Bài 3

|  |  |
| --- | --- |
| Câu hỏi | Trả lời |
| Câu 1 : Nhắc lại Method là gì? Mục đích sử dụng method? | * **Method** là tập hợp các câu lệnh để thực hiện một nhiệm vụ cụ thể. Nó giúp chia nhỏ chương trình, tái sử dụng mã, tăng tính tổ chức và dễ bảo trì. * **Mục đích :**   **+** Tái sử dụng mã: Viết một lần, dùng nhiều lần.  **+** Tăng tính tổ chức: Chia chương trình thành nhiều phần nhỏ dễ quản lý.  + Giảm lỗi, dễ bảo trì và kiểm thử. |
| Câu 2 : Cú pháp tạo method trong java? Cách gọi method trong java? | <access\_modifier> <return\_type> <method\_name>(<parameters>) {  }   * Ví dụ :   public int add(int a, int b) {  return a + b;  }   * Cách gọi :   + Gọi từ cùng class ( static )  int result = add(5, 3);  + Gọi từ object ( non – static )  MyClass obj = new MyClass();  int result = obj.add(5, 3); |
| Câu 3 : Giải thích các thành phần trong một method? | * Access modifier : Phạm bi truy cập ( public , private....) * Return type : Kiểu trả về ( int , void , String ..... ) * Method name : Tên method * Parameters : Danh sách tham số * Method body : khối lệnh thực thi |
| Câu 4 : Các loại method? | * Loại method : mô tả * Static method : gọi trực tiếp qua class * Instance method : gọi qua object * Getter/Setter : để truy cập và cập nhật thuộc tính * Constructor method : dùng để khởi tạo đối tượng |
| Câu 5 : Trong một class có thể tạo được bao nhiêu method cùng tên? Điều kiện tạo được nhiều method cùng tê ? | * Có thể tạo nhiều method cùng tên , gọi là method overloading * Điều kiện : khác nhau danh sách tham số ( số lượng , kiểu dữ liệu hoặc thứ tự ) * Ví dụ :   void show() {}  void show(int a) {}  void show(String s, int a) {} |
| Câu 6 : Tìm hiểu bộ nhớ stack và heap trong java? dữ liệu nào sẽ lưu ở stack, dữ liệu nào sẽ lưu ở heap? | * Stack   + Dữ liệu lưu trữ : biến cục bộ , lời gọi method  + Quản lý : tự động, theo nguyên tắc LIFO  + Tốc độ : nhanh hơn  + Thời gian sống : Ngắn ( kết thúc khi method kết thúc )   * Heap   + Dữ liệu lưu trữ : Object được tạo bằng new , array  + Quản lý : do JVM quản lý bằng Garbage Collector  + Tốc độ : chậm hơn  + Thời gian sống : lâu ( tồn tại đến khi không dùng nữa )   * Ví dụ :   public class Demo {  public static void main(String[] args) {  int x = 10; // x nằm ở Stack  Person p = new Person(); // p ở Stack, object Person ở Heap  }  } |
| Câu 7 : Tìm hiểu tham trị và tham chiếu trong java? | * Tham trị ( pass by value ) : là cách truyền bản sao của giá trị biến và method * Mọi thay đổi trong method không ảnh hưởng đến biến gốc * Tham chiếu ( Pass by Reference ) * Thay đổi trong method ảnh hướng trực tiếp đến dữ liệu gốc |