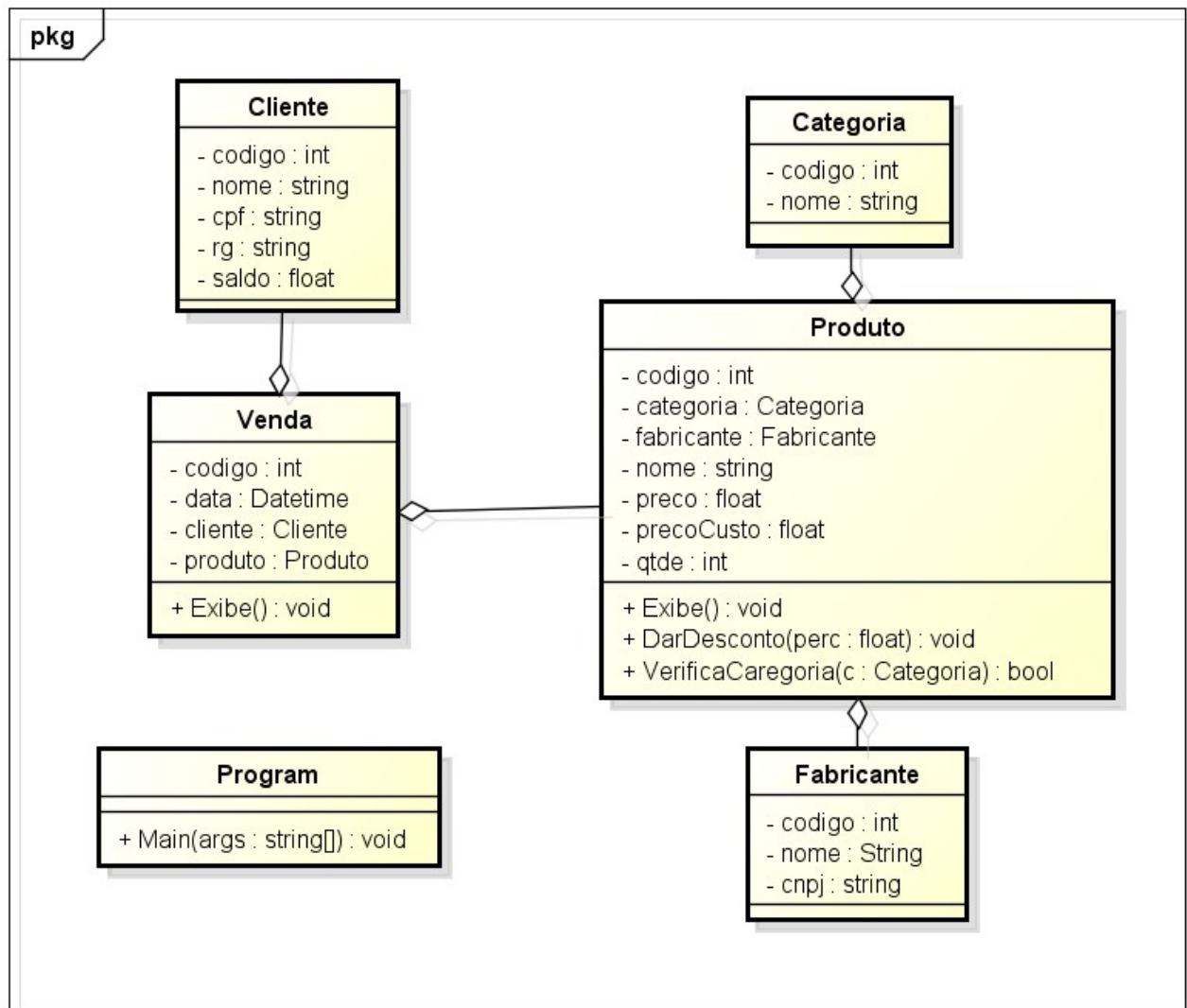


Crie um projeto no Visual Studio.NET (em C#) chamado
VENDA-RA<SEU_NUMERO_RA>

Segue abaixo um diagrama de classes (UML):



Crie as classes conforme foram especificadas acima.

O sinal de "-" significa modificador de acesso **PRIVATE**.

O sinal de "+" significa modificar de acesso **PUBLIC**.

Crie *accessors* para TODOS OS ATRIBUTOS, mesmo os que você não usará.
(Use o propfull tab tab!)

Crie um construtor para cada uma das classes. Os construtores devem receber como parâmetro valores que carregam automaticamente todos os atributos/propriedades dos objetos, exceto:

- construtor Produto()
 - recebe todos menos o qtde e o preço do produto.
 - **qtde** deverá ser automaticamente 10.
 - **preço** deverá automaticamente o precoCusto informado com acréscimo de 20%.
- construtor Venda()
 - recebe todos menos data, que deve ser automaticamente DateTime.Now;
 - Deve subtrair em 1 a qtde contida no objeto Produto recebido pelo construtor.

Métodos do Projeto

- classe Venda
 - Exibe(): Não deve receber nem retornar informação alguma, apenas mostrar na tela as seguintes informações:

VENDA

CODIGO: [CODIGO DA VENDA]
DATA: [DATA DA VENDA]

PRODUTO: [NOME DO PRODUTO]
CATEGORIA: [NOME DA CATEGORIA]
FABRICANTE: [NOME DO FABRICANTE]
CNPJ: [CNPJ DO FABRICANTE]
VALOR DE CUSTO: [VALOR DE CUSTO]
VALOR DE VENDA: [VALOR DE VENDA]
QUANTIDADE EM ESTOQUE: [QTDE]

CLIENTE: [NOME DO CLIENTE]
CPF: [CPF DO CLIENTE]
RG: [RG DO CLIENTE]

- classe Produto
 - Exibe(): Não deve receber nem retornar informação alguma, apenas mostrar na tela as seguintes informações:

PRODUTO

PRODUTO: [NOME DO PRODUTO]
CATEGORIA: [NOME DA CATEGORIA]
FABRICANTE: [NOME DO FABRICANTE]
CNPJ: [CNPJ DO FABRICANTE]
VALOR DE CUSTO: [VALOR DE CUSTO]
VALOR DE VENDA: [VALOR DE VENDA]
QUANTIDADE EM ESTOQUE: [QTDE]

- void DarDesconto(float perc)
 - Não retorna nada (void) mas recebe um percentual que deve ser aplicado no atributo preco Exemplo: Se o preco (de venda) custa 100 reais e darDesconto(0.10) é enviado, o atributo preco deverá valer 90 reais.
 - Importante: Em hipótese alguma o preço de venda deverá ser menos que o preço de custo. Se um desconto for pedido e isso acontecer, o desconto não será aplicado e com Console.WriteLine() deverá ser enviado ao console uma mensagem de erro.
- bool VerificaCategoria(Categoria c)
 - Serve para simples verificação. O objeto categoria passado como parâmetro (c) deve ser comparado com o atributo/propriedade categoria deste Produto. Se a categoria for a mesma (objetos iguais), retorne **true**, senão false.

TESTES

No método Main() na classe Program:

- Crie dois objetos do tipo Fabricante ("Grow" e "Estrela", informe qualquer código e CNPJ no formato conhecido)
- Crie dois objetos do tipo Categoria: "Jogos de Tabuleiro" e "Bonecas";
- Crie dois objetos do tipo Produto para cada Fabricante: "Boneca Mônica" e "Colonizadores de Catan" (fabricante Grow), "Banco Imobiliário" e "Lu Patinadora" (fabricante Estrela); aplique as categorias adequadas aos produtos; invente todos os outros atributos.
- Aplique um desconto de 10% em "Colonizadores de Catan" e 30% em "Lu Patinadora" (neste último não deve ser possível aplicá-lo, deve dar a msg de erro).
- Chame o método Exibe() dos 4 objetos do tipo Produto.
- Crie dois objetos do tipo Cliente. Você deve ser um deles, e professor o outro. Invente todos os outros atributos.
- Instancie objetos do tipo Venda. Você deve comprar as bonecas e o professor os jogos de tabuleiro.
- Chame o método Exibe() das quatro vendas.