

Đệ quy (Recursion)

- Một hàm đưa ra lời gọi đến chính nó được gọi là hàm đệ quy
- Java cũng như hầu hết ngôn ngữ lập trình bậc cao hỗ trợ hàm đệ quy, giúp lập trình tính toán thuận tiện trên các đối tượng dữ liệu có cấu trúc đệ quy

Đệ quy (Recursion)

- Một hàm đưa ra lời gọi đến chính nó được gọi là hàm đệ quy
- Java cũng như hầu hết ngôn ngữ lập trình bậc cao hỗ trợ hàm đệ quy, giúp lập trình tính toán thuận tiện trên các đối tượng dữ liệu có cấu trúc đệ quy

```
public class Main {  
    public static int f(int n){  
        if(n == 1) return 1;  
        return f(n-1) + n;  
    }  
  
    public static void main(String[] args) {  
        System.out.println(f(4));  
    }  
}
```

Đệ quy (Recursion)

- Một hàm đưa ra lời gọi đến chính nó được gọi là hàm đệ quy
- Java cũng như hầu hết ngôn ngữ lập trình bậc cao hỗ trợ hàm đệ quy, giúp lập trình tính toán thuận tiện trên các đối tượng dữ liệu có cấu trúc đệ quy

```
public class Main {  
    public static int f(int n){  
        if(n == 1) return 1;  
        return f(n-1) + n;  
    }  
  
    public static void main(String[] args) {  
        System.out.println(f(4));  
    }  
}
```



10

Đệ quy (Recursion)

- Một hàm đưa ra lời gọi đến chính nó được gọi là hàm đệ quy
- Java cũng như hầu hết ngôn ngữ lập trình bậc cao hỗ trợ hàm đệ quy, giúp lập trình tính toán thuận tiện trên các đối tượng dữ liệu có cấu trúc đệ quy

```
public class Main {  
    public static int fibonacci(int n){  
        if(n <= 1) return 1;  
        return fibonacci(n-1) + fibonacci(n-2);  
    }  
    public static void main(String[] args) {  
        for(int n = 1; n <= 10; n++)  
            System.out.print(fibonacci(n) + " ");  
    }  
}
```

Đệ quy (Recursion)

- Một hàm đưa ra lời gọi đến chính nó được gọi là hàm đệ quy
- Java cũng như hầu hết ngôn ngữ lập trình bậc cao hỗ trợ hàm đệ quy, giúp lập trình tính toán thuận tiện trên các đối tượng dữ liệu có cấu trúc đệ quy

```
public class Main {  
    public static int fibonacci(int n){  
        if(n <= 1) return 1;  
        return fibonacci(n-1) + fibonacci(n-2);  
    }  
    public static void main(String[] args) {  
        for(int n = 1; n <= 10; n++)  
            System.out.print(fibonacci(n) + " ");  
    }  
}
```



1 2 3 5 8 13 21 34 55 89