

## Đề ôn tập giữa kỳ

### Môn: Cấu trúc dữ liệu và giải thuật 1

#### Đề 02

*Sinh viên tự viết lớp theo yêu cầu bên dưới và viết hàm main để kiểm tra lại.*

Cho lớp **RegularPolygon** quản lý các đa giác đều (giả sử các dữ liệu truyền vào đều thỏa mãn để tạo ra đa giác đều) với các thuộc tính được mô tả như sau:

- Tên đa giác: **name** (chuỗi ký tự, kiểu String)
- Số lượng cạnh: **sideAmount** (số nguyên, kiểu int)
- Độ dài cạnh: **sideLength** (số thực, kiểu double)
- Màu sắc: **color** (chuỗi ký tự, kiểu String)
- Số lượng đa giác hiện có: **quantity** (số nguyên dương, kiểu int, thuộc tính tĩnh - static)

Sinh viên định nghĩa lớp **RegularPolygon** vào file **RegularPolygon.java** theo mô tả.

#### Câu 1:

Viết hàm khởi tạo cho lớp **RegularPolygon**:

- Khởi tạo không tham số: **RegularPolygon()** (name = "", sideAmount = 3, sideLength = 1, color = "red")
- Khởi tạo đầy đủ tham số: **RegularPolygon(String name, int sideAmount, double sideLength, String color)**
- Khởi tạo tham số: **RegularPolygon(String name, int sideAmount)** (sideLength = 1, color = "red")
- Khởi tạo sao chép: **RegularPolygon(RegularPolygon rp)**

#### Câu 2:

Sinh viên viết hàm nhận vào và cài đặt thuộc các thuộc tính:

- Nhận vào và cài đặt tên đa giác: **setName(String)**
- Nhận vào và cài đặt số lượng cạnh của đa giác: **setSideAmount (int)**
- Nhận vào và cài đặt độ dài cạnh của đa giác: **setSideLength (double)**
- Nhận vào và cài đặt màu của đa giác: **setColor(String)**

#### Câu 3:

Sinh viên viết hàm trả về số lượng đa giác hiện có: **getQuantity()** (phương thức tĩnh - static)

**Câu 4:**

Sinh viên viết hàm trả về các thuộc tính:

- Trả về tên đa giác: **getName()**
- Trả về số lượng cạnh: **getSideAmount()**
- Trả về độ dài cạnh: **getSideLength()**
- Trả về màu sắc: **getColor()**

**Câu 5:**

Trong file **RegularPolygon.java**, sinh viên cài đặt các phương thức:

- Trả ra đa giác hiện tại là hình gì với prototype hàm: **public String getPolygon()**
  - Nếu **số cạnh bằng 3** thì trả về “Hình Tam Giác”
  - Nếu **số cạnh bằng 4** thì trả về “Hình Tứ Giác”
  - Nếu **số cạnh bằng 5** thì trả về “Hình Ngũ Giác”
  - Nếu **số cạnh bằng 6** thì trả về “Hình Lục Giác”
  - Nếu **số cạnh > 6** thì trả về “Hình Đa Giác Có Hơn 6 Cạnh”
- Tính chu vi của đa giác thông qua hàm: **public double parameter()**.  
Chu vi = số cạnh \* chiều dài cạnh
- Tính diện tích của đa giác: **public double area()** với các công thức sau:

$$\text{Diện tích} = (\text{độ dài cạnh})^2 * a$$

Số lượng cạnh	a
3	0.433
4	1
5	1.72
6	2.598

Nếu số lượng cạnh lớn hơn 6 thì trả về diện tích = -1.

Ví dụ:

- Độ dài cạnh là 3, hình vuông, ta có diện tích là  $3 * 3 * 1 = 9$ .
- Độ dài cạnh là 3, hình ngũ giác, ta có diện tích là  $3 * 3 * 1.72 = 15.48$
- Độ dài cạnh là 3, hình bát giác, ta có diện tích là -1.

### Câu 6:

Ghi đè (override) và cài đặt phương thức so sánh hai đa giác có bằng nhau hay không. Hai đa giác bằng nhau khi có số lượng cạnh bằng nhau, độ dài cạnh bằng nhau và màu sắc giống nhau. Nếu màu sắc của một trong hai đa giác là null thì hai đa giác không bằng nhau. Prototype là:

**public boolean equals(Object obj)**

Ví dụ với thông tin sau:

Đa giác	Số lượng cạnh	Độ dài cạnh	Màu sắc
s1	3	4	"blue"
s2	3	4	"blue"
s3	3	5	"blue"
s4	4	4	"blue"
s5	3	4	null
s6	3	4	null

Lời gọi phương thức **s1.equals(s2)** trả về **true**.

Lời gọi phương thức **s1.equals(s3)** trả về **false**.

Lời gọi phương thức **s1.equals(s4)** trả về **false**.

Lời gọi phương thức **s1.equals(s5)** trả về **false**.

Lời gọi phương thức **s5.equals(s6)** trả về **false**.

### Câu 7:

Cho lớp **PrismRPBased** là lớp quản lý hình lăng trụ đều kế thừa từ lớp **RegularPolygon** với mô tả như sau:

- Thuộc tính:
  - o Chiều cao lăng trụ: **height** (số thực, kiểu double)
- Phương thức:
  - o Khởi tạo không tham số: **PrismRPBased()** (height = 1)
  - o Khởi tạo đầy đủ tham số: **PrismRPBased(String name, int sideAmount, double sideLength, double height, String color)**
  - o Khởi tạo tham số: **PrismRPBased(String name, int sideAmount)** (height = 1)
  - o Khởi tạo tham số: **PrismRPBased(String name, int sideAmount, double height)**
  - o Trả về chiều cao của lăng trụ: **getHeight()**
  - o Nhận vào và cài đặt chiều cao lăng trụ: **setHeight()**

Sinh viên cài đặt thuộc tính và phương thức trên cho lớp **PrismRPBased** trong file **PrismRPBased.java**.

**Câu 8:**

Sinh viên cài đặt hàm với prototype: **public double getVolume()** để tính thể tích hình lăng trụ đều với công thức:

$$\text{Thể tích} = \text{diện tích đáy} * \text{chiều cao}$$

**Câu 9:**

Sinh viên thực hiện ghi đè (override) để cài đặt phương thức:

**public String toString()**

Để in ra đối tượng theo định dạng:

[Da giác mat day: (1), Dien tích day: (2), The tích: (3)]

Vị trí (1): dựa vào số cạnh – sideAmount để xác định đa giác mặt đáy:

- Nếu **số cạnh bằng 3** thì trả về “Hình Tam Giác”
- Nếu **số cạnh bằng 4** thì trả về “Hình Tu Giác”
- Nếu **số cạnh bằng 5** thì trả về “Hình Ngu Giác”
- Nếu **số cạnh bằng 6** thì trả về “Hình Lục Giác”
- Nếu **số cạnh > 6** thì trả về “Hình Đa Giác Co Hon 6 Canh”

Vị trí (2): diện tích mặt đáy của lăng trụ

Vị trí (3): thể tích của hình lăng trụ

Ví dụ:

**PrismRPBased prism** có 4 cạnh, diện tích là 9, thể tích là 90 thì khi gọi câu lệnh **System.out.println(prism)** kết quả in ra là:

[Da giác mat day: Hình Tu Giác, Dien tích day: 9, The tích: 90]

**-- HẾT --**