Como discutimos anteriormente, as classes são

um encapsulamento de propriedades de dados e

métodos de dados.

As propriedades são usadas para descrever os

dados que a classe manterá.

Os métodos informam quais são as operações que

podem ser executadas nos dados.

Para entender melhor a classe e os objetos,

vamos praticar com um desafio - abaixo:

de como seria uma classe.

Etapa 0

a partir de um novo projeto crie uma classe com o nome "Treinamentos”.

A classe tem as seguintes propriedades:

IdTreinamento – será usado para armazenar

um número exclusivo que identificaria o

Treinamento.

NomeTreinamento – será usado para armazenar o

nome do treinamento como uma string.

Uma classe também é composta por métodos.

Nossa classe será composta com seguintes métodos:

GetTreinamento – método que sera usado para obter os

dados de um treinamento especifico.

Portanto, se quiséssemos obter o nome de um treinamento,

este método retornaria a string com o nome.

SetTreinamento – método que sera usado para

definir o ID e o nome do Treinamento. Então, se

quisermos criar um treinamento qualquer, podemos

criar um objeto para isso. O objeto teria um ID de valor 1, por exemplo.

Na sequencia, atribuiríamos um nome ao treinamento.

O valor de ID de 1 e o nome do treinamento seriam armazenados

como uma propriedade do objeto.

Etapa 1

criação de uma nova classe, fornecer um nome e adicioná-la ao projeto.

Etapa 2

Construtor C#

Construtores C# são usados ​​para inicializar os valores dos campos de

classe quando seus objetos correspondentes são criados.

Um construtor é um método que tem o mesmo nome da classe. Se um construtor

for definido em uma classe, ele fornecerá o primeiro método que é

chamado quando um objeto é criado. \*\*\*ESTE É SOMENTE UM EXEMPLO: Suponha que tivéssemos uma classe chamada Funcionario. O método construtor também seria nomeado como Funcionario().

Elementos importantes que precisam ser observados sobre métodos

construtores:

- O modificador de acesso padrão C# para o construtor precisa ser

público - public.

- Não deve haver nenhum tipo de retorno - return - para o método construtor.

Agora, incorpore o construtores em seu código.

Use construtores para inicializar os campos IdTreinamento e

NomeTreinamento com alguns valores padrão quando o objeto for criado.

O primeiro passo é criar o construtor para a classe "Treinamentos."

Etapa 3

classe abstrata em C#

A classe abstrata nunca pode ser instanciada e é marcada pela

palavra-chave abstract. Uma classe abstrata contém zero ou mais métodos

abstratos nela. Atua como uma classe base e é projetada para ser

herdada por subclasses que implementam ou sobrescrevem seu método.

Crie uma classe abstrata em C#

Passo 1) Como primeira na etapa, crie uma classe abstrata - de à ela o nome que achar adequado.

A classe se chamará Treinamento - \*\*\*ESTA É UMA SUGESTÃO - e terá apenas um método abstrato para ser implementado e sobrescrito - de ao método o nome que achar adequado.

Passo 2) Agora, use a classe criada na etapa 1 como classe-filha e o mecanismo de herança deve ser praticado com a classe abstrata. Na classe filha implemente duas propriedades - IdTreinamento, NomeTreinamento e um método para acessar as propriedades e atribuir valor a cada uma delas. Depois, crie um novo método para retornar o valor da propriedade

NomeTreinamento.

Por fim, exiba em tela, uma pequena lista com os ids e nomes dos treinamentos

criados