

BỘ XÂY DỰNG
TRƯỜNG ĐHXD MIỀN TÂY

ĐÁP ÁN – THANG ĐIỂM
KỲ THI KTHP HỌC KỲ II NĂM HỌC 2024-2025

Trình độ: ĐẠI HỌC; Ngày thi: 20/5/2025

Môn: LẬP TRÌNH JAVA

ĐÁP ÁN ĐỀ THI CHÍNH THỨC

(Đáp án - thang điểm gồm 04 trang)

Câu	Phần	Nội dung	Thang điểm
I		TRẮC NGHIỆM	5.0
1	C	Tên tập tin Java có thể là Hello.java	0.25
2	A	2yVar	0.25
3	C	noVir	0.25
4	D	% Là toán tử Chia	0.25
5	C	Overriding chỉ xảy ra giữa lớp cha và lớp con.	0.25
6	D	int[] myArray1 = [9,9,9];	0.25
7	C	arrayName.length	0.25
8	D	floor() trả về một giá trị gần nhất bằng một số nguyên.	0.25
9	A	8 7	0.25
10	C	8 -7	0.25
11	A	a có giá trị là: 1	0.25
12	C	2	0.25
13	A	compile failure	0.25
14	D	for(int n=0;n<a.length; n++)	0.25
15	C	Phát biểu “super” có thể dùng cho phương thức static.	0.25
16	D	Tất cả các đáp án đều đúng	0.25
17	C	Là thư viện chứa các thành phần phần mềm tạo sẵn cung cấp các chức năng cho chương trình Java	0.25
18	D	private class test	0.25
19	D	Nhập một chuỗi	0.25
20	B	Scanner	0.25
II		LẬP TRÌNH	5.0

1	Câu 1	Viết chương trình tính giá trị x lũy thừa n (x^n) với x là số thực và n là số nguyên tùy ý nhập từ bàn phím. Ví dụ: $2^4 = 16$; $2^{-3} = 1/2^3 = 1/8$	1.5
2	Câu 2	Viết chương trình sử dụng đệ quy để tính số Fibonacci thứ n. Dãy Fibonacci được định nghĩa như sau: $Fibonacci(0) = 0$; $Fibonacci(1) = 1$; $Fibonacci(n) = Fibonacci(n-1) + Fibonacci(n-2)$, với $n \geq 2$	1.5
3	Câu 3	Viết chương trình cộng hai ma trận số nguyên A và B có cùng kích thước m x n. <u>Yêu cầu:</u> Nhập số hàng m và cột n, Nhập giá trị cho ma trận A và B, Tính và in ra ma trận tổng $C = A + B$.	2.0
Tổng điểm			10.0đ

Đáp án (Lập Trình) – 5.0 ĐIỂM

Câu 1: (1.5 điểm)

Viết chương trình tính giá trị x lũy thừa n (x^n) với x là số thực và n là số nguyên tùy ý nhập từ bàn phím. Ví dụ: $2^4 = 16$; $2^{-3} = 1/2^3 = 1/8$

Yêu cầu: Viết hàm tính lũy thừa, Xử lý trường hợp đặc biệt khi $n = 0$ và Xử lý thêm các trường hợp với số mũ âm ($x^{-n} = 1 / x^n$).

```

1  import java.util.Scanner;
2
3  public class PowerRecursiveCalculator {
4      // Hàm đệ quy tính lũy thừa  $x^n$  (n có thể âm)
5      public static double powerRecursive(double x, int n) {
6          if (n == 0) return 1;
7          if (n < 0) return 1 / powerRecursive(x, -n);
8          return x * powerRecursive(x, n - 1);
9      }
10
11     public static void main(String[] args) {
12         Scanner sc = new Scanner(System.in);
13
14         System.out.print("Nhập cơ số x (số thực): ");
15         double x = sc.nextDouble();
16
17         System.out.print("Nhập số mũ n (số nguyên, có thể âm): ");
18         int n = sc.nextInt();
19
20         double result = powerRecursive(x, n);
21         System.out.printf("%.4f ^ %d = %.6f\n", x, n, result);
22
23         sc.close();
24     }
25 }

```

Câu 2: (1.5 điểm)

Viết chương trình sử dụng đệ quy để tính số Fibonacci thứ n. Dãy Fibonacci được định nghĩa như sau:

Fibonacci(0) = 0;

Fibonacci(1) = 1;

Fibonacci(n) = Fibonacci(n-1) + Fibonacci(n-2), với $n \geq 2$

Yêu cầu: Viết hàm đệ quy tính Fibonacci; Nhập vào một số nguyên dương n từ bàn phím; Xuất kết quả số Fibonacci thứ n.

```
1  import java.util.Scanner;
2
3  public class FibonacciRecursive {
4
5      public static int fibonacci(int n) {
6          if (n <= 1) return n;
7          return fibonacci(n - 1) + fibonacci(n - 2);
8      }
9
10     public static void main(String[] args) {
11         Scanner sc = new Scanner(System.in);
12         System.out.print("Nhập số nguyên dương n: ");
13         int n = sc.nextInt();
14
15         System.out.println("Số Fibonacci thứ " + n + " là: " + fibonacci(n));
16         sc.close();
17     }
18 }
```

Câu 3: (2.0 điểm)

Viết chương trình nhân hai ma trận A (m x p) và B (p x n) để thu được ma trận kết quả C (m x n).

```
1  import java.util.Scanner;
2  public class MatrixAddition {
3      public static void main(String[] args) {
4          Scanner sc = new Scanner(System.in);
5          System.out.print("Nhập số hàng: ");
6          int m = sc.nextInt();
7          System.out.print("Nhập số cột: ");
8          int n = sc.nextInt();
9
10         int[][] A = new int[m][n];
11         int[][] B = new int[m][n];
12         int[][] C = new int[m][n];
13         System.out.println("Nhập ma trận A:");
14         for (int i = 0; i < m; i++)
15             for (int j = 0; j < n; j++)
16                 A[i][j] = sc.nextInt();
17         System.out.println("Nhập ma trận B:");
18         for (int i = 0; i < m; i++)
19             for (int j = 0; j < n; j++)
20                 B[i][j] = sc.nextInt();
21         // Cộng
22         for (int i = 0; i < m; i++)
23             for (int j = 0; j < n; j++)
24                 C[i][j] = A[i][j] + B[i][j];
25
26         System.out.println("Ma trận tổng C:");
27         for (int[] row : C) {
28             for (int val : row)
29                 System.out.print(val + " ");
30             System.out.println();
31         }
32         sc.close();
33     }
34 }
```