Thôg dịch

Chạy tới đâu dịch tới đó

Biên dịch

Dịch toàn bộ file một lần sau đó sử dụng kết quả biên dịch mà không cần dịch lại

THông dịch

Chạy tới đâu dịch tới đó

Biên dịch

Dịch toàn bọ file một lần sau đó sử dụng kết qua biên dịch mà không cần dịch lại

Thông dịch

Chạy tới đâu dịch tới đó

Biên dịch

Dịch toàn bộ file một lầm sau đó sử dụng kết quả biên dịch mà khôngcần dịch lại

THông dịch chạy tới đâu dịch tới đó

Biên dịch Dịch toàn bộ file một lần sau đó sử dụng kết quả biên dịch mà không cần dịch lại

Thông dịch chạy tới đâu dịch tới đó

Biên dịch dịch toàn bộ file một lần sau đó sử dụng kết quả biên dịch mà không cần dịch lại

Java là ngôn ngữ thông dịch hay biên dịch?

Java là ngôn ngư đa nền tảng và có thể biên dịch hoặc thông dịch tùy theo cách thực thi của nó

Java là ngôn ngư đa nền tảng và có thể biên dịch hoặc thông dịch tùy theo cách thực thi của nó

Java là ngôn ngư đa nền tảng và có thể biên dịch hoặc thông dihc tùy theo cách thực thi của nó

Java là ngôn ngư đa nên tảng và có thể biên dịch hoặc thông dịch tùy theo cách thực thi của nó

Java là ngôn ngư đa nên tang có thể biên dịch hoặc thông dịch tùy theo cách thực thi của nó

Java là ngôn ngư đa nên tảng có thể biên dịch hoặc thông dịc tùy theo cách thực thi của nó

Các kiểu dữ liệu trong java và giá trị mặc định khi khai báo

|  |  |
| --- | --- |
| Kiểu dữ liệu | Giá trị mặc định |
| Boolean(true và false) | False |
| Char(dùng để lưu trữ kí tự có kích cỡ 2 byte) | ‘\u0000’ |
| Byte(dùng để lưu trữ kiểu số nguyên có 1 byte (8bit)) | 0 |
| Short(dùng để lưu dữ kiểu số nguyên có 2 byte (16bit)) | 0 |
| Int(dùng để lưu trữ kiểu số nguyên có 4 byte (32bit)) | 0 |
| Long(dùng để lưu trữ kiểu số nguyên có 8 byte (64 bit)) | 0 |
| Float(dùng để lưu trữ kiểu số thực có 4 byte ( 32 bit)) | 0.0f |
| Double(dùng để lưu trữ kiểu số thực có 8 byte (64 bit)) | 0.0d |

|  |  |
| --- | --- |
| Kiểu dữ liệu | Giá trị mặc định |
| Boolean(true và false) | False |
| Char(dùng để lưu trữ kí tự có kích cỡ 2byte) | ‘\u0000\ |
| Byte(dùng để lưu trữ sô nguyên có kích cỡ 1 byte(8bit) | 0 |
| Short 2byte 16 bit |  |
| Int 4 byte 32 bit |  |
| Long 8 byte 62 bit |  |
| Float dùng để lưu trữ số thực kích cỡ 4 byte (32 bit) |  |
| Double 8 byte ( 64 bit) |  |

|  |  |
| --- | --- |
| Kiểu dữ liệu | Giá trị mặc định |
| Boolean(true và false) | False |
| Char(dùng để lưu trữ kí tự có kích cỡ 2 byte) | ‘\u0000’ |
| Byte (dùng để lưu trữ số nguyên có kích cỡ 1 byte(8 bit)) |  |
| Short 2 byte 16 bit |  |
| Int 4 byte 32 bit |  |
| Long 8 byte 64 bit |  |
| Float dùng để lưu trữ số thực có kích cỡ 4 byte (32 bit) | 0.0f |
| Double 8 byte (64 bit) | 0.0d |

|  |  |
| --- | --- |
| Kiểu dữ liệu | Giá trị mặc định |
| Boolean (true và false) | false |
| Char dùng để lưu trữ kiểu kí tự có kích cỡ 2 byte |  |
| Byte dùng để lưu trữ sô nguyên có kích cỡ 1 byte 8 it |  |
| Short |  |
| Int |  |
| Long |  |
| Float dùng để lưu trữ số thực có kích cỡ 4 byte 32 bit |  |
| Double 8 byte 64 bit |  |

Java platform là gì?

Là môi trường thực thi đa nền tảng cho các ứng dụng java, được thiết kế hoạt động trên nhiều hệ điều hành và kiến trúc phần cứng

Là môi trường thực thi đa nền tảng cho các ứng dụng java, được thiết kế hoạt động trên nhiều hệ điều hành và kiến trúc phần cứng

Là môi trường thực thi đa nền tảng cho các ứng dụng java, được thiết kế hoạt động trên nhiều hệ điều hành và kiến trúc phần cứng.

Là môi trường thực thi đa nền tảng cho các ứng dụng java, được thiết kế hoạt động trên nhiều hệ điều hành và kiến trúc phần cứng.

Là môi trường thực thi đa nền tảng cho các ứng dụng java, được thiết kế hoạt động trên nhiều hệ điều hành và kiến trúc phần cứng

La mo I truong thuc thi da nen tang cho cac ung dung java, duoc thiet ke hoat dong tren nhieu he dieu hanh va kien truc phan cung

Là môi trường thực thi đa nền tảng cho các ứng dụng java, được thiết kế hoạt động trên nhiều hệ điều hành và kiến trúc phần cứng

Là môi trường thực thi đa nền tảng cho các ứng dụng java, được thiết kế hoạt động trên nhiều hệ điều hành và kiến trúc phần cứng

Sự khác nhau giữa bộ nhớ stack và heap trong java.

|  |  |
| --- | --- |
| Heap | Stack |
| Là một khu vực lưu trữ bộ nhớ được chia sẽ giữa các luồng ứng dụng trong java, để lưu trữ các đối tượng được tạo trong quá trình chạy. | Là một khu vực lưu trữ bộ nhớ được chia sẽ để lưu trữ các biến cục bộ và tham số trong các phương thức và khối lệnh |
| Heap được quản lý bởi bộ thu gom rác để giải phóng bộ nhớ không sử dụng | Stack tự giải phóng bộ nhớ khi phương thức thanh toán |
| Khi tạo đối tượng, bộ nhớ được cấp phát trên Heap | Khi tạo đối tượng các biến cục bộ và tham số được lưu trên stack |

Sự khác nhau giữa bộ nhớ heap và stack trong java

|  |  |
| --- | --- |
| Heap | Stack |
| Là một khu vực lưu trữ bộ nhớ được chia sẽ giữa các luồng ứng dụng trong java, để lưu trữ các đối tượng được tạo trong quá trình chạy | Là một khu vực lưu trữ bộ nhớ được chia sẽ để lưu trữ các biến cục bộ và tham số trong các phương thức và khối lệnh |
| Heap được quản lý bởi bộ thu gom rác để giải phóng bộ nhớ không sử dụng | Stack tự giải phóng bộ nhớ khi phương thức thanh toán |
| Khi tạo đối tượng, bộ nhớ được cấp phát trên heap | Khi tạo đối tượng các biến cục bộ và tham số được lưu trữ trên stack |

|  |  |
| --- | --- |
| Heap | Stack |
| Là một khu vực lưu trữ bộ nhớ được chia sẽ giữa các luồng ứng dụng trong java, để lưu trữ các đối tượng trong quá trình chạy | Là một khu vực lưu trữ bộ nhớ được chia sẽ để lưu trữ các biến cục bộ và tham số trong các phương thức và khối lệnh |
| Được quản lý bởi bộ thu gom rác để giải phóng bộ nhớ không sử dụng | Tự giải phóng bộ nhớ khi phương thức thanh toán |
| Khi tạo đối tượng , bộ nhớ được cấp phát trên heap | Khi tạo đối tượng các biến cục bộ và tham số được lưu trữ trên stack |

Trình bài các cách khởi tạo giá trị một mảng trong java?

1. Khởi tạo mảng có kích thước cố định
2. Khởi tạo mảng có giá trị ban đầu

Phần tử của mảng có thể dùng kiểu dữ liệu nào và có giá trị mặc định là gi?

1. Mảng trong java có thể dùng kiểu nguyên thủy và đối tượng
2. Giá trị mặc định của phần tử trong mảng tùy thuộc vào kiểu dữ liệu của phần tử đó

OOP là gì

Oop (object oriented progaming)

Là ngôn ngữ lập trình hướng đối tượng

Là kỹ thuật cho phép lập trình viên ánh xạ những đối tượng từ bên ngoài thực tế vào ngôn ngữ lập trình

Phân biệt class và object

Class:

* Là một bản thiết kế tạo ra đối tượng
* Class định nghĩa các thuộc tính và phương thức của đối tượng

Object:

* Đối tượng là thể hiện cụ thể của class

Class:

* Là một bản thiết kế tạo ra đối tượng
* Class định nghĩa các thuộc tính và phương thức của đối tượng

Object:

* Đối tượng là thể hiện cụ thể của class

Class:

* Làmột bản thiết kế tảoa đối tượng
* Class định nghĩa các thuộc tính và phươgn thức của đôi tượng

Object:

* Dốid tượng kà thể hiện cụ thể của class

Class:

* Là một bản thiết kế tạo ra đối tượng
* Class định nghĩa các thuộc tính và phương thức của đối tượng

Object:

* Đối tượng là thể hiện cụ thể của class

Class:

* Là một bản thiết kế tạo ra đối tượng
* Class định nghĩa các thuộc tính và phương thức của đối tượng

Object:

* Đối tượng là thể hiện cụ thể của class

Contructor là gì?

Là một phương thức khởi tạo đối tượng

Đuọc gọi tại thời điểm khởi tạo đối tượng nó khởi tạo các giá trị để cung cấp dữ liệu cho đối tượng

Là một phương thức khởi tạo đối tượng

Được gọi tại thời điểm khởi tạo đối tượng nó khởi tạo các giá trị để cung cấp dữ liệu cho đối tượng

Là một phương thức khởi tạo đối tượng

Được gọi tại thời điểm khơi tạo đối tượng nó khởi tạo các giá trị để cung cấp dữ liệu cho đối tượng

Cách khai báo contructor và đặc điểm contructor trong java?

Tên contructor phải trùng khớp với tên lớp không có kiểu trả về và không có dữ liệu trả về

Contructor có thể có hoặc không có tham số

Mặc định nếu không khai báo contructor thì java sẽ cung cấp contructor mặc định không có tham số

Trong 1 class thì có thể có nhiều contructor

Có thể gọi contructor khác thông qua từ khóa this và lưu ý this được gọi đầu tiên trong nội dung của phương thức khởi tạo

Tên contructor phải trùng vơi tên lớp không có kiểu trả về và không có dữ liệu trả về

Contructor có thể có hoặc không có tham số

Mặc định nếu k khai báo contructor thì java sẽ cung cấp contructor mặc định không có tham số

Trong 1 class thì có thể có nhiều contructor

Có thể gọi contructor thông qua từ khóa this lưu ý khi gọi this thì được gọi đầu tiên trong nội dung của phương thức khởi tạo

Tên contructor phải trùng khớp với tên lớp không có kiểu trả về và không có dữ liệu trả về

Contructor có thể có hoặc không có tham số

Mặc định nếu không khai báo contructor thì java sẽ cung cấp contructor mặc định không có tham số

Trong 1 class thfi có thể có nhiều contructor

Có thể gọi contructor khác thông qua từ khóa this và lưu ý khi gọi this thì được gọi đầu tiên trong nội dung của phương thức khơi tạo

Tên contructor phải trùng khớp với tên class không có kiểu trả về và không có dữ liệu trả về

Controctor có thể có hoặc k có tham số

Mặc định nếu không khai báo contructor thì java sẽ cung cấp contructor mặc định không có tham số

Trong 1 class thì có thể có nhiều contructor

Có thể gọi contructor khác thông qua từ khóa this và lưu ý khi gọi this thì dược gọi đầu tiền trong nội dung của phương thức khởi tạo

Tên contructor phải trùng khớp với tên class không có kiểu trả về và không có dữ liệu trả về

Contructor có thể có hoặc không có tham số

Mặc định nếu không khai báo contructor thì java sẽ cung cấp contructor mặc định không có tham số

Trong 1 class có thể có nhiều contructor

Có thể gọi contructor thông qua từ khóa this và lưu ý khi gọi this thì được gọi đầu tiên trong nội dung của phương thức khởi tạo

Phân biệt contructor và method

|  |  |  |
| --- | --- | --- |
|  | Contructor | Method |
| Kiểu trả về | K có kiểu trả về | Có hoặc không có kiểu trả về |
| Đặt tên | Trùng tên với class | Có thể giống hoặc khác tên với class |
| Biên dịch | Nếu không khai báo thì java sẽ cung cấp một contructor mặc định không có tham số | Java không có cung cấp method mặc định |
| Số lần gọi | Chỉ được gọi 1 lần khi khởi tạo đối tượng | Có thể gọi được nhiều lần |

Phân biệt contructor và method

|  |  |  |
| --- | --- | --- |
|  | Contructor | Method |
| Kiểu trả về | Không có kiểu trả về | Có hoặc k có kiểu trả về |
| Đặt tên | Trùng tên với class | Trùng tên hoặc khác tên với class |
| Biên dịch | Nếu không khai báo thì java sẽ cung cấp một contructor mặc định không có tham số | Java không có cung cấp method mặc định |
| Số lần gọi | Chỉ được gọi 1 lần khi khởi tạo đối tương | Có thể gọi nhiều lần |

Phân biệt contructor và method

|  |  |  |
| --- | --- | --- |
|  | Contructor | Method |
| Kiểu trả về | Không có kiểu trả về | Có hoặc k có kiểu trả về |
| Đặt tên | Trùng tên với class | Trùng tên với class hoặc khác tên với class |
| Biên dịch | Nếu không khai báo contructor thì java sẽ cung cấp một contructor mặc định không có tham số | Java không cung cấp method mặc định |
| Số lần gọi | Chỉ được gọi 1 lần khi khởi tạo đối tượng | Có thể gọi nhiều lần |

Tính bao đóng là gì? Làm sao để thu được tính bao đóng trong java?

Tính bao đóng:

Là kỹ thuật ẩn dấu đi thông tin của đối tượng, chỉ hiển thị thông tin cần thiết của đối tượng

Hình ảnh về tính bao đóng.

* Access modifier. Private, public, default,…
* Get, set
* Mục đích: cho phép kiểm soát được quyền truy cập vào thuộc tính private

Kiểm soát giá trị được gán cho hệ thống

Tính bao đóng: là kỹ thuật ẩn dấu đi thông tin của đối tượng, chỉ hiển thị thông tin cần thiết của đối tượng

Hình ảnh về tính bao đóng

* Access modifier. Private, public, default
* Get, set
* Mục đích: cho phép kiểm soát được quyền truy cập vào thuộc tính private

Kiểm soát giá trị được gán cho hệ thống

Tính bao đóng: là kỹ thuật ẩn dấu đi thông tin của đối tượng, chỉ hiển thị thông tin cần thiết của đối tượng

Hình ảnh về tính bao đóng

* Access modifier, private, public, default
* Get, set
* Mục đích: cho phép kiểm soát được quyền truy cập vào thuộc tính private

Kiểm soát giá trị được gán cho hệ thống

Tham trị, tham chiếu

Biến tham trị

* Lưu trữ giá trị trực tiếp của dữ liệu
* Được truyền vào phương thức dưới dạng giá trị của biến đó không ảnh hưởng đến biến gốc bên ngoài phương thức

Biến tham chiếu

* Lưu trữ địa chỉ vùng nhớ chứa dữ liệu
* Được truyền vào phương thức dưới dạng địa chỉ của vùng nhớ chứa dữ liệu và thay đổi dữ liệu trong phương thức cũng ảnh hưởng đến biến gốc bên ngoài phương thức

Tham trị, tham chiếu

Biến tham trị

* Lưu trữ giá trị trực tiếp của dữ liệu
* Được truyền vào phương thức dưới dạng giá trị của biến đó không ảnh hướng đến biến gốc bên ngoài phương thức

Biến tham chiếu

* Lưu trữ địa chỉ vùng nhớ chứa dữ liệu
* Được truyền vào phương thức dưới dạng địa chỉ vùng nhớ chứa dữ liệu và thay đổi dữ liệu trong phương thức cũng ảnh hướng đến biến gốc bên ngoài phương thức

Biến tham trị

* Lưu trữ giá trị trực tiếp của dữ liệu
* Được truyền vào phương thức dưới dạng giá trị của biến đó không ảnh hưởng đến biến gốc bên ngoài phương thức

Biến tham chiếu

* Lưu trữ địa chỉ vùng nhớ chứa dữ liệu
* Được truyền vào phương thức dưới dạng địa chỉ vùng nhớ chứa dữ liệu và thay đổi dữ liệu trong phương thức cũng ảnh hưởng đến biến gốc bên ngoài phương thức

Từ khóa static dùng để làm gì?

Static sử dụng để định nghĩa các biến và phương thức trong một lớp hoặc một hàm

Biến được khai báo static là các biến tồn tại trong suốt vòng đời của chương trình

Các phương thức được khai báo bằng static thì có thể gọi trực tiếp từ lớp mà không cần tạo đối tượng

Static dùng để quản lý bộ nhớ của lớp một cách có hệ thống và gọn hơn

Static sử dụng để định nghĩa các biến và phương thức trong một lớp hoặc một hàm

Biến được khai báo static là các biến tồn tại trong suốt vòng đời của chương trình

Các phương thức được khai báo bằng static thì có thể gọi trực tiếp từ lớp mà không cần tạo đói tượng

Static dùng để quản lý bộ nhớ của lớp một cách có hệ thống và gọn hơn

Stactic sử dụng để định nghĩa các biến và phương thức trong một lớp hoặc một hàm

Biến được khai báo static là các biến tồn tại trong suốt vòng đời của chương trình

Các phương thức được khai báo bằng static thì có thể gọi trực tiếp từ lớp mà không cần khởi tao đối tượng

Static dùng để quản lý bộ nhớ của lớp một cách có hệ thống và gọn hơn

Static sử dụng để định nghĩa các biến và phương thức trong một lớp hoặc một hàm

Biến được khai báo static là các biến tồn tại trong suốt vòng đời của chương trình

Các phương thức được khai báo static thì có thể gọi trực tiếp từ lớp mà không cần khởi tạo đối tượng

Static dùng để quản lý bộ nhớ của lớp một cách có hệ thống và gọn hơn

Ràng buộc khi sử dụng static

Phương thức static chỉ có thể gọi phương thức static khác

Dùng phương thức thì có thể gọi phương thức biến static

Dùng khối static để khởi tạo giá trị cho biến static

Phương thức static chỉ có thể gọi phương thức static khác

Dùng phương thức thì có thể gọi phương thức biến static

Dùng khối static để khởi tạo giá trị cho biến static

Phương thức static chỉ có thể gọi phương thức static khác

Dùng phương thức thì có thể gọi phương thức biến static

Dùng khối static để khởi tạo giá trị cho biến static

Các loại biến trong java?

Biến local (biến cục bộ)

* Biến khai báo trong method, contructor, block
* Không dùng được access modifier và cần khởi tạo giá trị mặc định
* Biến local được khởi tạo khi method, contructor, block thực hiện chức năng và bị hủy khi method, contructor, block hoàn thành xong chức năng

Biến instance (biến toàn cục)

* Biến được khai báo trong lớp, ngoài contructor, method, block
* Biến cục bộ được tạo khi mà đối tượng được tạo bởi từ khóa new và hủy khi đối tượng hủy

Biến static

* Khai báo bằng từ static
* Biến static được tạo khi chương trình bắt đầu chạy và hủy khi chương trình dừng

Biến local (biến cục bộ)

* Biến khai báo trong method, contructor, block
* Không dùng được access modifier và cần khởi tạo giá trị mặc định
* Biến local được khởi tạo khi method, contructor, block thực hiện chức năng và bị hủy khi method, contructor, block hoàn thành xong chức năng

Biến instance (biến toàn cục)

* Biến được khai báo trong lớp, ngoài contructor, method, block
* Biến cục bộ được tạo khi mà đối tượng được tạo bởi từ khóa new và hủy khi đối tượng hủy

Biến static

* Khai báo bằng từ khóa static
* Biến static được tạo khi chương trình bắt đầu chạy và hủy khi chương trình dừng

Biến local (biến cục bộ)

* Biến khai báo trong method, contructor, block
* Không dùng được access modifier và cần khởi tạo giá trị mặc định
* Biến local khởi tạo khi method, contructor, block thực hiện chức năng và bị hủy khi method, contructor, block hoàn thành xong chức năng

Biến instance (biến toàn cục)

* Được khi báo trong class, ngoài contructor, method, block
* Biến cục bộ được tạo khi mà đối tượng được tạo bởi từ khóa new và hủy khi đối tượng hủy

Biến static

* Khai báo bằng từ khóa static
* Biến static được tạo khi chương trình bắt đầu chạy và hủy khi dừng chương trình

Trình bày các loại access modifier và phạm vi truy cập

Private: trong class

Default: Trong class, trong package, ngoài package nếu class có mối quan hệ kế thừa

Public: Toàn bộ project có thể truy cập được

Private: trong class

Default: trong class, trong package, ngoài package nếu class có mối quan hệ kế thừa

Public: toàn bộ project có thể truy cập được

Kế thừa trong java?

Là cho phép lớp con sử dụng lại đặc điểm và hành vi của lớp cha

Mục đích: tái sử dụng code

Java không hỗ trợ đa kế thừa

Lớp con kế thừa được những tài sản nào của lớp cha

Lớp con có thể kế thừa tất cả các thành phần không phải là private của lớp cha

Có thể định nghĩa lại các phương thức của lớp cha trong lớp con để thay đổi hành vi của chúng

Lớp con có thể kế thừa tất cả các thành phần không phải private của lớp cha

Có thể định nghĩa lại các phương thức của lớp cha trong lớp con để thay đổi hành vi của chúng

Lớp object là gì?

Lớp object là lớp gốc của hệ thống phân cấp lớp trong java

Mọi lớp đều là lớp con của object

Lớp object cung cấp các phương thức dùng chung cho tất cả các đối tượng

Lớp object là lớp gốc của hệ thống phân cấp lớp trong java

Mọi lớp đều là lớp con của object

Lớp object cung cấp các phương thức dùng chung cho tất cả các đối tượng

Lớp object là lớp gốc của hệ thống phân cấp lớp trong java

Mọi lớp đều là lớp con của object

Lớp object cung cấp các phương thức dùng chung cho tất cả các đối tượng

Lớp object là lớp gốc của hệ thống phân cấp lớp trong java

Mọi lớp đều là lớp con của object

Lớp object cung cấp các phương thức dùng chung cho tất cả các đối tượng

Khái niệm đa hình

Là khả năng một đối tượng có thể hiện hành vi theo nhiều cách khác nhau phụ thuộc vào ngữ cảnh

Là khả năng một đối tượng có thể hiện hành vi theo nhiều cách khác nhau tùy thuộc vào ngữ cảnh

Là khả năng một đối tượng có thể hiện hành vi theo nhiều cách khác nhau tùy thuộc vào ngữ cảnh

Phân biệt overloading và overriding

|  |  |
| --- | --- |
| overloading | overriding |
| Thể hiện tính đa hình tại compile time | Thế hiện tính đa hình tại runtime |
| Thêm hành vi cho phương thức | Thay đổi hành vi hiện tại của phương thức |
| Có thể khác nhau về số lượng và kiểu dữ liệu tham số | Số lượng và kiểu dữ liệu của tham số phải giống nhau |
| Xảy ra trong cùng 1 class | Xảy ra ở 2 class có quan hệ kế thừa |

|  |  |
| --- | --- |
| overloading | overriding |
| Thể hiện tính đa hình tại compile time | Thể hiện tính đa hình tại run time |
| Thêm hành vi cho phương thức | Thay đổi hành vi hiện tại cho phương thức |
| Có thể khác nhau về số lượng và kiểu dữ liệu tham số | Số lượng và kiểu dữ liệu của tham số phải giống nhau |
| Xảy ra trong cùng 1 class | Xảy ra ở 2 class có quan hệ kế thừa |

|  |  |
| --- | --- |
| overloading | overriding |
| Thể hiện tính đa hình tại compile time | Thể hiện tính đa hình tại run time |
| Thêm hành vi cho phương thức | Thay đổi hành vi hiện tại cho phương thức |
| Có thể khác nhau về số lượng và kiểu dữ liệu tham số | Số lượng và kiểu dữ liệu của tham số phải giống nhau |
| Xảy ra trong cùng 1 class | Xảy ra ở 2 class có quan hệ kế thừa |