

Paradigma e Linguagem Orientada a Objetos

Prof. Fabio T. Matsunaga
Ciência da Computação
Universidade Estadual do Paraná (UNESPAR)
Campus Apucarana

Lista de exercícios – Classes e Objetos

1. Escreva uma classe em Java chamada Contador, que contenha um atributo de valor inteiro. A classe deve encapsular esse valor contendo os seguintes métodos:
 - Construtores sobrecarregados que inicializem um valor de diferentes formas:
 - Inicialize com zero (construtor sem parâmetro);
 - Inicialize com um valor (construtor com um parâmetro).
 - Zerar;
 - Incrementar;
 - Decrementar;
 - Retornar o valor do contador.
2. Escreva uma classe em Java chamada Reta2D, que represente uma reta ($y = ax + b$) no plano Cartesiano. Os atributos da classe devem ser o coeficiente angular a e o coeficiente linear da reta b . Em seguida, forneça os seguintes métodos da classe:
 - Construtores sobrecarregados que inicialize a equação da reta de acordo com os seguintes valores recebidos:
 - dois valores representando o coeficiente angular a e o coeficiente linear da reta b ;
 - dois pontos (x_1, y_1) e (x_2, y_2) , a partir dos quais a equação da reta é calculada a partir da equação $y = m(x - x_1) + y_1$, em que $m = \frac{y_2 - y_1}{x_2 - x_1}$.
 - *Getters* para o coeficiente angular e linear da reta;
 - Método que verifique se um ponto pertence à reta;
 - Método que gere uma String da equação no formato $y = ax + b$;
 - Método que dado um valor de x , retorne o valor de y ;
 - (Desafio) Método que dada uma outra reta, retorne o ponto de intersecção da reta (x, y) ou retorne `null` se as retas forem paralelas.
3. Escreva uma classe em Java chamada Circunferencia2D, que represente uma circunferência no plano Cartesiano. O atributo da classe deve ser o raio e o centro (x, y) da circunferência. Forneça os seguintes métodos da classe:
 - Construtores sobrecarregados que recebam os seguintes valores:
 - valor do raio e um ponto (x, y) , posicionando o centro da circunferência no ponto (x, y) ;
 - somente o valor do raio, posicionando o centro da circunferência na origem $(0, 0)$.
 - Método de acesso ao valor do raio da circunferência (*getter*);
 - Método de acesso ao centro do círculo (*getter*);
 - Métodos de aumentar e diminuir o círculo de acordo com um valor de raio;
 - Método para retornar a área do círculo ($A = \pi r^2$);
 - Método para retornar o comprimento da circunferência ($P = 2\pi r$);

- Método para mover a circunferência para o local indicado por um outro ponto.
4. Crie uma classe denominada `Hora` que representará uma hora com o formato padrão. A classe contém três atributos: horas, minutos e segundos. Essa classe também conterá os métodos:
- Construtores sobrecarregados
 - Um construtor sem parâmetros, que inicialize a hora com 00 : 00 : 00;
 - Um construtor com três parâmetros, que inicialize as horas, minutos e segundos com os valores passados;
 - Getters e setters;
 - Método para retornar a quantidade total da hora em segundos;
 - Método que passe por parâmetro um valor de hora (um objeto `Hora` ou valores em horas, minutos e segundos) e calcule a diferença da hora atual com o valor passado;
 - Método para retornar o horário em formato string "hh:mm:ss";
5. Implemente uma classe em Java denominada `Pessoa`. Essa classe pessoa deve conter os seguintes atributos: nome (`String`), cpf (`String`) e data de nascimento (`String`), além de ter um construtor que inicialize esses três atributos. Em seguida, faça uma herança a partir da classe `Pessoa` mais duas subclasses: `Professor`, `Aluno`. As subclasses, além de ter os métodos e atributos da classe `Pessoa`, deve conter também alguns atributos particulares. `Professor` por exemplo pode possuir um salário (`double`) e uma escola (`String`). Já o `aluno` possuirá um número de matrícula (`long`), uma escola (`String`) e a série (`int`).
6. Crie uma classe denominada `Vetor` que representará um vetor de inteiros e suas propriedades. A classe conterá dois atributos: um vetor de inteiros e o seu tamanho. Essa classe também conterá os métodos:
- Construtores sobrecarregados
 - Um construtor sem parâmetros, que inicialize somente o tamanho como zero (vetor vazio);
 - Um construtor com dois parâmetros, um vetor e o seu tamanho, que inicialize os atributos com os valores passados por parâmetro;
 - Getters e setters;
 - Método para inserir um número no vetor;
 - Método para calcular o somatório do vetor e retornar o valor;
 - Método para inverter o vetor;
 - Método para ordenar o vetor;
 - Método para imprimir os valores do vetor;
7. Implemente um programa em Java que faça o gerenciamento de pedidos de um cliente. Crie uma classe chamada `Produto`, contendo o preço e a quantidade e métodos que calculem o preço total. Em seguida, Crie uma classe `Pedido` que contenha uma certa quantidade de produtos. Esta classe deve conter métodos para adicionar produtos e calcular o preço total que o cliente irá pagar.