Roteiro prático de programação orienta à objetos (POO)

Objetivo: praticar a criação de classes e objetos

Parte I – Criando Classes e Objetos usando C#

Etapa 1 (10 minutos): Pesquise na internet ou youtube sobre o seguinte tema “Visual Studio 2022?”.

Dicas de vídeos:

1) C# Tutorial For Beginners & Basics - 1. Installing Visual Studio 2022 & Quick Tips
https://www.youtube.com/watch?v=67oWw9TanOk&list=PL82C6-O4XrHfoN_Y4MwGvJz5BntiL0z0D

2) C# para iniciantes - Aprendendo a usar o Visual Studio - Parte 1
https://www.youtube.com/watch?v=F8_VFb7Vyvo

Etapa 2 (20 minutos): demonstração sobre criação de projetos para o uso de classes de objetos com o Vusual Studio 2022 feitas pelo professor.

Etapa 3 (20 minutos): criação de projetos no Visual Studio 2022. Siga as etapas abaixo para criação da classe Ponto e TestaPonto conforme a aula teórica.

3.1) Crie a seguinte estrutura de diretórios na sua máquina:

C:\java\<seu_nome>\projetos

Exemplo para o nome do professor Geraldo:

C:\java\geraldo\projetos

3.2) Na IDE crie o projeto nomeado ProjetoPonto e o armazene na estrutura de diretórios criada no item 3.1.

Nome do projeto: ProjetoPonto
Diretório para o projeto: C:\java\<seu_nome>\projetos

3.10 – Adicione uma classe com o nome **Ponto.cs** ao seu projeto, com os atributos x e y que encapsulam as coordenadas de um objeto do tipo Ponto. Os atributos devem ser declarados públicos por enquanto.

3.11 – Inclua na classe Ponto os seguintes métodos:

```
public void incrementarCoordenadas(int deltaX, int deltaY){
   x = x + deltaX;
```

```
    y = y + deltaY;
}
public void imprimirCoordenadas(){
        // imprime as coordenadas x e y
        Console.WriteLine("x={0}, y={1}", x, y);
}
public void zerarCoordenadas(){
    x = 0;
    y = 0;
}
```

3.12 – Acrescente ao método “main” do seu projeto o seguinte código:

```
static void Main(string args[]){

    Console.WriteLine("Início do main...");

    Ponto p1 = new Ponto();

    p1.imprimirCoordenadas();
    p1.incrementarCoordenadas(5, 2);
    p1.zerarCoordenadas();

    p1.imprimirCoordenadas();
    p1.incrementarCoordenadas(5, 2);
    p1.zerarCoordenadas();
    p1.imprimirCoordenadas();

    Ponto p2 = new Ponto();
    p2.imprimirCoordenadas();
    p2.incrementarCoordenadas(6, 9);

    p2.imprimirCoordenadas();
    p2.incrementarCoordenadas(5, 2);
    p2.imprimirCoordenadas();
    p2.zerarCoordenadas();
   Console.WriteLine("Fim do main...");

}
```

3.13 – Teste a criação de outros objetos do tipo Ponto no projeto.

3.14 – Compile, execute e faça outros testes que julgar necessário e aguarde as orientações do professor.

3.15 – Criação e uso de métodos assessores.
Mude a visibilidade dos atributos para privada. Por que isso é importante em OO?
Houve alguma mudança de compilação na classe principal? Quais e por que?

3.16 – Criação e uso de métodos assessores.
Crie os métodos assessores (“gets e sets”) para os atributos. Por que isso é importante em OO?

3.17 – Criação e uso de métodos construtores.
Reveja nos slides da aula teórica o conceito de método construtor. Inclua na classe Ponto o método construtor vazio que inicializa as coordenadas x e y com o valor 0.

3.18 – Criação de método construtor parametrizado.
Crie o método construtor na classe Ponto que inicializa os valores das coordenadas x e y com os valores informados pelo usuário.

3.19 – Inclua no main do projeto o código abaixo que cria os objetos p3 e p4. Execute o projeto e veja o resultado.

```
Ponto p3 = new Ponto(2, 6);
p3.imprimirCoordenadas();
p3.incrementarCoordenadas(6, 9);

p3.imprimirCoordenadas();
p3.incrementarCoordenadas(5, 2);
p3.imprimirCoordenadas();

p3.zerarCoordenadas();

Ponto p4 = new Ponto();
p4.imprimirCoordenadas();
p4.incrementarCoordenadas(6, 9);

p4.imprimirCoordenadas();
p4.incrementarCoordenadas(5, 2);
p4.imprimirCoordenadas();

p4.zerarCoordenadas();
```

3.20 – Crie os objetos p5 e p6 que tem as coordenadas iniciais (10, 20) e (5, 8) respectivamente. Na sequência inclua o código que incrementa a coordenada do ponto p5 dos valores 2 e 3. Faça a respectiva chamada para imprimir as coordenadas do ponto.

3.21 – Inclua na classe Ponto o atributo descricao que armazena uma descrição para cada objeto criado. Exemplo: “p1”, “p2”, etc. Inclua o construtor com 3 parâmetros contemplando o atributo descricao.

3.22 – Crie na classe Ponto o método toString(). Qual a utilidade da implementação do método toString em objetos?

3.23 – Altere o código fonte para permitir que o usuário crie quantos objetos quiser com coordenada diferentes para x e para y. Após solicitar os valores o programa deve instanciar o mostrar os valores dos objetos instanciados. Ao ser informada qualquer coordenada x ou y com valor negativo, o programa deverá ser encerrado.

3.24 – Altere o código fonte para declarar um objeto do tipo ArrayList no método Main da classe principal do projeto e que deverá ser utilizado para guardar todos os objetos do tipo **Ponto** criados pelo usuário. Se for informado um valor negativo para x ou para y, o programa deverá informar o total de objetos que foram instanciados e mostrar o conteúdo de cada objeto instanciado, mostre o resultado da invocação do método *toString* de cada objeto, e armazenado no objeto do tipo ArrayList.