

# CLASSES AMIGAS MVC -POO

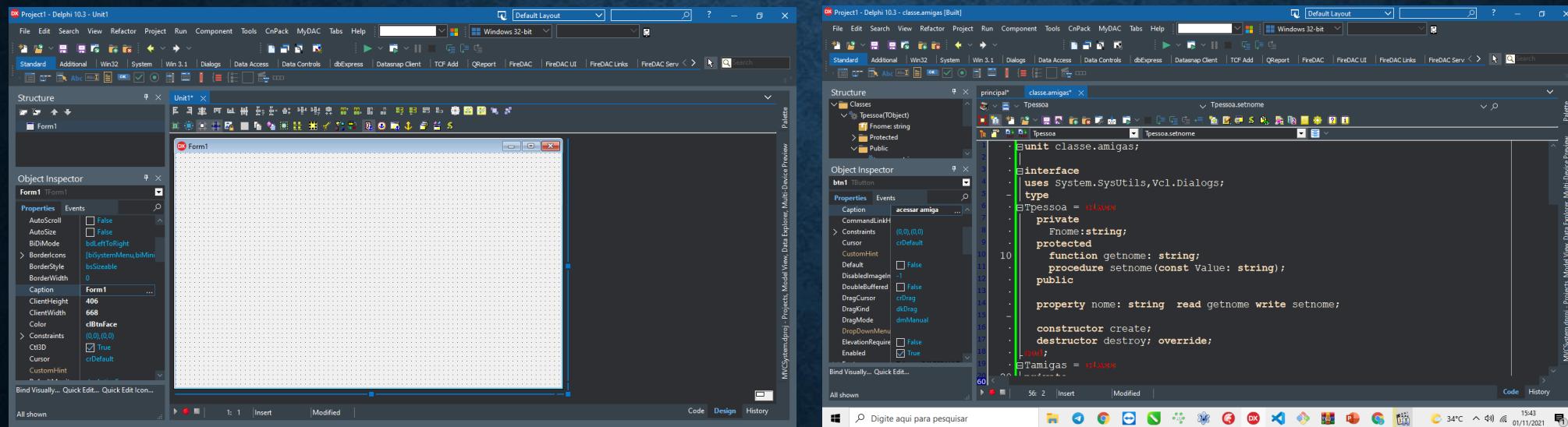


DevTeam

Concretas e Abstratas

# CLASSE AMIGAS É

- É aquela mesmo na sessão private da mesmo unit você pode usar a classe.
- Quando estamos usando um método de uma classe da sessão private na mesmo unit, isso não é correto, Assim temos que criar uma nova unit com o principio da responsabilidade única. Onde cada classe deve assumir suas responsabilidades.
- Veremos isso na prática para exemplificar o caso:



# VAMOS ENTENDER O CASO SIMPLES.

- Tpessoa = class
- Private
- Fnome: string;
- Public
- Property nome: string read getNome write setNome;
- End;
- Tamiga = class
- Private
- Procedure minhaamiga;
- Public



Classes existentes



Classes existentes

Não herda  
de ninguém



Ctrl+Shift+C



DevTeam

# IREMOS FOCAR NA PROCEDURE SET

```
function Tpessoa.getname: string;  
begin  
  Result:= Fnome;  
end;  
  
procedure Tpessoa.setname(const Value: string);  
begin  
  Fnome :=Value;  
end;
```

Aqui está tudo normal



```
procedure Tpessoa.setname(const Value: string);  
var  
amiga:Tami  
begin  
  Fnome type Tamigas: class(TObject);  
  amiga.Fnome :=Value;  
end;
```

Criamos a variável

```
procedure Tpessoa.setname(const Value: string);  
var  
amiga:Tamigas;  
begin  
  Fnome :=Value;  
  amiga.;  
end;
```

Chamamos o método

# ISSO É POSSÍVEL PORQUE AS MESMAS CLASSES ESTÃO NA MESMO UNIT POR ISSO ELAS SE TONAM AMIGAS.

```
procedure Tamigas[minhaamiga];
begin
  showmessage('Olá sou a classe amiga');
end;
```

```
procedure Tpessoa.setnome(const Value: string);
var
  amiga:Tamigas;
begin
  Fnome :=Value;
  amiga[minhaamiga]; //chamada da classe amiga
end;
```

Não esqueça disso.

Aqui nós iremos entender melhor o posicionamento Quando se tratar de classe amigas, porque estamos Pedindo para a mesma um favor de ser usada.

SetNome da classe Tpessoa que é privada!

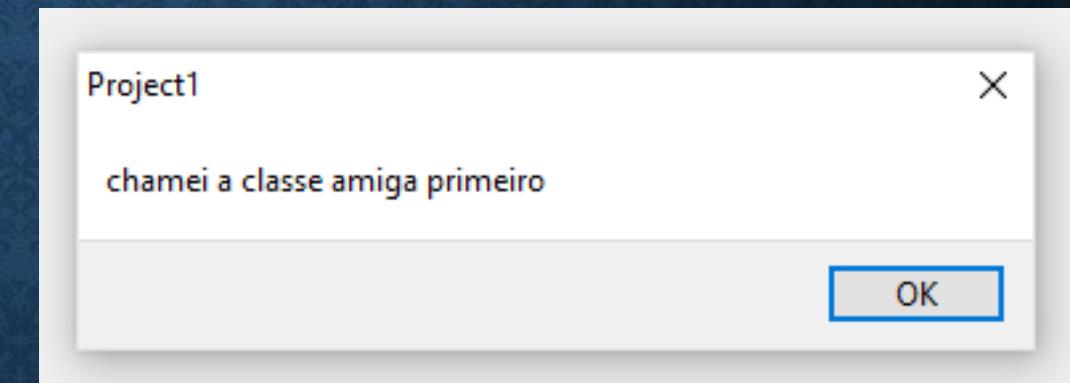
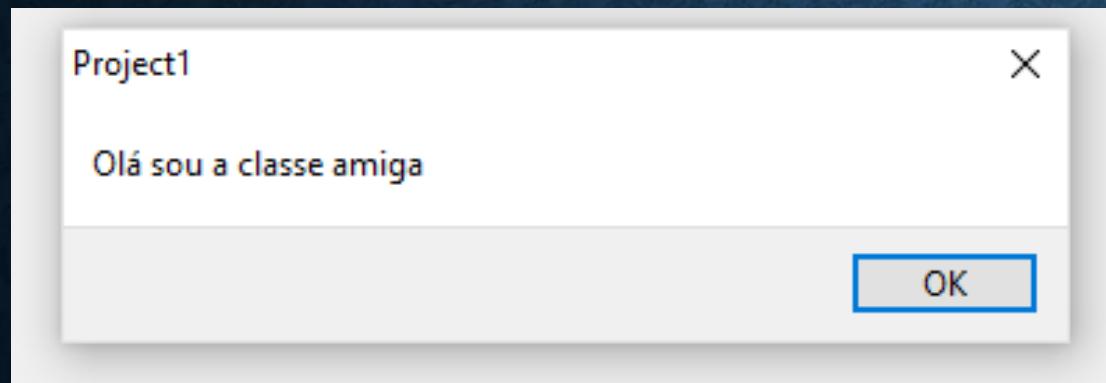


# IREMOS CHAMAR O MÉTODO DA CLASSE AMIGA.

```
procedure TForm1.btn1Click(Sender: TObject);
var
  pessoa:Tpessoa;
  amiga:Tamigas;          ← Chame as classes
begin
  pessoa:=Tpessoa.create;
  amiga:=Tamigas.Create;   ← Instancie as classes( crie)
  try
    pessoa.nome:= 'chamei a classe amiga primeiro';
    ShowMessage (pessoa.nome); | ← Destrua as classes
  finally
    FreeAndNil (amiga);
    FreeAndNil (pessoa);
  end;
end;
```

ATRAVÉS DO MÉTODO DA CLASSE PESSOA

# OBSERVE QUE FOI CHAMADO



O correto é criar a sua própria classe.