

Lista 2: Nivelamento – Introdução a Programação por Objetos

1) Crie as classes conforme a descrição abaixo:

a) Crie a classe Aluno com os atributos privados: nome, do tipo String, idade, do tipo int, peso, do tipo double, formando, do tipo boolean e sexo, do tipo char.

b) Crie o construtor da classe Aluno que recebe parâmetros para inicializar os atributos nome, idade, peso e sexo. O atributo formando deve ser inicializado automaticamente com falso.

c) Crie os métodos de acesso (Get) para todos os atributos.

d) Crie os métodos modificadores (Set) para todos os atributos.

e) Crie a classe TesteAluno com um método Main.

f) No método Main:

- Peça para o usuário entrar com os valores necessários para criar um aluno, instancie um objeto Aluno, depois imprima todos os atributos do objeto Aluno usando os métodos de acesso (Get) de Aluno.
- Depois, pergunte novamente ao usuário os valores da idade, do peso e se o aluno é formando. Altere estes atributos com os parâmetros informados por meio dos métodos modificadores (Set).
- Por fim, imprima novamente os valores dos atributos do objeto Aluno usando os métodos de acesso (Get) de Aluno

2) Crie uma classe Cliente, que represente os clientes de um banco. A classe deve ter os seguintes atributos:

- codigo (tipo int)
- nome (tipo string)
- eClienteEspecial (tipo bool) //Informa se o cliente pertence a categoria Cliente Especial
- limiteCredito (tipo double) //Limite de crédito do cliente
-

a) Escreva um método construtor para a classe. O construtor deve receber como parâmetro o código e o nome do cliente. No construtor o atributo limiteCredito deve receber zero, e o atributo eClienteEspecial deve receber false.

b) Escreva um método GetNome. O método não deve ter parâmetros e deve retornar o atributo nome.

c) Escreva um método AtualizarLimite que atualize o limite de crédito do cliente. O método deverá receber como parâmetro um valor do tipo double. Caso o cliente seja especial, o seu limite de crédito deverá ser atualizado com o valor recebido como parâmetro e o método deverá retornar true, indicando que a operação foi realizada. Caso contrário, o método deverá retornar false.

d) Escreva um método AtualizarCategoria que atualize a categoria do cliente (o seu atributo eClienteEspecial). O método deverá receber como parâmetro um boolean. Além de atualizar o referido atributo com o valor recebido como parâmetro, se este valor for false, o atributo limiteCredito deverá ser atualizado com o valor 0 (zero). Visto que, somente Cliente Especiais podem ter crédito.

e) Crie uma classe Teste que contenha o método Main. No Main, crie três clientes e utilize todos os métodos da classe Cliente.

3) Crie as classes conforme a descrição abaixo:

- Crie a Classe Turma, com atributos privados código (int), e ano (int). Crie um construtor que receba parâmetros para inicializar os atributos e os métodos de acesso e métodos modificadores (Gets e Sets).
- Crie a classe Aluno descrita no **exercício 1**. Adicione nesta classe um atributo privado turma do tipo Turma. Altere o construtor para receber um parâmetro que inicialize o novo atributo e crie o método de acesso e o modificador para este novo atributo.
- Crie métodos ToString em ambas as classes. O método não deve ter parâmetros e deve retornar uma String com os valores dos atributos.

- Crie a classe **TesteAluno** com método **Main**. Instancie um objeto **Aluno**, e utilize os métodos **setIdade**, **setPeso** e **ToString**.
4. Em uma agenda telefônica os contatos são cadastrados com os seguintes dados:
- nome: nome do contato;
 - celular: cadeia de caracteres com o número do celular do contato;
 - email: cadeia de caracteres com o e-mail do contato;
 - aniversario: data de aniversário do contato (contendo dia e mês).

Escreva um programa em C#, para simular uma agenda telefônica. O programa deve apresentar um menu com as seguintes operações:

1. Inserir um novo contato
2. Buscar um contato pelo nome
3. Imprimir todos os dados de todos os contatos da agenda telefônica
4. Encerrar programa

Para tanto, crie as seguintes classes

Classe **Data**, com atributos dia e mês. Crie um construtor, métodos **Get** e **Set**.

Classe **Contato**, com atributos nome, celular, email, aniversario (tipo **Data**). Crie um método construtor que receberá como parâmetro: nome, celular, email dia do aniversário e mês do aniversário. Crie um outro método construtor que receberá como parâmetro: nome, celular, data do aniversário (tipo **Data**). Crie os métodos **Get** e **Set**.

Classe **AgendaTelefonica**, com atributos: agenda (tipo vetor de **Contatos**), quant (tipo **int**). O atributo quant deve armazenar a quantidade de contatos que estão na agenda. Assim, esse atributo deverá ser incrementado a cada nova inserção. A classe **AgendaTelefonica** deverá ter os seguintes métodos:

- Método construtor: deverá receber o tamanho da agenda (**int**). Deve ser criado o vetor agenda e o atributo quant deve ser inicializado com zero.
- InserirContato: o método deverá receber um **Contato** como parâmetro e deverá ser inseri-lo no vetor agenda, caso ainda tenha espaço disponível na agenda.
- BuscarContato: o método deverá receber um nome (**string**) como parâmetro e deverá retornar um objeto **Contato**, caso o nome conste na agenda. Caso o nome não conste na agenda deverá ser retornado o valor **null**.
- Imprimir agenda: método sem parâmetros e sem retorno. O método deverá imprimir os dados de todos os contatos da agenda.

Classe **Teste**, deverá ter o método **Main**. No **Main** deve ser instanciado um objeto **Agencia Telefonica** com tamanho 50. O método deverá apresentar o menu de opções para o usuário. A opção escolhida pelo usuário deverá ser lida e executada. O programa deverá repetir esses passos até que a opção 4 seja escolhida pelo usuário.