

**BÀI ÔN THI GIỮA KỲ**  
**MÔN: LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG**

**YÊU CẦU SINH VIÊN ĐỌC KỸ TRƯỚC KHI LÀM BÀI:**

1. Tạo một thư mục đặt tên là **MSSV\_Họtên** (viết liền, không dấu).
2. **Sinh viên nhận 02 folder Cau1 và Cau2**, sao chép 02 folder nhận được vào thư mục MSSV\_Họtên vừa tạo.
3. Sau khi làm bài xong, sinh viên nén thư mục MSSV\_Họtên và nộp. Bài nộp đúng là bài nộp có thư mục MSSV\_Họtên chứa 02 thư mục Cau1 và Cau2 chứa bài làm.

**PHẦN CÂU HỎI**

Sinh viên lưu ý đảm bảo bài làm của từng câu phải biên dịch, chạy được và đặt đúng thư mục theo hướng dẫn. Tất cả các trường hợp **làm sai yêu cầu**, sinh **lỗi** trong **quá trình biên dịch** hoặc trong **quá trình chạy** khi chấm đều bị **0 điểm cả câu**.

Phần hàm main cho sẵn ở các câu trong đề chỉ để sinh viên kiểm thử bài làm, sinh viên không nhất thiết phải xóa hàm main khi nộp nhưng phải đảm bảo hàm main không gây lỗi cho bài làm.

Sinh viên chú ý **KHÔNG** chỉnh sửa tên file, tên hàm, thứ tự tham số trong các file đã nhận và làm bài trực tiếp trên các file này.

**Câu 1:**

Trong thư mục **Cau1** chứa sẵn file **JavaBasic.java**, sinh viên mở file và thực hiện các yêu cầu sau:

- a. Cài đặt hàm **public static int sumNegativeElements(int[] a)** để tính tổng tất cả các phần tử âm trong mảng.
- b. Cài đặt hàm **public static String uppercaseFirstVowels(String str)** để trả về chuỗi in hoa các ký tự đầu tiên của mỗi từ và là nguyên âm. Cho biết các từ trong chuỗi được cách nhau bằng dấu khoảng trắng. Ví dụ: cho chuỗi họ tên là “nguyen tran anh oanh” thì chuỗi trả về là “nguyen tran Anh Oanh”.
- c. Cài đặt hàm **public static int findMinNegativeElement(int[] a)** trả về vị trí số âm nhỏ nhất trong mảng. Nếu mảng không có số âm thì trả về -1. Vị trí đầu tiên của mảng là 0.
- d. Cài đặt hàm **public static String getName(String str)** để trả về họ tên trong chuỗi truyền vào có định dạng “Ho ten: <phần họ tên cần tách ra>”. Ví dụ chuỗi truyền vào là “Ho ten: Nguyen Van Hiep” thì chuỗi trả về sẽ là “Nguyen Van Hiep”.
- e. Cài đặt hàm **public static int findFirstMod3Element(int[] a)** trả về vị trí của số đầu tiên trong mảng chia hết cho 3. Nếu mảng không có số âm thì trả về -1. Vị trí đầu tiên của mảng là 0.
- f. Cài đặt hàm **public static int countString(String str, String k)** trả về số lượng từ **k** trong chuỗi **str** truyền vào. Ví dụ: cho chuỗi **str** truyền vào là “Pham Thi Uyen Uyen” và chuỗi **k** là “Uyen” thì phương thức trên sẽ trả về 2. Không phân biệt chữ hoa chữ thường khi đếm từ.

**Câu 2:**

Trong thư mục **Cau2**, chứa 02 file là **Rectangle.java** và **TestRectangle.java**, sinh viên mở file **Rectangle.java** và hiện thực lớp **Rectangle** theo yêu cầu sau:

**a. Thuộc tính:**

- Thuộc tính tên hình chữ nhật, màu hình chữ nhật: name, color (kiểu String)
- Thuộc tính chiều rộng và chiều dài: width, length (kiểu double)

**b. Phương thức:**

- Phương thức **khởi tạo** đầy đủ tham số:

**Rectangle(String name, String color, double width, double length)**

- Phương thức **get**:

Trả về tên hình chữ nhật và màu sắc: **getName(), getColor()**

Trả về chiều rộng và chiều dài: **getWidth(), getLength()**

- Phương thức **set**:

Nhận vào và cài đặt tên hình chữ nhật và màu sắc: **setName(String name), setColor(String color)**

Nhận vào và cài đặt chiều rộng, chiều dài: **setWidth(double width), setLength(double length)**

- Phương thức tính chu vi hình chữ nhật:

**public double getPerimeter()**

- Phương thức trả về loại hình chữ nhật:

**public String getType()**

- Diện tích hình chữ nhật  $\geq 10$  thì trả về "A".
- Diện tích hình chữ nhật  $< 10$  và  $\geq 5$  thì trả về "B".
- Diện tích hình chữ nhật  $< 5$  thì trả về "C".

*Lưu ý: Sinh viên chỉ trả về kết quả kiểu String, KHÔNG in ra màn hình*

- Phương thức kiểm tra hình hiện tại có phải là hình vuông hay không:

**public boolean isSquare()**

Nếu là hình vuông thì trả về true, ngược lại trả về false. Biết hình vuông là hình chữ nhật có chiều dài bằng chiều rộng.

- Phương thức tìm chiều dài đường chéo hình chữ nhật:

**public double calculateDiagonalLine()**

- Phương thức tạo ra một hình chữ nhật mới với tỉ lệ truyền vào:

**public Rectangle resize(double rate)**

Tham số truyền vào sẽ là một số thập phân đại diện cho tỉ lệ thay đổi, hình chữ nhật mới trả về sẽ có chiều dài = chiều dài hình gốc \* tỉ lệ, chiều rộng = chiều rộng hình gốc \* tỉ lệ. Nếu tỉ lệ là 1 thì hình mới sẽ có chiều dài và chiều rộng bằng hình gốc.

- Phương thức in thông tin hình chữ nhật (sinh viên chú ý kĩ các ký tự khoảng trắng và ký tự dấu câu phải giống hoàn toàn với kết quả ví dụ):

**public String toString()**

Định dạng: **Rectangle[tên hình chữ nhật, chiều dài, chiều rộng, diện tích, loại hình chữ nhật]**

Ví dụ một đối tượng Rectangle rec = new Rectangle("Ruby", "Red", 6, 8) sẽ có định dạng khi in ra màn hình là Rectangle[Ruby, 8.0, 6.0, 48.0, A]

Sinh viên dùng file ***TestRectangle.java*** trong thư mục **Cau2** để kiểm tra lại bài làm của mình. File này đã chứa sẵn phương thức main, sinh viên viết thêm code vào file này để kiểm thử các phương thức đã hiện thực.