

A5LP1 – Exercícios

Aula 4 - 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>). Usar o mesmo usuário e senha do Sistema Aurora.

Forma de Entrega: Para exercícios com implementação de programas console, devem ser entregues os arquivos das classes (extensão CS). Para exercícios com implementação de programas Windows Forms, devem ser entregues as pastas dos projetos que contêm as aplicações desenvolvidas. Para exercícios com banco de dados, também devem ser entregues as instruções SQL usadas (extensão SQL ou TXT). Todos os arquivos da lista devem ser compactados em um único arquivo (extensão RAR ou ZIP), cujo nome deverá conter a aula, o nome e um sobrenome do aluno. Por exemplo: Aula2_JoaoSilva.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.: O material da disciplina e as listas de exercícios estão disponíveis no ambiente Moodle.

1. 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.
2. Faça uma cópia do programa anterior e modifique-o para que passe a usar propriedades ao invés dos métodos **get** e **set**. As propriedades devem ser usadas na forma simplificada, por exemplo: `public int codigo {get; set; }.`
3. Faça uma cópia do programa anterior e modifique as propriedades para a forma não simplificada. Inclua verificações nos métodos **set** para que os atributos possam ser modificados somente se cumprirem determinadas condições. 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. Faça um programa console 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 informado, o programa deverá efetuar a operação, solicitando as informações ao usuário quando necessário. Os dados dos clientes devem ser armazenados em três vetores, um para cada dado (codigo, nome e CPF).
 - 1) **Inserir cliente** – Solicite ao usuário o nome e o CPF do cliente (o código deve ser gerado automaticamente a partir do número 1).
 - 2) **Consultar cliente específico** – Solicite ao usuário o código do cliente. A consulta deve apresentar ao usuário somente os dados do cliente informado.
 - 3) **Consultar clientes** – A consulta deve apresentar ao usuário os dados de todos os clientes cadastrados.

A aplicação deve possuir as seguintes classes:

- **Cliente** – Classe que contém os atributos privados (**vetorCodigo[10]**, **vetorNome[10]** e **vetorCpf[10]**), além dos métodos públicos:
bool **inserirCliente**(int codigo, string nome, string cpf)
void **setCodigo**(int codigo)
void **setNome**(int codigo, string nome)
void **setCpf**(int codigo, string cpf)
int[] **getVetorCodigo**()
string[] **getVetorNome**()
string[] **getVetorCpf**()

No método **inserirCliente**, se o nome recebido tiver pelo menos um espaço (indicando que o sobrenome foi informado) e o CPF tiver 11 caracteres, deve-se chamar os métodos **setCodigo**, **setNome** e **setCPF**, além de retornar “true” para indicar que houve sucesso na inserção dos dados. Estes métodos, por sua vez, devem atribuir os valores recebidos como argumento aos vetores **vetorCodigo**, **vetorNome** e **vetorCpf**, iniciando na posição zero de cada vetor (use o código do cliente para indicar o índice dos vetores (código - 1)). Os métodos **getVetorCodigo**, **getVetorNome** e **getVetorCPF** devem retornar os vetores **vetorCodigo**, **vetorNome** e **vetorCpf** respectivamente.

- **Inicio** – Classe que contém o método **Main**, onde é instanciada a classe **Cliente**. Na opção 1, o objeto criado deve chamar o método **inserirCliente** desta classe, para inserir os dados do cliente nos vetores. Se este método retornar “true”, o código do cliente deve ser incrementado. Nas opções 2 e 3, o objeto deve chamar os métodos **getVetorCodigo**, **getVetorNome** e **getVetorCPF** para recuperar e apresentar os dados dos clientes cadastrados ao usuário.
5. Faça uma cópia do programa do exercício anterior e modifique-o para que armazene os dados dos clientes em um vetor de objetos do tipo **Cliente[10]**, ao invés de armazená-los em três vetores separados. Cada objeto armazenado no vetor representará um cliente, ou seja, cada objeto encapsulará os dados de cada cliente (código, nome e CPF).