

Prática 7

Atividade 1: Registro de produto de mercado

1. Vamos simular o registro de um produto de supermercado no sistema.
 2. Crie um projeto Java padrão e dê o nome de `RegistroProduto`
 3. Crie uma classe e dê o nome de `Produto`, com as seguintes variáveis de classe:
 1. `nome`: tipo `String` e acesso `private`
 2. `codigo`: tipo `String` e acesso `private`
 3. `estoque`: tipo `int` e acesso `private`
 4. `preco`: tipo `float` e acesso `private`
 4. Defina os métodos `getters` e `setters` (os métodos de atribuição e de pegar valores das variáveis de classe):
 1. `setNome`: recebe um valor do tipo `String` como parâmetro e retorna `void`; esse método apenas atribui o valor recebido como parâmetro na variável de classe `nome`
 2. `getNome`: sem parâmetros e retorna `String`; esse método apenas retornar o valor da variável de classe `nome`
 3. `setCodigo` e `getCodigo`, `setEstoque` e `getEstoque`, `setPreco` e `getPreco`
 5. Na `main`, peça para o usuário registrar um produto no programa:
 1. pedir nome, código, quantidade em estoque e preço do produto
 6. Depois, mostre na tela todas as informações do produto registrado
-

Atividade 2: Simular um *software* de envio de e-mail

1. Vamos simular a construção de um e-mail
2. Crie um projeto Java padrão e dê o nome de `EmailApp`
3. Crie uma classe e dê o nome de `Email`, com as seguintes variáveis de classe:
 1. `assunto`: tipo `String` e acesso `private`
 2. `destinatario`: tipo `String` e acesso `private`
 3. `remetente`: tipo `String` e acesso `private`
 4. `corpo`: tipo `String` e acesso `private`
4. Na `main`, vamos simular a preparação do e-mail
5. Peça para o usuário digitar o e-mail dele
6. Depois, peça para o usuário digitar o destinatário do e-mail.
7. Depois, peça para o usuário digitar o assunto.
8. E, por fim, peça para digitar o corpo do e-mail.
9. Apresente o e-mail completo na tela.

Atividade 3: Simular placar de jogo de basquete

1. Vamos simular um placar de jogo de basquete
2. Crie um projeto Java padrão e dê o nome de `PlacarBasquete`
3. Crie uma classe e dê o nome de `Placar`, com as seguintes variáveis de classe:
 1. `pontuacaoMandante`: tipo `int` e acesso `private`
 2. `pontuacaoVisitante`: tipo `int` e acesso `private`
 3. `faltasMandante`: tipo `int` e acesso `private`
 4. `faltasVisitante`: tipo `int` e acesso `private`
4. Defina os seguintes métodos na classe `Placar`:
 1. `getters` e `setters` para todas as variáveis de classe (métodos para atribuir e pegar os valores das variáveis de classe)
 2. `cesta2PontosMandante`: sem parâmetros e sem retorno; esse método soma 2 pontos à pontuação do time mandante
 3. `cesta3PontosMandante`: sem parâmetros e sem retorno; esse método soma 3 pontos à pontuação do time mandante
 4. `cesta2PontosVisitante`: mesma coisa, mas para o time visitante
 5. `cesta3PontosVisitante`: mesma coisa, mas para o time visitante
 6. `faltaDoMandante`: sem parâmetros e sem retorno; esse método adiciona 1 falta ao time mandante
 7. `faltaDoVisitante`: sem parâmetros e sem retorno; esse método adiciona 1 falta ao time visitante;
5. Na `main`, simule um jogo de basquete:
 1. Instancie um objeto da classe `Placar`
 2. Escreva algumas linhas de código para simular uma partida de basquete, somando alguns pontos e faltas para ambos times usando os métodos da classe `Placar`.
 3. Depois, mostre o placar final da partida na tela.

Atividade Extra: Desafio

Faça um programa parecido com o anterior, mas agora simulando um Placar de jogo de Futebol, marcando número de gols de ambos times, número de faltas e números de cartões amarelos e vermelhos.