(dotnet) .NET

**Paradigmas de programação**

Estruturar e executar um programa

Paradigmas de programação

Programação estruturada > um tipo de fim

PE( programação estruturada) vcs.POO (PROGRAMAÇÃO ESTRUTURADA A OBJETOS)

PE> sequência, seleção e repetição

Sequência> comandos de cima para baixo, básico

Seleção, sequencias q só devem ser executadas se uma condição for satisfeita

Repetições: sequências q devem ser executadas repetidamente até uma condição for satisfeita(For, while, do-while)

Programação orientada a objetos (PO)  
Objetos da vida real, procedimentos, métodos, modificar campos

Atributos/métodos.

Programação orientada a objetos

Abstração

Encapsulamento

Herança

Polimorfismo

Orientação a objetos em .NET

Tudo em .NET é objeto

Classes, como se fosse um esboço, molde para um objeto, contendo dentro si as principais informações para sua criação.

Objetos similares fazem parte da mesma classe.

Diagrama de classes: nome de classe, atributos, métodos.

(UML)

String é uma frase (palavra completa)

Date time, data e tempo

Classe publica (acessada qql parte do programa)

Código private (privativo)

Objetos que possuem atributos, comportamentos e um estudo

Produto obj + new Produto () ;

Pública “+”) ;

Privada “-“) ;

Protegida “#).

Visibilidade

Public , sem limitações de acesso

Protected internal acesso somente á classe, que derivam dela e ao próprio montador

Protected acesso limitado a própria classe e as derivadas

Internal, acesso limitado ao próprio assembly

Private acesso limitado a própria classe

Evita confusão do uso de variáveis globais no programa, deixando mais fácil de identificar em qual estado cada variável vai estar a cada momento do programa, já que a restrição de acesso nos permite identificar quem consegue modifica-la.

Valor, referência e ponteiro

Tipos por valor

O C# TEM DUAS CATEGORIAS DE TIPOS POR VALOR E REFERÊNCIA

Inteiros, integers

Implementação

Byte 8 bts inteiro sem sinal (0 255)

Sbyte trabalha com -127 a 128

Ushort inteiro de 16bts sem sinal 0 65 525

Uint

Int inteiro de 32 bts com sinal -2

Ulong inteiro de 64 bts sem sinal

Long inteiro de 64 bts com sinal

Double e float

Ponto flutuante são bastante convencionais, as operações de ponto flutuante não geram erros

Double ponto flutuante 15 digitos de precisão

Float ponto flutuante binário 7 digitais precisão

# (char) ~soa armazenados no padrão Unicode e usam 16 bits por caractere.

Unicote permite armazenar qql tipo de caract

Tipos de referencias, armazena referencia nos seus dados

Duas variáveis podem conter a referencia a um mesmo objeto

Operações podem afetar a outra

Strings semelhantes ao char, aceitam até 1 giga de caracteres

Usada para representar palavras, textos,frases ou textos de um programa

STRINGS

São consideradas imutáveis, não podem ser alteradas depois de criadas.

CLASSES

Tipo definido por um usuário

Podem conter campos, métodos e propriedades.

Podem derivar outras classes

Arrays lista de valor, valores referenciados

Tipo de ponteiro ou apontador tipo de dado refere-se a um tipo de valor, chama o dado na memória através de seu endereço

Métodos

Determina o comportamento e ação de determinada classe, controla o estado de um objeto

Funções que realizam tarefas

Difundidos os getters,setters e constructo

Getters ou métodos assessores solicitam o acesso a informações de um determinado produto sem dar acesso diretamente a ele, colocando ali uma barreira de proteção p/ proteção p dados

Get {return código; }

Setters modificados

Pedido de alteração será verificado p/ ele alterar o determinado atributo ou objeto

Set { código = value; }

CONSTRUCT método construtor, inicializa, da forma á classe

Start qdo vai inicializar determinadas funções

Executado apenas uma vez

Não pode ser chamado diretamente

Tem q ser public

Par q ele possa acessar qql método e atributo

PROPRIEDADES

Dentro das classes são as propriedades

Propriedades consistem em um par de métodos “get” e “set”recuperar o valor do objeto atribuir o valor a um campo

Para cada método existe uma variável

Evetnso são mensagens que a classe dispara em determinadas situação

É necessário q um método seja escrito para ser executado quando ocorrer o evento

A classe apenas fica sabendo q existe esse método em tempo de execução.

DELEGATE> guarda o endereço e chama a função