



PUC Minas

PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS
Laboratório de Algoritmos e Técnicas de Programação

Observações:

- Cópias serão desconsideradas, ou seja, a nota será igual a 0 (zero).
- Implemente os programas utilizando a linguagem C#.
- Na resolução dos exercícios só podem ser utilizados comandos vistos nas aulas.

Lista de Exercícios 09 - POO

01. Escreva um programa para cadastrar um cliente em uma loja. As informações necessárias são: “nome”, “endereço” e “telefone”. Crie uma Cliente com todos campos privados, dois métodos construtores diferentes e propriedades. No Main, crie três clientes, e utilize todas as propriedades.

02. Crie uma classe chamada Data que possua três campos: dia, mês e ano. Sua classe deve ter um construtor que inicialize os três campos, assumindo que os valores inseridos sejam corretos. Acrescente um método ToString que retorne uma string com o dia, mês e ano separados por barras (/).

Implemente uma classe Teste (com método Main) que demonstre todas as potencialidades da classe Data, isto é, crie objetos Data e utilize os métodos e propriedades da classe data.

03. Crie uma classe "Circulo" com o campo "raio". Implemente um método construtor e a propriedade. Adicione os seguintes métodos:

- CalcularArea: método sem parâmetros, que deve calcular e retornar a área do círculo
- CalcularDiametro: método sem parâmetros, que deve calcular e retornar o diâmetro do círculo.
- CalcularPerimetro: método sem parâmetros, que deve calcular e retornar o diâmetro do círculo.

Implemente uma classe Teste (com método Main), crie 4 círculos e mostre a utilização de todos os métodos da classe Circulo.

04. Crie uma classe "Aluno" com os campos "nome", "matricula" e "notas" (array de double com tamanho 10). Todos os campos devem ser privados. Adicione um método para calcular e retornar a média das notas do aluno.

Implemente uma classe Teste (com método Main), crie dois alunos e imprima a média das notas dos alunos.

05. Altere o programa do Exercício 1 para armazenar 10 clientes. No método Main deve ser criado um vetor de Clientes (Cliente[]). Todas as informações dos clientes devem ser lidas do teclado. Modifique o nome do primeiro cliente (usando a propriedade), e imprima os dados de todos os clientes (implemente o método ToString na classe Cliente).

06. Crie uma classe “Cargo” com os seguintes campos privados: “código do cargo” e “valor da hora trabalhada”.

Crie uma classe “Funcionario” com os seguintes campos privados: “cargo” (tipo Cargo), “sexo” (char) e “quantidade de horas trabalhadas” (double). Essa classe deve possuir ainda, um método para calcular o salário final, ou seja, o salário de acordo com o cargo (valor da hora trabalhada multiplicado pelo número de horas trabalhadas).

Implemente uma classe Teste (com método Main), crie três cargos e três funcionários. Calcule e imprima o salário final dos funcionários.

07. Crie as seguintes classes:

- Crie a classe Etiqueta com os campos privados marca (string) e tamanho (string). Crie seu construtor, que recebe parâmetros para inicializar os campos.
- Crie a classe Produto com os campos privados nome (string), preço (double) e etiqueta, do tipo Etiqueta. Crie um construtor, que receba como parâmetro o nome, preço e um objeto Etiqueta. Crie

outro construtor, que receba o nome e preço do produto, e também a marca e tamanho da etiqueta como parâmetros (o objeto Etiqueta deve ser instanciado no construtor).

- Crie uma classe TesteProduto com o método Main, onde você deve ler os dados de um produto, instanciar um produto e exibir na tela os campos do produto criado. Leia os dados de outro produto, instancie outro produto e exiba na tela os campos do produto criado (utilize os dois construtores). Depois altere os campos do produto criado anteriormente. Exiba novamente na tela os dados do produto.

08. Crie as classes “Cliente”, “ContaCorrente” e “Agencia” conforme abaixo:

- a) A classe Cliente possui os campos privados “nome” e “CPF”, ambos do tipo String, e um campo conta do tipo ContaCorrente.
- b) A classe ContaCorrente tem os campos privados número e dígito, ambos inteiros, o campo agência do tipo Agencia e o campo saldo (double). Crie também um método depositar que receba um parâmetro double com o valor do depósito e aumente o saldo da conta. Crie também um método "sacar" que receba um parâmetro double com o valor do saque e diminua o saldo da conta. A conta não pode ficar negativa. Crie, finalmente, um método "ConsultarSaldo" que retorna uma String contendo o número da conta corrente com dígito, o número da agência com dígito e o saldo da conta corrente.
- c) A classe Agencia tem os campos privados nome (String), número e dígito do tipo int.
- d) Para testar crie uma classe CaixaEletronico, que irá conter o método Main. No Main instancie um cliente com os seguintes dados:

Nome: Ademar da Silva

- CPF: 123231518-12

- Conta Corrente: 1234, Dígito: 4

- Agência: Pampulha, 7890, Dígito: 5

- Saldo Inicial: R\$150.00

Faça as seguintes operações:

- sacar 140.0

- consultar saldo

- sacar 200.0

- consultar saldo

- depositar 25.45

09. Considere a classe Estacionamento, cuja assinatura está definida abaixo:

```
using System;
class Estacionamento
{
    private String nome; //nome do estacionamento.

    private int numTotalVagas; //número total de vagas do estacionamento.

    private int numVagasLivres; //número de vagas livres no estacionamento.

    private String[] vagas; // vetor que armazena cada uma das vagas do
    estacionamento. Caso a vaga esteja ocupada, este vetor indica a placa do veículo que a
    ocupa.

    public Estacionamento(String nome, int numTotalVagas);
    public int Estacionar(String placa);
    public int ObterVagaOcupada(String placa);
    public void RetirarVeiculo(String placa);
    public int ObterNumVagasLivres();
    public void ExibirOcupacao();
}
```

Implemente todos os métodos da classe Estacionamento acima, obedecendo às seguintes descrições:

- a) Estacionamento(String nome, int numTotalVagas): Construtor que inicializa os campos nome e numTotalVagas com os valores passados por meio dos parâmetros desse método; o atributo numVagasLivres com o número total de vagas do estacionamento; e instancia o vetor vagas.

- b) Estacionar(String placa): Se existirem vagas disponíveis no estacionamento, encontra a primeira posição vazia do vetor vagas e insere a placa informada como parâmetro nessa posição, deve-se retornar a posição que foi feita a inserção. Se não existirem vagas disponíveis, esse método deve retornar -1;
- c) ObterVagaOcupada(String placa): Retorna o número da vaga, ou seja, a posição do vetor vagas, ocupada pelo veículo com a placa informada como parâmetro. Se a placa informada não for encontrada, deve retornar -1;
- d) RetirarVeiculo(String placa): Retira o veículo cuja placa corresponde à informada como parâmetro para esse método, marcando a vaga correspondente com null;
- e) ObterNumVagasLivres(): Retorna o número de vagas disponíveis no estacionamento;
- f) ExibirOcupacao(): Imprime o número de cada vaga do estacionamento e a placa do veículo que a ocupa ou a informação de que a vaga está vazia.

Crie a classe Program com o método Main, conforme descrito abaixo:

```
using System;
class Program
{
    static void Main(string[] args)
    {
        int vagaOcupada;
        Estacionamento e = new Estacionamento("Estacionamento", 10);

        vagaOcupada = e.Estacionar("HKT0098");
        vagaOcupada = e.Estacionar("OLP4290");
        vagaOcupada = e.Estacionar("HJB0495");
        vagaOcupada = e.Estacionar("OWB3904");

        Console.WriteLine("Ocupação após a chegada de quatro clientes:");
        e.ExibirOcupacao();
        //HKT0098 OLP4290 HJB0495 OWB3904 vazia vazia vazia vazia vazia vazia

        vagaOcupada = e.ObterVagaOcupada("HKT0098");
        Console.WriteLine("Veículo HKT0098 está na vaga {vagaOcupada}");
        // Veículo HKT0098 está na vaga 0

        e.RetirarVeiculo("HKT0098");
        Console.WriteLine("Após a retirada do veículo de placa HKT0098:");
        e.ExibirOcupacao();
        //vazia OLP4290 HJB0495 OWB3904 vazia vazia vazia vazia vazia vazia

        vagaOcupada = e.Estacionar("HTP5619");
        vagaOcupada = e.Estacionar("BOL4861");
        vagaOcupada = e.Estacionar("HGT9436");
        Console.WriteLine("Ocupação após a chegada de mais 3 clientes:");
        e.ExibirOcupacao();
        //HTP5619 OLP4290 HJB0495 OWB3904 BOL4861 HGT9436 vazia vazia vazia vazia

        Console.WriteLine("Vagas livres: {e.ObterNumVagasLivres()}");
        //Vagas livres: 4
        Console.ReadLine();
    }
}
```