

BÀI 6. KIẾN THỨC NÂNG CAO VỀ LỚP VÀ PHƯƠNG THỨC

Yêu cầu chung:

- Tạo mới package: javacore.basic.lesson06.
- Tạo mới static class: Practice.java

1. Thêm mới một phương thức static có tên là sum() với tham số biến đổi kiểu double. Thực hiện viết code cho phương thức này để thực hiện tính tổng các tham số truyền vào.

▪ Hướng dẫn:

- Thêm phương thức static với tham số biến đổi
- Tên phương thức **getSum()**
- Tham số truyền vào: **double ...x**
- **public static final double sum(double ... x){}**

2. Bổ sung thêm 2 phương thức static để tìm số lớn nhất và nhỏ nhất từ tham số biến đổi.

▪ Hướng dẫn:

- Thêm 2 phương thức static với tham số biến đổi
- Tên phương thức: **getMin()**, **getMax()**
- Tham số truyền vào: **double ... x**
- **public static final double getMin(double ... x){}**
- **public static final double getMax(double ... x){}**

3. Bổ sung thêm phương thức static để chuyển đổi chuỗi ký tự thường thành chuỗi ký tự hoa. Ví dụ: “nguyen van a” sẽ được chuyển thành “NGUYEN VAN A”. Lưu ý không được sử dụng các thư viện, hàm có sẵn.

▪ Hướng dẫn:

- Thêm phương thức static có một tham số.
- Tên phương thức: **toUpper()**
- Tham số truyền vào: **String str**
- **public static String toUpper(String str){}**

4. Bổ sung thêm phương thức static để chuyển đổi chuỗi ký tự Hoa thành chuỗi ký tự thường. Ví dụ: “NGUYEN VAN A” sẽ được chuyển thành “nguyen van a”. Lưu ý không được sử dụng các thư viện, hàm có sẵn.

▪ **Hướng dẫn:**

- Thêm phương thức static có một tham số.
- Tên phương thức: **toLowerCase()**
- Tham số truyền vào: **String str**
- **public static String toLowerCase(String str){}**

5. Bổ sung thêm phương thức static để chuyển đổi các ký tự đầu tiên của một chuỗi sang ký tự hoa. Ví dụ: “nguyen van a” sẽ được chuyển thành “Nguyen Van A”. Lưu ý chỉ được sử dụng một số thư viện, hàm như phần hướng dẫn.

▪ **Hướng dẫn:**

- Thêm phương thức static có một tham số.
- Tên phương thức: **toUpperFirstChar()**
- Tham số truyền vào: **String str**
- Để tách chuỗi theo dấu cách (ký tự trắng), sử dụng phương thức **split(" ")**. Phương thức này sẽ trả về mảng các từ trong chuỗi ban đầu.
String[] words = str.split(" ");
- Để duyệt các từ trong mảng words và lấy ký tự đầu tiên của mỗi từ, sử dụng phương thức **charAt()**
char firstChar = words[i].charAt(0);
- Để chuyển ký tự vừa tìm được sang ký tự Hoa, sử dụng phương thức **toUpperCase()**
String upperFirstChar = String.valueOf(firstChar).toUpperCase().charAt(0);
- Thay đổi từ cũ bằng từ đã chuyển đổi ký tự đầu sang ký tự hoa, sử dụng việc nối 2 chuỗi giữa ký tự đầu tiên đã thành chữ Hoa và chuỗi còn lại.
words[i] = upperFirstChar + words[i].substring(1);
- Thực hiện nối mảng các từ thành chuỗi cuối cùng, sử dụng phương thức **String.join(words, " ")**
String result = String.join(" ", words);

6. Bổ sung thêm phương thức static để tính toán số [Fibonacci](#) thứ position. Lưu ý phải sử dụng đệ quy.
 - Hướng dẫn:
 - Thêm phương thức static có một tham số
 - Tên phương thức: **getFibonacci()**
 - Tham số truyền vào: **int position**
 - **public static long getFibonacci(int position){}**
7. Tạo mới class Solution.java. Bên trong có phương thức main(). Thực hiện gọi tất cả các phương thức của class Paractice.java
8. Sửa lại class Solution.java để có thể chạy được yêu cầu sau:
 - Dòng đầu tiên là số lượng Testcase T của bài toán ($1 \leq T \leq 100$)
 - Input được cho theo định dạng sau:
 - Mỗi testcase được cho trên 5 dòng với:
 - Dòng số 1 là số tự nhiên n ($5 \leq n \leq 1000$) cho biết lượng phần tử của mảng 1 chiều.
 - Dòng thứ 2 là n số thực tương ứng của mảng 1 chiều
 - Dòng thứ 3 là một chuỗi tối đa 1000 ký tự (str1)
 - Dòng thứ 4 là một chuỗi tối đa 1000 ký tự (str2)
 - Dòng thứ 5 là một số nguyên k cho biết số thứ k trong dãy Fibonacci cần tìm. ($1 \leq k \leq 50$)
 - Xem thêm example input để hiểu rõ hơn
 - Output được in ra theo đúng định dạng sau:
 - Mỗi case được in ra trên 8 dòng.
 - Dòng đầu tiên là chuỗi Case #case: với case là stt của case
 - Dòng thứ 2: tổng của dãy số thực đã cho
 - Dòng thứ 3: giá trị min của dãy số thực đã cho
 - Dòng thứ 4: giá trị max của dãy số thực đã cho
 - Dòng thứ 5: chuỗi str1 sau khi chuyển sang chữ IN HOA
 - Dòng thứ 6: chuỗi str2 sau khi chuyển sang chữ thường
 - Dòng thứ 7: chuỗi str1 – str2 sau khi chuyển chữ cái đầu thành chữ in Hoa
 - Dòng thứ 8: giá trị của số Fibonacci thứ k.
 - Xem thêm example output để hiểu rõ hơn

- Example:

- INPUT:

Giá trị	Mô tả
1	Số lượng testcase
5	Số lượng phần tử
1.0 2.0 3.0 4.0 5.0	Mảng 1 chiều các số thực
nguyen vAn A	Chuỗi str1
NGUYEN VAN b	Chuỗi str2
4	Giá trị k

- OUTPUT:

Case #1:

Sum: 15.00

Min: 1.0

Max: 5.0

To upper: NGUYEN VAN A

To lower: nguyen van b

To upper first char: Nguyen Van A - Nguyen Van B

Fibonacci(4): 3