



Disciplina	Curso	Turno	Período
Algoritmos e Estruturas de Dados I	Ciência da Computação	Manhã	1º
Professor Felipe Cunha (felipe@pucminas.br)			

## Lista de Exercícios 11

1. Crie uma classe chamada `NotaFiscal` que um hardware utilize para representar uma nota fiscal em uma loja de peças. Uma nota fiscal deve incluir quatro dados como atributos:
  - a. Número da peça (string);
  - b. Descrição da peça (string);
  - c. Quantidade comprada (inteiro);
  - d. Preço (número real).

A classe deve incluir ainda getters e setters para cada um dos atributos. Ainda, deve haver um método `getTotalNota` que calcule e retorne o total de um vetor de objetos. Escreva um programa que teste cada um dos métodos da classe.

2. Crie uma classe `Empregado` que inclua os atributos `nome` (string), `sobrenome` (string) e `salário mensal` (número real). A classe deve incluir construtores para cada um dos atributos para evitar lixo e também getters e setters. Crie um programa que teste a classe, criando dois objetos, calculando o salário anual de cada, dando um aumento de 10% e calcule o salário anual novamente.
3. Crie uma classe `Aluno` com atributos que armazenem o nome, a série que cursa e o grau. Crie um getter e um setter, além de uma variável static que conte a quantidade de objetos criados, a ser utilizada no construtor e destrutor. Crie um vetor com 15 objetos desta classe e preencha os dados através do setter.
4. Crie uma classe `Estacionamento` para armazenar dados de um estacionamento. Os atributos devem representar a placa e modelo do carro além da hora de entrada e saída do estacionamento. Utilize dois objetos da classe `Tempo` criada no exercício anterior. Crie métodos para:
  - a. Funcionar como getter e setter;
  - b. Inicializar os dados com vazio ou zero;
  - c. Imprimir os dados de um carro estacionado;
  - d. Calcular e retornar o valor a ser pago pelo carro estacionado. Considere o preço de R\$1,50 por hora. Utilize o método da classe `Tempo`.

Crie um vetor de 5 posições e preencha-o para testar sua classe.

5. Crie a classe `Veiculo`, contendo o peso, a velocidade máxima, e o preço. Inclua um construtor sem argumentos e um com argumentos para inicialização dos atributos. Crie métodos que funcionem como getter e setter e print.

Crie a classe `Motor`, contendo o número de cilindros e a potência. Inclua um construtor sem argumentos e um com argumentos para inicialização dos atributos. Crie métodos que funcionem como getter e setter e print.

Crie a classe `CarroPasseio`, derivada das classes `Veiculo` e `Motor`. Inclua atributos como modelo e cor. Crie um construtor para esta classe que chame explicitamente o construtor das classes base, um getter e um setter, além de um método print, que utiliza o método print da classe base. Crie um driver para testar sua aplicação.

6. Crie a classe Caminhao, derivada das classes Motor e Veiculo. Inclua os atributos toneladas, altura máxima e comprimento. Crie um construtor para esta classe que chame explicitamente o construtor das classes base, um getter e um setter, além de um método print, que utiliza o método print da classe base. Crie um driver para testar sua aplicação.
7. Crie uma classe utilitária (Utils) com um método para converter temperaturas em graus Fahrenheit para graus Celsius (toCelsius()), ambas representadas em ponto-flutuante (double). Porém, caso o valor a ser convertido seja menor que zero absoluto (-459,67°F) deve-se lançar a exceção.