

Paradigma e Linguagem Orientada a Objetos

Prof. Fabio T. Matsunaga
Ciência da Computação
Universidade Estadual do Paraná (UNESPAR)
Campus Apucarana
Entregar até o dia 01/11/2016

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 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.
3. 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";

4. Escreva uma classe `EntradaDeCinema` com a seguinte estrutura (dados):
 - Data `dataDoFilme` (use a classe `Data` implementada durante as aulas)
 - Hora `horario` (use a classe `Hora` implementada durante as aulas)
 - `int sala`
 - `float valor`
 - Ações (métodos):
 - Construtor: com a finalidade de inicializar todos os atributos.
 - `CalculaDesconto`: que deve receber como parâmetro a data de nascimento do cliente (do tipo `Data`) e caso seja menor de 12 anos, deve ser dado um desconto de 50% no valor normal.
 - `CalculaDesconto`: que deve receber como parâmetro a data de nascimento do cliente (do tipo `Data`) e o número de sua carteira de estudante (do tipo `int`). Se o estudante tiver idade entre 12 e 15 anos, deve ser dado um desconto de 40%, de 16 a 20 um desconto de 30% e mais que 20 anos um desconto de 20% no valor normal.
 - `CalculaDescontoHorário`: esse método deve dar um desconto de 10% sobre o valor afe-rido após todas as outras opções de desconto, caso o horário do filme seja antes das 16 horas.
 - `toString()`: que deve imprimir todos os dados do ingresso.
 - Desenvolva uma aplicação que leia os dados necessários para instanciar e imprimir o ingresso para clientes normais, menores de 12 anos e estudantes.
5. Implemente uma classe abstrata chamada `FormaGeometrica` com dois métodos abstratos: `calcularArea` e `CalcularPerimetro`. Em seguida, crie três classes herdadas dessa forma: `Retangulo`, `Circulo` e `Quadrado`. Redefina a forma de calcular a área e o perímetro em cada uma dessas formas, definindo os atributos corretos para cada um (lado para o `Quadrado`, lado A e lado B para o `Retangulo` e raio para a `Circunferência`). Crie uma interface chamada `AlterarForma`, que contenham um métodos abstratos para aumentar e diminuir a forma. Use essa interface em cada subclasse, reimplementando os devidos métodos.
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 va-zio);
 - 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 (pesquise um método de ordenar um vetor);
 - Método para imprimir os valores do vetor;
7. Escreva uma classe `Conjunto`, que represente um conjunto de tamanho variável (crescimento de array por demanda) de elementos do tipo `String`. Escreva os seguintes membros para a classe:
 - Um método que permita adicionar um elemento para o conjunto (o elemento não pode existir no conjunto);

- Um método que permita verificar se um dado elemento pertence ao Conjunto;
- Um método uniao, que retorne um novo conjunto de acordo com a semântica da operação união entre conjuntos (um novo conjunto, sem elementos repetidos, com a combinação dos elementos dos dois conjuntos originais, o que recebeu a mensagem e o que foi passado como parâmetro);
- Um método inter, que retorne um novo conjunto de acordo com a semântica da operação interseção entre conjuntos (um novo conjunto, sem elementos repetidos, com os elementos que estejam nos dois conjuntos originais, o que recebeu a mensagem e o que foi passado como parâmetro);
- Um método menos, que retorne um novo conjunto de acordo com a semântica da operação subtração entre conjuntos (um novo conjunto, sem elementos repetidos, com os elementos do conjunto que recebeu a mensagem, e que não existam no conjunto passado como parâmetro).