
POO - Lista 06 - Vetores e Matrizes

Prof. Gilbert Azevedo da Silva

I. Implementar classes em C# para representar:

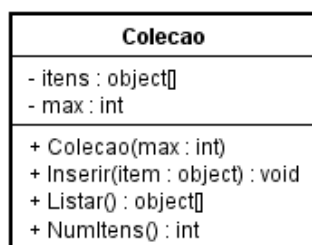
1. Uma Coleção

Escrever a classe Coleção de acordo com o diagrama UML apresentado.

A classe deve manter um vetor de objetos de quaisquer classes, utilizando para isso um vetor de objects. O vetor deve ter a capacidade de armazenar um número máximo de elementos, dado pelo atributo max, cujo valor é informado no construtor da classe.

Os métodos Inserir e Listar da classe devem ser usados para inserir e listar os objetos na coleção. O método NumItens deve retornar o número de elementos inseridos até o momento.

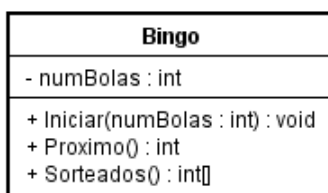
Insira outros atributos e métodos nas classes, caso necessário.



2. Um Bingo

Escrever a classe Bingo de acordo com o diagrama UML apresentado.

A classe é utilizada para realizar um jogo de bingo. O método Iniciar inicia uma partida, definindo o número de bolas do jogo. O método Próximo sorteia uma bola, retornando o seu número (deve ser um valor entre um e o número de bolas ou menos um, caso todas as bolas já tenham sido sorteadas). O método Sorteados retorna um vetor com todas as bolas já sorteadas. Insira outros atributos e métodos nas classes, caso necessário.

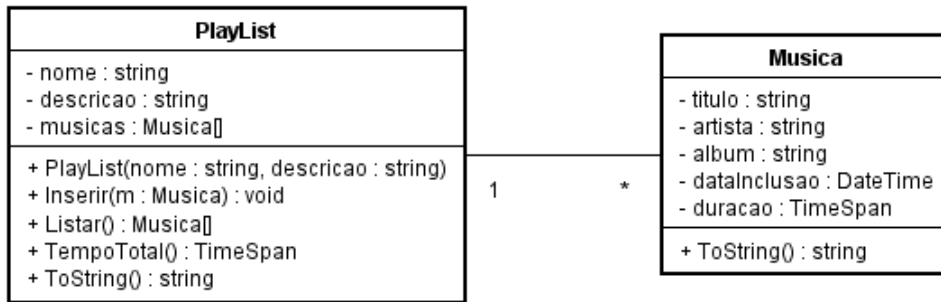


3. Uma PlayList

Escrever as classes PlayList e Música acordo com o diagrama UML apresentado.

As classes são usadas para registrar as playlists de um aplicativo de músicas. Cada playlist deve ter um nome e uma descrição além de uma lista de músicas. O construtor da classe recebe o nome e a descrição inicial da playlist e possui métodos para inserir uma música e listar as músicas já inseridas. O método TempoTotal deve retornar o tempo total da lista dado pela soma da duração de todas as músicas inseridas na lista.

Cada música tem informações sobre título, artista e álbum da música, além das informações de data de inclusão na playlist e tempo de duração. O método ToString de cada classe deve retornar um texto com os atributos do objeto. Insira outros atributos e métodos nas classes, caso necessário.

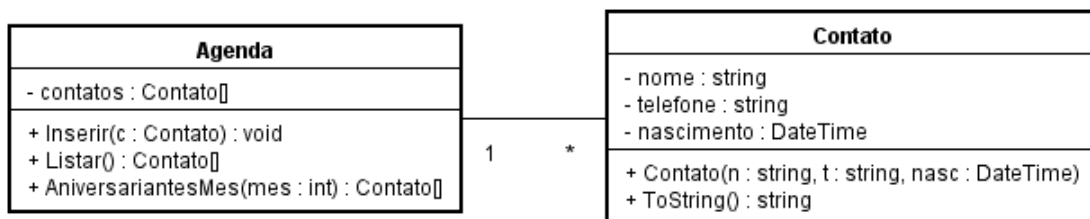


4. Uma Lista de Aniversariantes

Escrever as classes Agenda e Contato de acordo com o diagrama UML apresentado abaixo.

As classes devem ser usadas para gerenciar uma agenda de contato e obter lista de aniversariantes do mês. A classe Agenda mantém um vetor com os contatos inseridos na agenda. Deve ter os métodos Inserir e Listar para inserir um contato na agenda e retornar os contatos já inseridos, além do método AniversariantesMes que retorna uma lista com os contatos que aniversariam no mês informado.

A classe Contato define os dados a ser armazenados para cada contato que são recebidos no construtor da classe. O método ToString deve retornar um texto com os atributos do objeto. Insira outros atributos e métodos nas classes, caso necessário.



5. Uma Empresa

Escrever as classes Empresa e Cliente de acordo com o diagrama UML apresentado abaixo.

As classes são usadas para cadastrar os clientes de uma empresa de cartões de crédito.

A classe Empresa possui uma lista (vetor) com os clientes da empresa e define as seguintes operações: Inserir, para cadastrar um cliente e Listar, para retornar um vetor com os clientes já cadastrados.

A classe Cliente modela um cliente que deve possuir nome, cpf e limite de crédito. Um cliente pode também ter um sócio, também cliente. Nesta situação, o seu limite de crédito deve ser somado ao limite de crédito de seu sócio e vice-versa. A sociedade deve acontecer em pares, ou seja, quando um cliente A for sócio de um cliente B, B deve ser sócio de A. O construtor da classe recebe os dados iniciais do cliente. O método SetSocio é usado para definir a sociedade entre dois clientes; GetLimite retorna o limite do cliente, considerando o limite de seu sócio quando existir; ToString retorna um texto com os dados do cliente. Insira outros atributos e métodos nas classes, caso necessário.

