

LP4EL – Exercícios

Aula 2 – Programação Orientada a Objetos com C#

Instruções para entrega das listas de exercícios:

Meio de Entrega: As resoluções das listas de exercícios devem ser entregues exclusivamente por meio do ambiente Moodle (<http://eadcampus.spo.ifsp.edu.br>).

Forma de Entrega: Para exercícios com uma única classe, deve ser entregue o arquivo da classe (extensão CS) referente ao exercício. Por exemplo: Exercicio3.cs. Para exercícios com mais de uma classe, cada exercício deve ter uma pasta, na qual serão colocados os arquivos CS referentes ao exercício. Por exemplo: Para o Exercício 4, deve existir uma pasta “Exercicio4” contendo todos os arquivos CS deste exercício. **Entregue apenas os arquivos CS.** Todas as pastas e arquivos da lista devem ser compactados em um único arquivo (extensão RAR ou ZIP), cujo nome deverá conter a aula, o primeiro e o segundo nomes do aluno. Por exemplo: Aula3_JoaoCarlos.zip.

Prazo de Entrega: O prazo de entrega está definido na própria página de exercícios do Moodle, lembrando que o sistema bloqueia o envio de arquivos após a data e horário indicados.

Obs.: A resolução deste(s) exercício(s) deve ser feita de forma INDIVIDUAL. Listas de exercícios com uma ou mais respostas idênticas serão desconsideradas integralmente para efeitos de nota de participação.

1. **(Atividade assíncrona referente ao dia 20/09/2021 – Vale presença em 1 aula)** Faça um programa console em C# que utilize os seguintes conceitos de programação orientada a objetos: classes, objetos, atributos e métodos. A aplicação deve possuir as seguintes classes:
 - **Produto** – Classe que contém os atributos privados (codigo, descricao, preco e ativo), além dos métodos públicos **get** e **set**. Os métodos **get** devem retornar os respectivos valores dos atributos da classe **Produto**. Os métodos **set** devem atribuir os valores recebidos como argumento aos respectivos atributos da classe.
 - **Inicio** – Classe que contém o método **Main**. Deve ser criado um objeto **Produto** que chamará os métodos **get** e **set** da classe **Produto**. Use os métodos **set** para atribuir os valores informados pelo usuário aos atributos da classe **Produto**, e os métodos **get** para apresentar seus novos valores ao usuário.

Obs.: Crie um projeto específico “Exercicio1”, mas escreva as duas classes em um mesmo arquivo CS.
2. **(Atividade assíncrona referente ao dia 20/09/2021 – Vale presença em 1 aula)** Faça uma cópia do programa anterior e modifique-o para que passe a usar propriedades autoimplementadas ao invés dos métodos **get** e **set**.
3. **(Atividade assíncrona referente ao dia 20/09/2021 – Vale presença em 1 aula)** Faça uma cópia do programa anterior e modifique as propriedades para a forma explícita. Inclua verificações nos métodos **set** para que os atributos possam ser modificados somente se cumprirem determinadas condições, por exemplo: `codigo != 0`, `preco > 0`. Inclua um construtor e um destrutor na classe **Produto**. No construtor, inicialize os atributos da classe. No destrutor, apenas emita uma mensagem ao usuário.
4. **(Atividade assíncrona referente ao dia 25/09/2021 – Vale presença em 5 aulas)** Faça um programa em C# que apresente ao usuário as opções a seguir, enquanto ele não digitar a opção 0 (zero). De acordo com o número da opção informada, o programa deverá efetuar a operação, solicitando as informações ao usuário quando necessário. Os dados dos clientes devem ser armazenados em dois vetores de objetos, um do tipo `PessoaFisica[10]`, que

armazenará objetos contendo código, endereço, telefone, nome, CPF dos clientes, e um vetor do tipo PessoaJuridica[10], que armazenará objetos contendo código, endereço, telefone, razaoSocial e CNPJ dos clientes. Seguem as opções:

- 1) **Inserir cliente** – Primeiramente, pergunte ao usuário se o cliente é pessoa física (PF) ou jurídica (PJ). Para pessoa física, solicite o nome, o CPF, o endereço e o telefone. Para pessoa jurídica, solicite a razão social, o CNPJ, o endereço e o telefone. O código do cliente não deve ser informado pelo usuário, deve ser gerado automaticamente a partir do número 1. Além disso, a sequência de códigos deve ser única para pessoas físicas e jurídicas, ou seja, não devem existir clientes PF e PJ com um mesmo código. Informe ao usuário se houve sucesso ou não na inserção.
- 2) **Remover cliente** – Primeiramente, pergunte ao usuário se o cliente é pessoa física (PF) ou jurídica (PJ). Solicite então o código do cliente. Na remoção, deve ser atribuído o valor “null” à posição do vetor relacionada ao cliente informado. Informe ao usuário se houve sucesso ou não na remoção (por exemplo: código inexistente).
- 3) **Consultar clientes** – A consulta deve apresentar ao usuário os dados de todos os clientes cadastrados, separados em pessoas físicas e pessoas jurídicas.

A aplicação deve possuir as seguintes classes:

- **Cliente** – Classe que contém os atributos privados int codigo, string endereco e string telefone, além dos métodos protegidos **get** e **set** necessários.
- **PessoaFisica** – Classe filha da classe **Cliente**. Contém os atributos privados String nome e String cpf, além dos métodos públicos **get** e **set** necessários.
- **PessoaJuridica** – Classe filha da classe **Cliente**. Contém os atributos privados String razaoSocial e String cnpj, além dos métodos públicos **get** e **set** necessários.
- **Inicio** – Classe que contém o método **Main**, onde são instanciadas as classes **PessoaFisica** e **PessoaJuridica**. Na opção 1, deve-se chamar um método que usará os métodos **set** para criar e retornar um objeto contendo os dados da pessoa física ou da pessoa jurídica, dependendo do tipo de cliente informado. Este objeto deve ser inserido então no vetor de pessoas físicas ou pessoas jurídicas. Por fim, o código deve ser incrementado para inserção do próximo cliente. Na opção 2, deve-se percorrer o vetor de objetos (pessoa física ou jurídica) em busca de um objeto que contenha o código de cliente igual ao código informado pelo usuário. Ao encontrá-lo, deve-se atribuir “null” ao objeto. Na opção 3, deve-se chamar um método que usará os métodos **get** para recuperar os dados de cada objeto armazenado nos vetores de pessoa física e pessoa jurídica e apresentá-los ao usuário.