

Exercícios : **Programação Orientada a Objetos : Classes**

- 1) Criar uma classe chamada Complexo. Ela possui como atributos as partes real (float) e imaginária (float).

Construtor

- Sem parâmetros;
- Construtor com a parte real e imaginária como parâmetros;

Propriedades

- Permitir consultar a parte real e a parte imaginária.

Métodos

- Método que permite alterar os valores do número complexo (recebe 2 floats como parâmetros que devem ser atribuídos aos atributos real e imaginário)
- Método que retorna o módulo do número complexo ($\sqrt{\text{real}^2 + \text{imaginário}^2}$).

Programa Teste

Criar, na classe Program, um objeto da Complexo que recebe um número complexo e mostra o módulo do número.

- 2) Criar uma classe chamada Retângulo. Ela possui como atributo a altura (float) e largura (float).

Construtor

- Sem parâmetros;
- Construtor com a altura e largura como parâmetros;

Propriedades

- Permitir alterar os valores da altura e da largura (alterar e consultar os valores).

Métodos

- Método que devolve a área do retângulo ($\text{Área} = \text{largura} * \text{altura}$);
- Método que devolve o perímetro do retângulo ($\text{Perímetro} = 2 * \text{altura} + 2 * \text{largura}$).

Programa Teste

Criar, na classe Program, um objeto da classe Retângulo. Atribuir os valores da altura e da largura e imprimir a área e o perímetro.

- 3) Criar uma classe chamada Hora. Ela possui como atributo a hora (byte), minuto (byte) e segundo (byte).

Construtor

- Método construtor sem parâmetros;
- Método construtor com os 3 valores;

Propriedades

- Consultar os valores individuais dos atributos.

Métodos

- Método que atualiza o próximo segundo (13:45:45 --> 13:45:46);
- Método que atualiza o segundo anterior (13:45:45 --> 13:45:44);
- Método que adiciona um determinado número de segundos passado como parâmetro (13:45:45 + 125 segundos = 13:47:50);
- Método que devolve o tempo decorrido a partir de um objeto da classe Hora.
- Método que devolve o tempo em segundos corridos

Programa Teste

Criar, na classe Program, dois objetos classe Hora que representem dois horários. O programa deve utilizar os métodos implementados.

- 4) Criar uma classe chamada Conta. Ela possui como atributos um número de conta (int), o nome do dono da conta (string) e o saldo (float).

Construtor

- Método construtor sem parâmetros;

Propriedades

- Permitam alterar os atributos (alterar e consultar os atributos).

Métodos

- Método depositar que acrescenta um valor depositado ao saldo;
- Método sacar que recebe um valor a ser sacado e retorna true se operação bem **sucedido**.

Programa Teste

Criar, na classe Program, um objeto conta com um saldo inicial de 1000. Após deve ser sacado 200 e depositado 500 reais. Por fim, imprimir o saldo da conta.

- 5) Considere uma classe chamada Ponto, que armazena um ponto no espaço (coordenadas X e Y). Ela possui como atributos a coordenada X (int) e a coordenada Y (int).

Construtor

- Método construtor sem parâmetros;
- Método construtor com parâmetros (coordenada X e coordenada Y);

Propriedades

- Permitam alterar os atributos (alterar e consultar os atributos).

Métodos

- Substitua o método ToString(), o qual retorna uma representação do objeto em formato de string, para que retorne o valor dos atributos da classe no formato "(X,Y)".

```
public override string ToString() {}
```

Programa Teste

Criar, na classe Program, três objetos da classe Ponto. Atribuir valores para as coordenadas X e Y de cada um destes objetos e imprimir os dados na tela.

- 6) Com base na classe Ponto implementada no exercício 5, escreva uma classe chamada FiguraGeometrica, que possui como atributo três objetos Ponto (p1, p2, e p3). Crie os construtores e as Propriedades get/set que achar necessárias

Esta classe também possui os seguintes métodos:

- Calcular a distância entre dois pontos **float CalcDistancia(Ponto a, Ponto b)**, onde, sendo (X1,Y1) o ponto **a** e (X2,Y2) o ponto **b**, a fórmula para o cálculo da distância entre dois pontos é: $distancia = \sqrt{(X_1 - X_2)^2 + (Y_1 - Y_2)^2}$. Este método não deve estar disponível para o usuário (deve ser private).
- Método que retorna o tipo do triângulo em relação aos lados: **string TipoTriangulo(Ponto a, Ponto b, Ponto c)**. Os tipos de triângulo, em relação aos lados são: **Equilátero** (três lados iguais), **Isósceles** (dois lados iguais) e **Escaleno** (três lados diferentes).
- Método Retornar o tipo do triângulo em relação aos ângulos: **string TipoAngulos(Ponto a, Ponto b, Ponto c)**. Os tipos de triângulo em relação aos ângulos, sendo **a** o maior lado, são:
 - o $a^2 = b^2 + c^2$, o triângulo é **Retângulo**;
 - o $a^2 > b^2 + c^2$, o triângulo é **Obtusângulo**;
 - o $a^2 < b^2 + c^2$, o triângulo é **Acutângulo**.

Programa Teste

Altere a classe Program. Crie um objeto da classe FiguraGeometrica e inicialize este objeto com os objetos Ponto que você já possuía. Após escreva o tipo do triângulo em relação aos lados.

- 7) Criar uma classe chamada Estatística para realizar algumas análises estatísticas de uma série de números alimentados individualmente. Antes de definir a classe, vamos analisar as equações para calcular a média μ e o desvio padrão σ :

$$\mu = \frac{1}{N} \sum_{i=1}^N x_i$$

$$\sigma = \sqrt{\frac{\sum_{i=1}^N x_i^2 - \left(\sum_{i=1}^N x_i\right)^2 / N}{N-1}}$$

onde N é o número de amostras e x_i é o i-ésimo elemento. Observando as equações dadas, podemos concluir que não é necessário armazenar as amostras, mas o somatório das amostras ($\sum_{i=1}^N x_i$) e o somatório do quadrado das amostras ($\sum_{i=1}^N x_i^2$). Com estes dois somatórios, conseguimos estimar a média e o desvio padrão.

Com esta informação, a classe deve possuir os seguintes atributos:

- amostras (int): armazena o número de amostras inseridas até o momento;
- somatorioX (float): armazena o somatório das amostras inseridas;
- somatorioX2 (float): armazena o somatório do quadrado das amostras inseridas;

Construtor

Esta classe não necessita de um construtor, pois os dados serão inseridos posteriormente.

Propriedades

Deve ser implementada uma única propriedade: Amostras. Esta propriedade deve retornar o número de amostras (atributo amostra). Não deve permitir que o atributo seja alterado.

Métodos

- Método **Inserir** que insere um valor (float) passado como parâmetro no objeto. O método deve incrementar o atributo **amostra**, acumular o valor no atributo **somatorioX** e acumular o quadrado do valor no atributo **somatorioX2**;
- Método **Media** que retorna a média (float) dos valores inseridos, com base na fórmula da média μ .
- Método **DesvioPadrao** que retorna o desvio padrão (float) dos valores inseridos, com base na fórmula do desvio padrão σ .

Programa Teste

Faça um programa que gere números aleatórios em um laço e insira valores no objeto criado com a classe Estatística. Após o laço, mostre a média e o desvio padrão estimados pela classe.

Dica

A classe Random permite gerar números aleatórios.

Random rnd = new Random();

No código abaixo, um número aleatório do tipo double é gerado:

double number = rnd.NextDouble();

- 8) Criar uma classe chamada JogodaVelha que implemente uma partida do jogo da velha. Além do atributo do jogo, a classe deve armazenar quantas vitórias para X e para O e o total de partidas (criar propriedades para cada uma destas informações). A classe deve fornecer um método que reinicia o tabuleiro (cada reinício deve incrementar o contador de jogadas). O método jogada deve receber a posição do tabuleiro a ser preenchida e retornar se está ocupado (valor 0), término do jogo por empate (valor 1) ou vitória (valor 2), jogada ok (valor 3). Faça um programa que realize o jogo da velha.
- 9) Criar uma classe chamada Polígono. Ela possui como o número de lados (int), e um vetor que armazena o tamanho de cada lado (float).

Construtor

- Método construtor sem parâmetros;
- Método construtor com o número de lados.

Métodos

- Método que retorna o número de lados;
- Método que recebe um vetor com os tamanhos dos lados (float);
- Método que retorne um determinado lado do polígono;
- Método que calcula o perímetro;
- Método que calcula a área sem código para o cálculo (deve obrigar a classe que herda implementar o método)

Em seguida crie 2 classes que devem herdar da classe Polígono: Triangulo e Quadrilatero. As classes herdadas devem ter as mesmas funcionalidades que a do pai, mas com o detalhe do tipo de polígono. Faça as adequações necessárias.

Programa Teste

Criar, na classe Program, um objeto Triangulo e outro Quadrilatero. Em seguida, inicialize ambos objetos com os tamanhos. Mostre os valores dos tamanhos e a área de cada objeto utilizando um objeto do tipo Poligono.

10) Criar uma classe chamada Data. Ela possui como atributo a dia (byte), mês (byte) e ano (short).

Construtor

- Método construtor sem parâmetros;
- Método construtor com os 3 valores;

Propriedades

- Consultar/alterar os valores individuais dos atributos. Na alteração de dia ou mês, deve verificar se o valor passado é válido (meses de 1 a 12 e dias de 1 ao valor máximo do respectivo mês). Quando houver algum problema, o mesmo não deve ser modificado.

Métodos

- Método que atualiza o próximo dia (15/2/1998 --> 16/2/1998, 31/03/2003 → 1/4/2003); A regra para o cálculo dos anos bissextos:
 - o De 4 em 4 anos é ano bissexto.
 - o De 100 em 100 anos não é ano bissexto.
 - o De 400 em 400 anos é ano bissexto.
 - o Prevalecem as últimas regras sobre as primeiras.

Para melhor entender

- o **São bissextos** todos os anos múltiplos de 400, por exemplo: 1600, 2000, 2400, 2800...
- o **São bissextos** todos os múltiplos de 4, exceto se for múltiplo de 100 mas não de 400, por exemplo: 1996, 2004, 2008, 2012, 2016...
- o **Não são bissextos** todos os demais anos.

Programa Teste

Criar, na classe Program, um programa que lê uma data, cria o objeto da classe Data e mostra o próximo dia com base no método da classe.