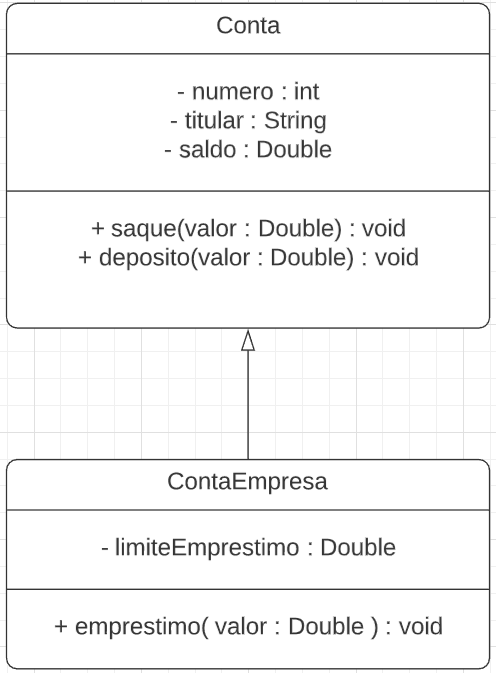
# Upcasting e Downcasting 15/12/2020

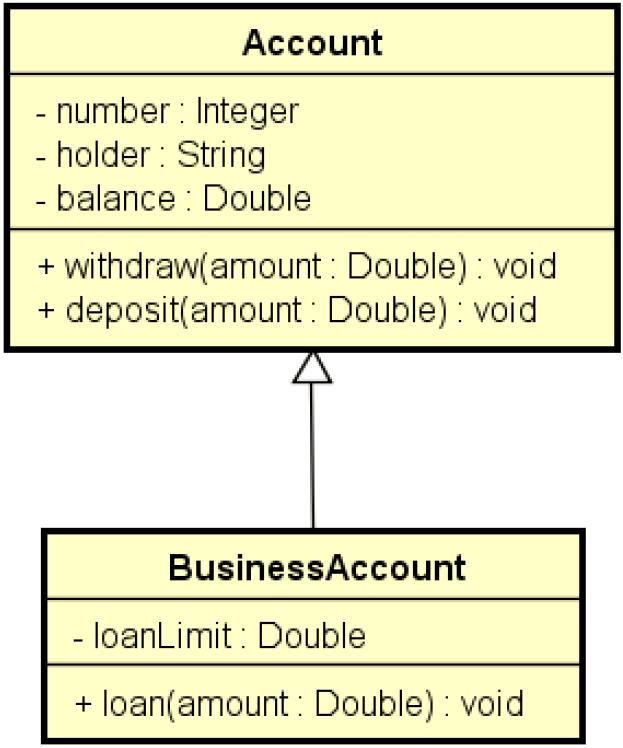
## Upcasting

* Casting da subclasse para superclasse (Casting = converter de um tipo para outro).
  + Seria como converter um objeto do tipo ContaEmpresa para uma variável do tipo Conta.
* Uso comum: polimorfismo (próximas aulas)

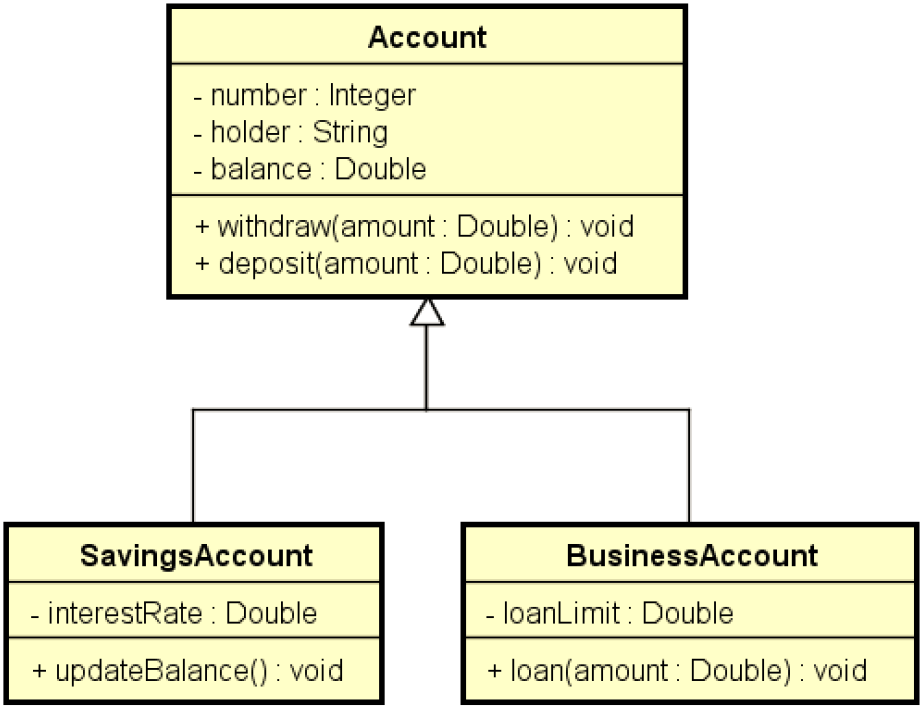
## Downcasting

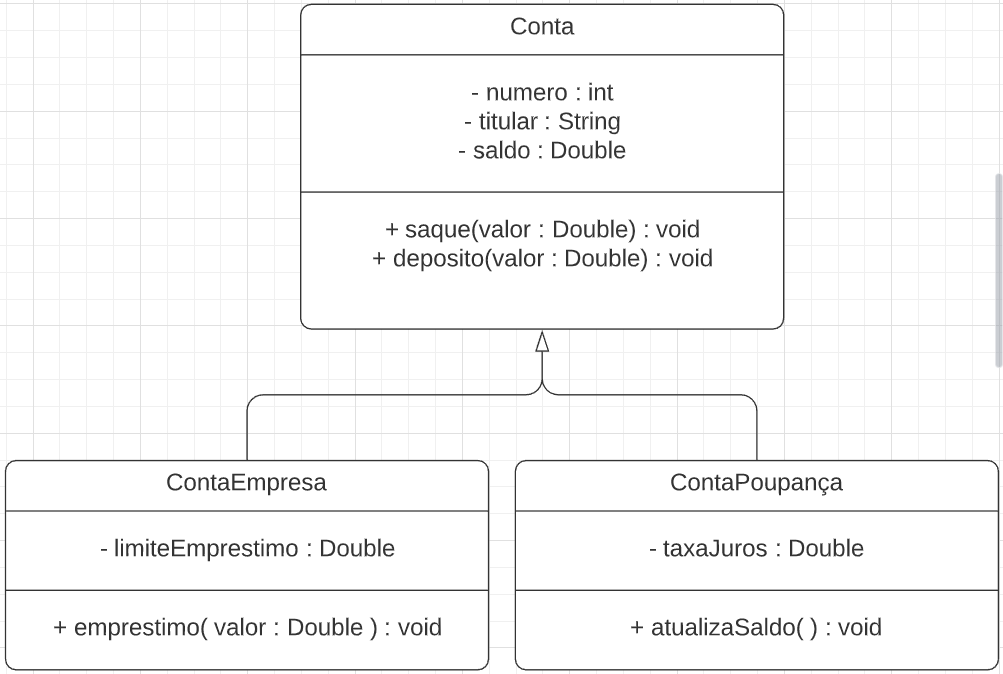
* Casting da superclasse para subclasse
  + Seria como converter um objeto da Conta para uma variável da ContaEmpresa.
* Palavra **as**
* Palavra **is**
* Uso comum: métodos que recebem parâmetros genéricos (ex: Equals – será visto posteriormente)

****

****

**Exemplo:**

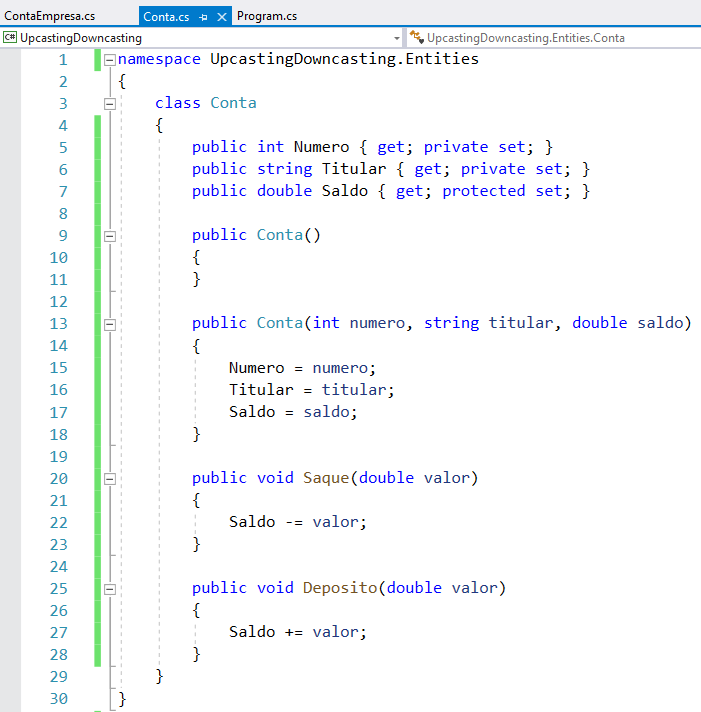
****

****

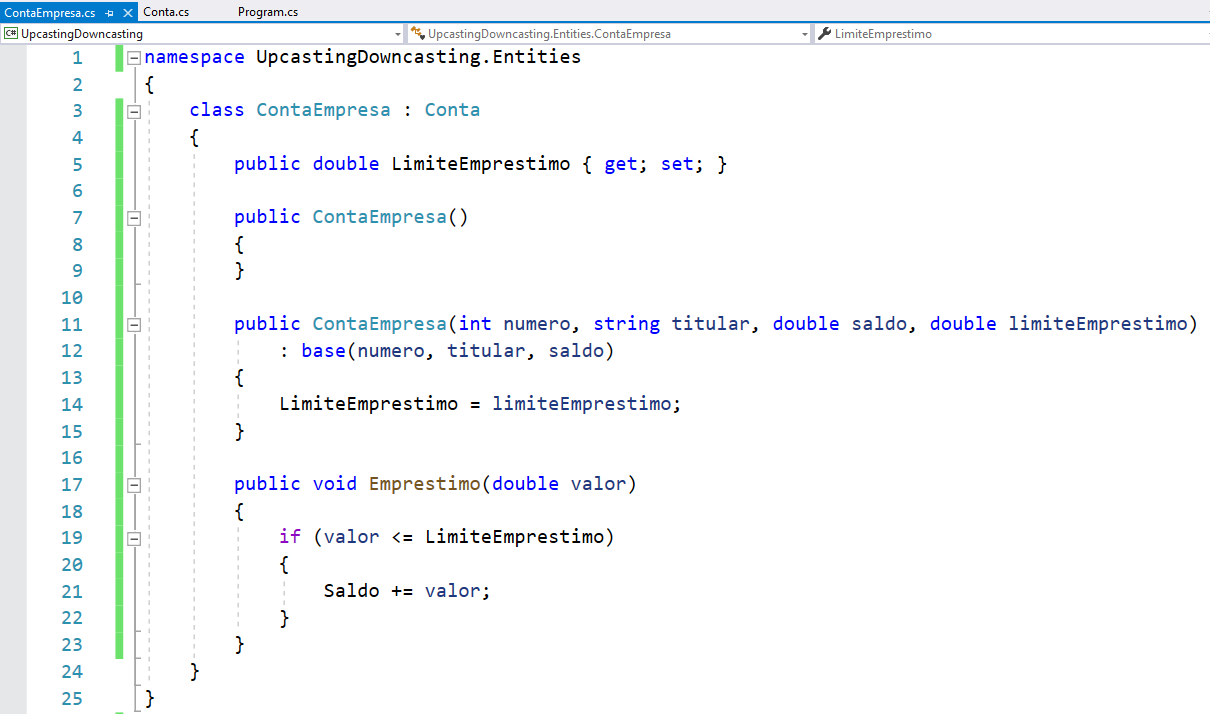
Vamos pegar o código da aula de Herança.

Criei um projeto com o nome UpcastingDowncasting, criei a pasta Entities. Dentro da pasta Entities criei as classes Conta e ContaEmpresa.

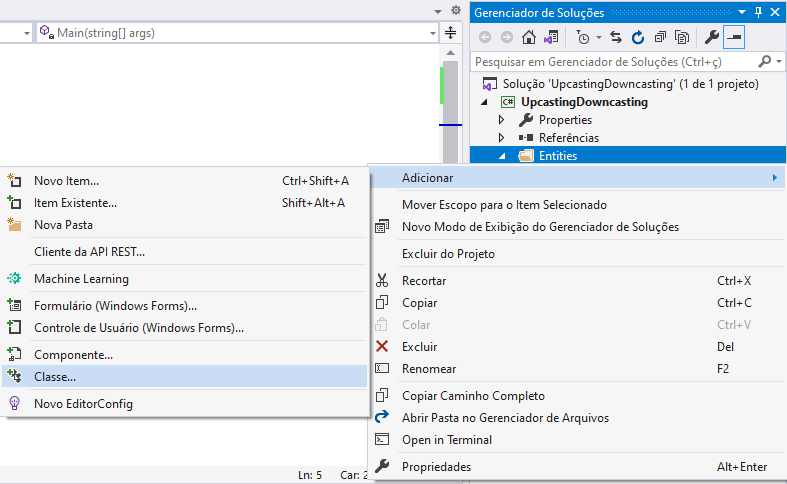
Classe Conta:



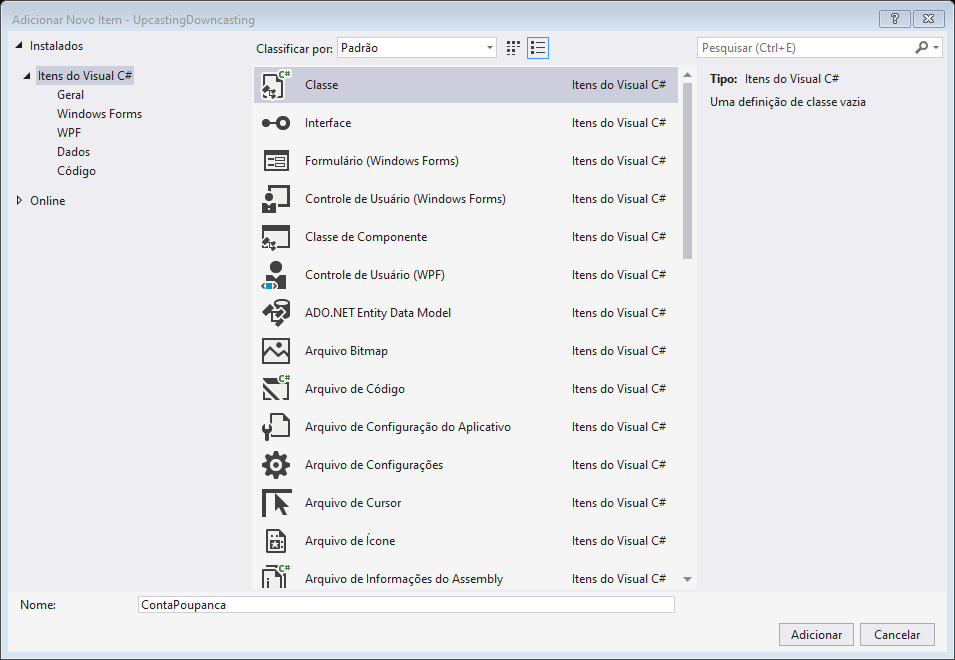
Classe ContaEmpresa:



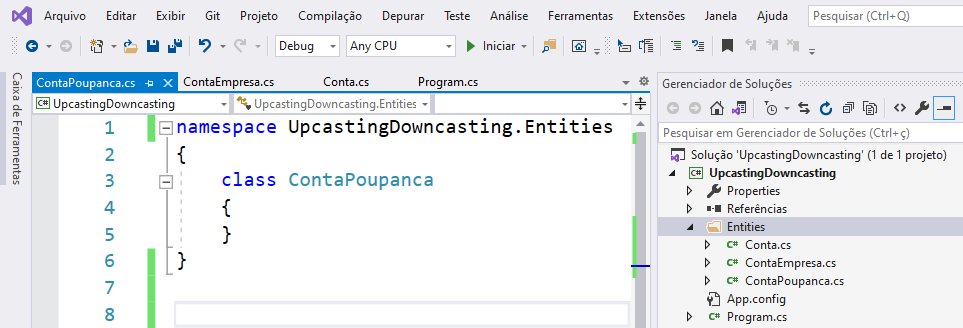
Criando a classe **ContaPoupanca**



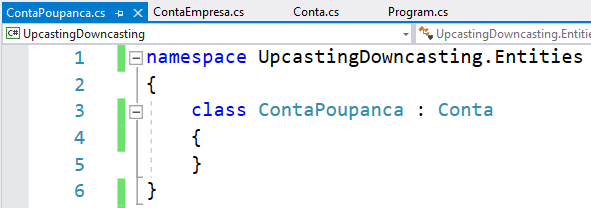
Nome ContaPoupanca



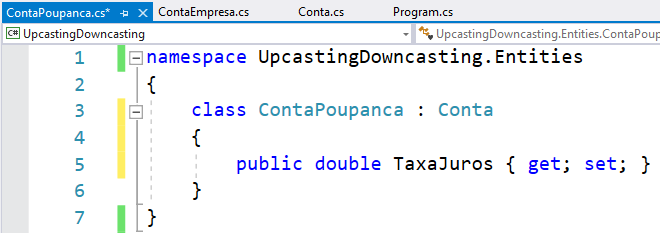
Classe criada.



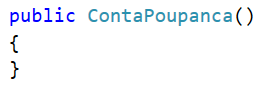
Criando Herança



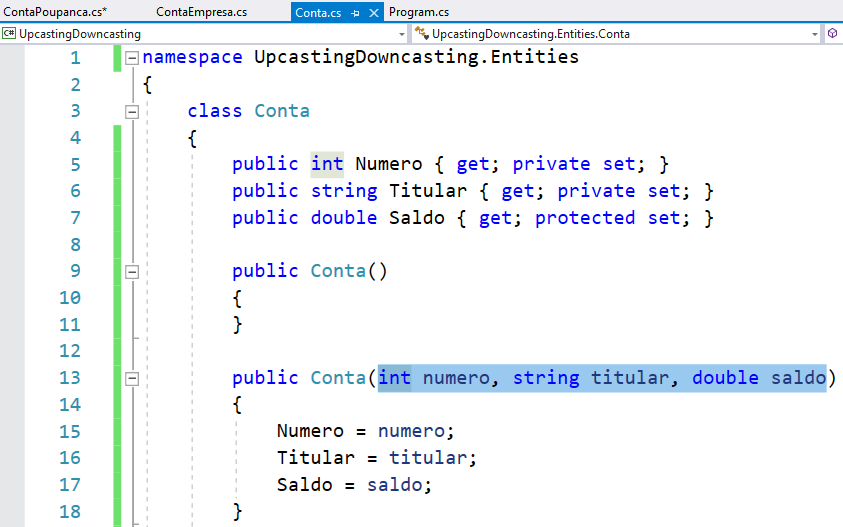
Incluindo a propriedade taxaJuros



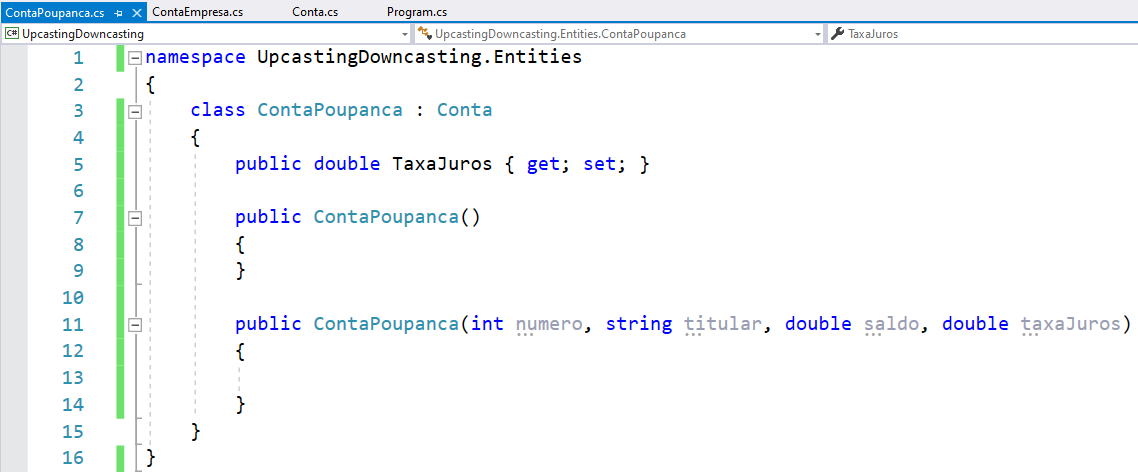
Construtor padrão



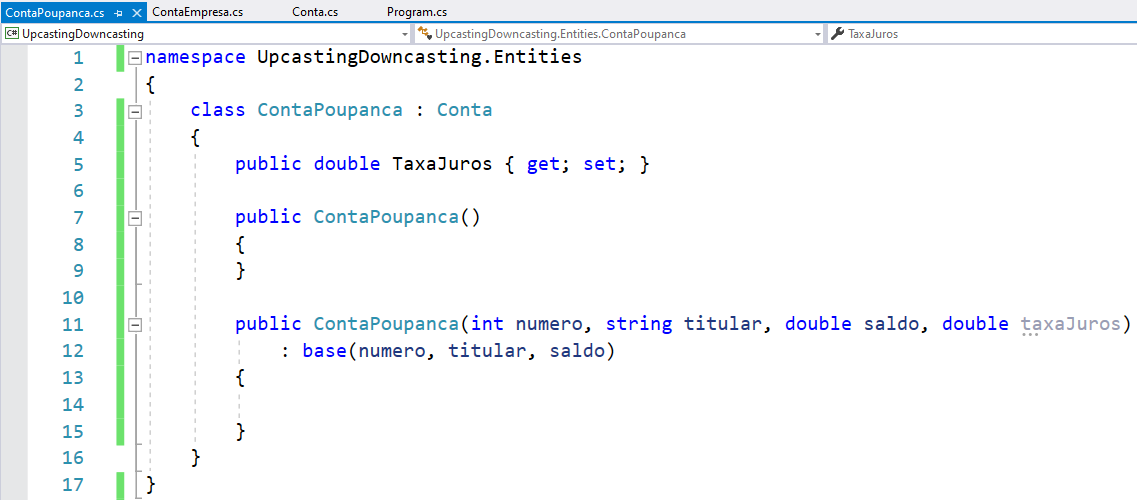
Construtor com argumentos. Para isso, podemos copiar os parâmetros do construtor da classe Conta.



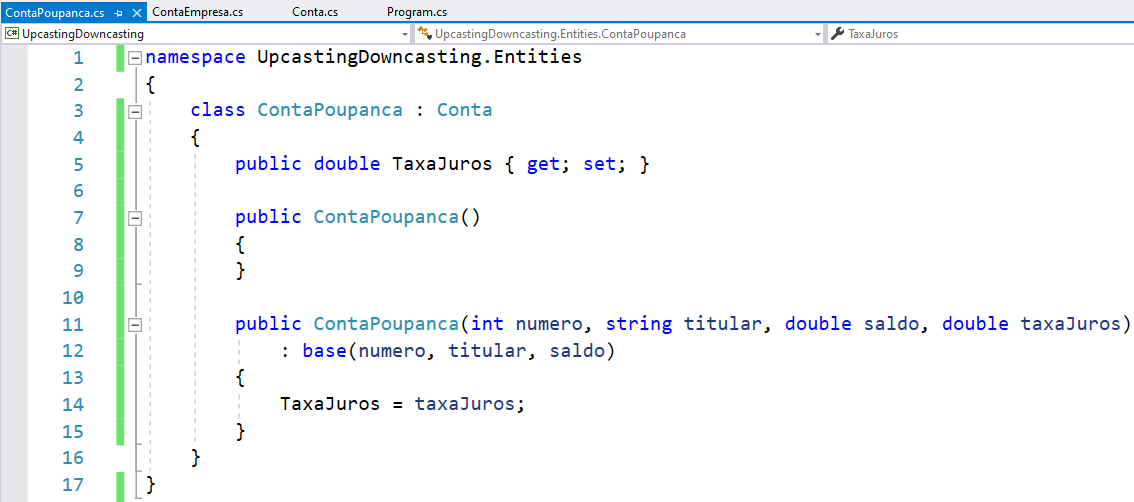
E colar no construtor da classe ContaPoupanca, e acrescentar o parâmetro próprio da classe.



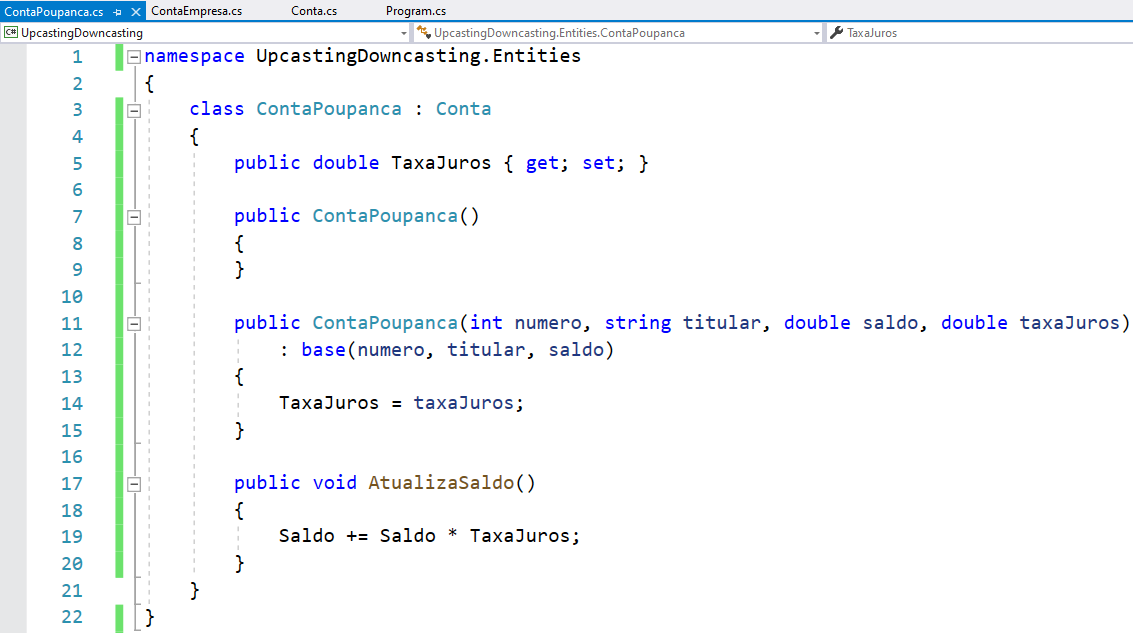
Agora vamos aproveitar o construtor da classe Conta.



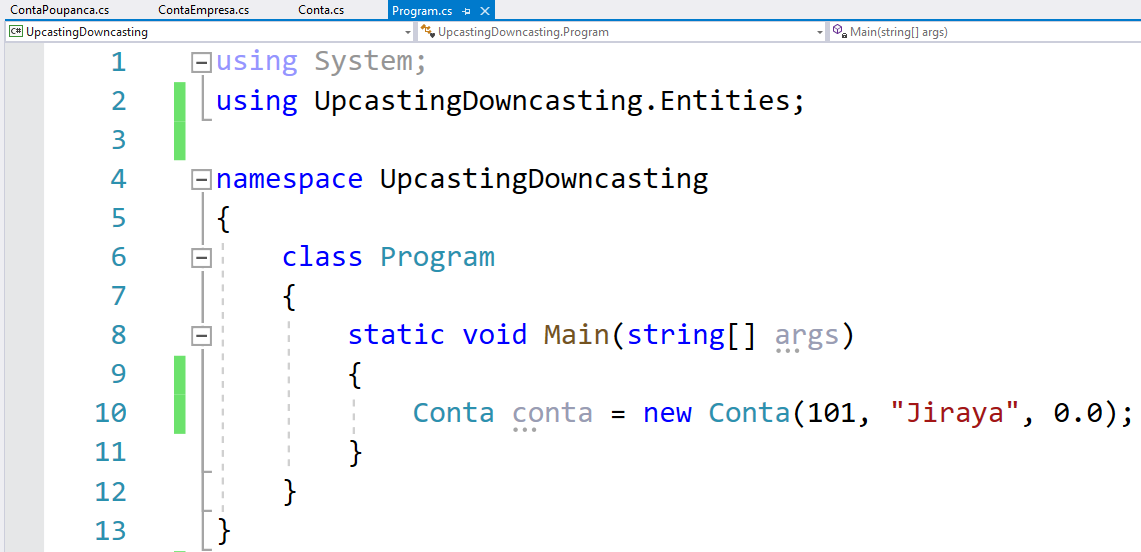
Agora, no corpo do construtor, vamos dizer que o atributo TaxaJuros irá receber o argumento taxaJuros que chegou como parâmetro de entrada.



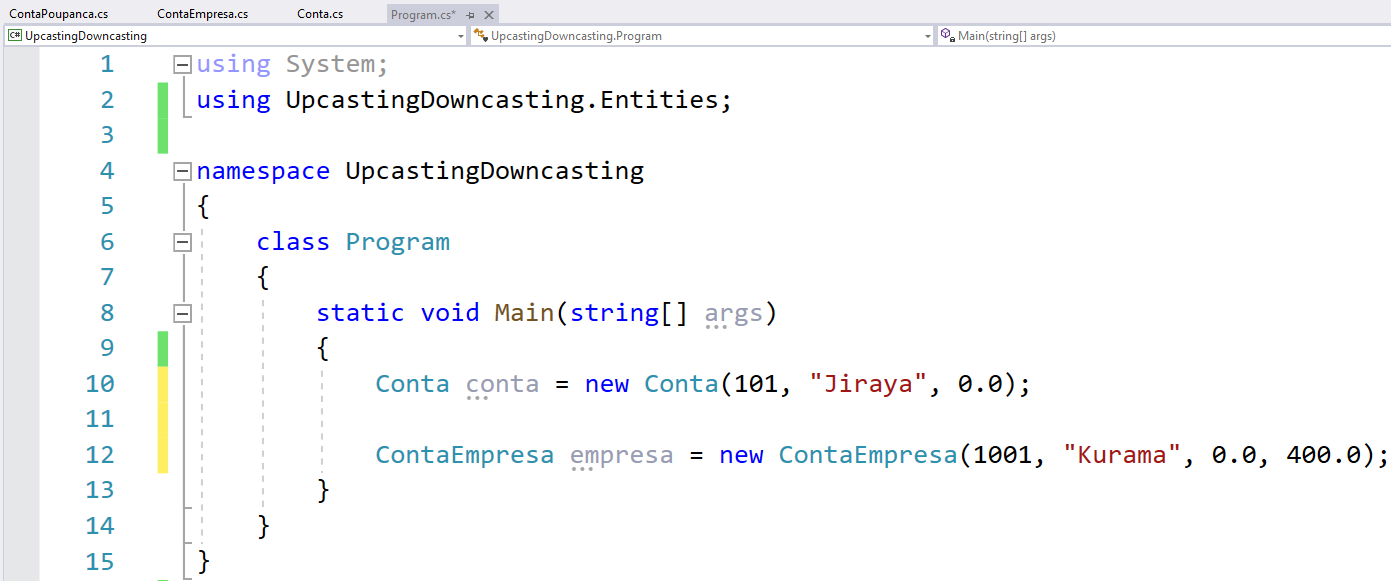
Vamos criar o método AtualizaSaldo. Esse método atualiza o saldo da conta de acordo com a taxa de juros.



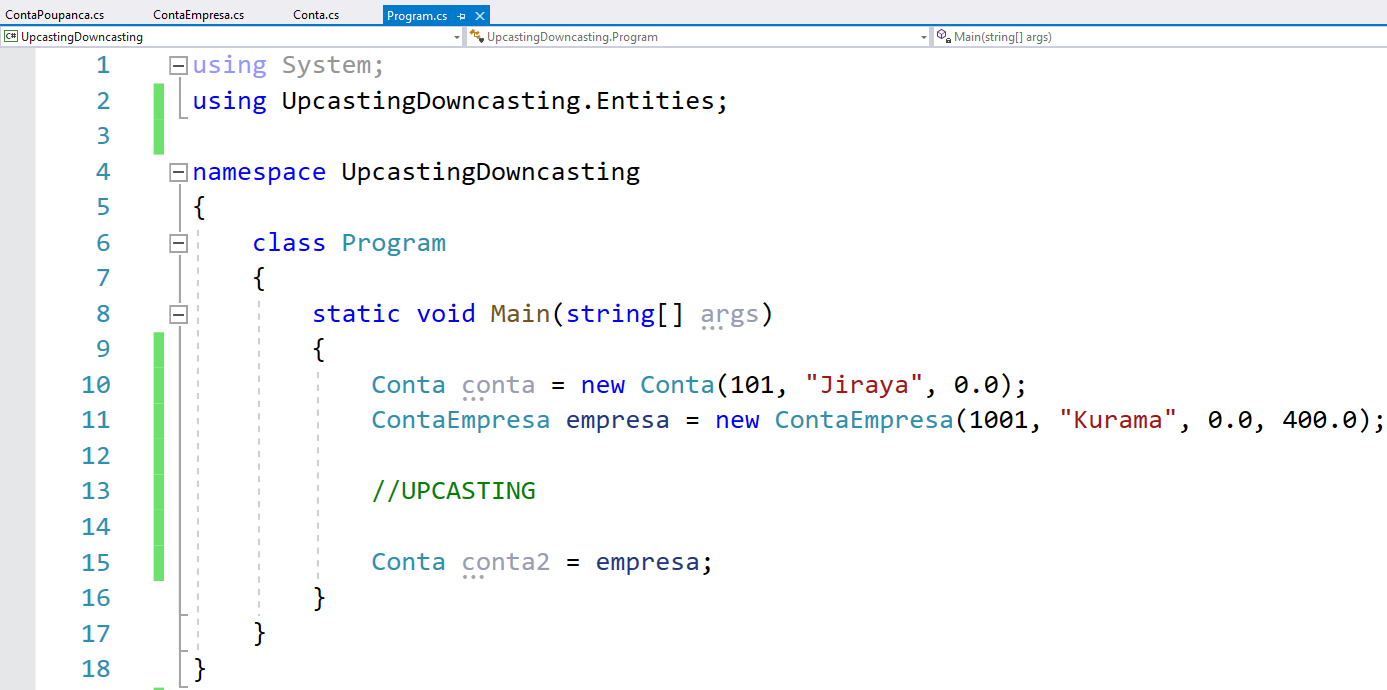
Vamos testar agora no programa principal. Vamos instanciar um objeto do tipo Conta.

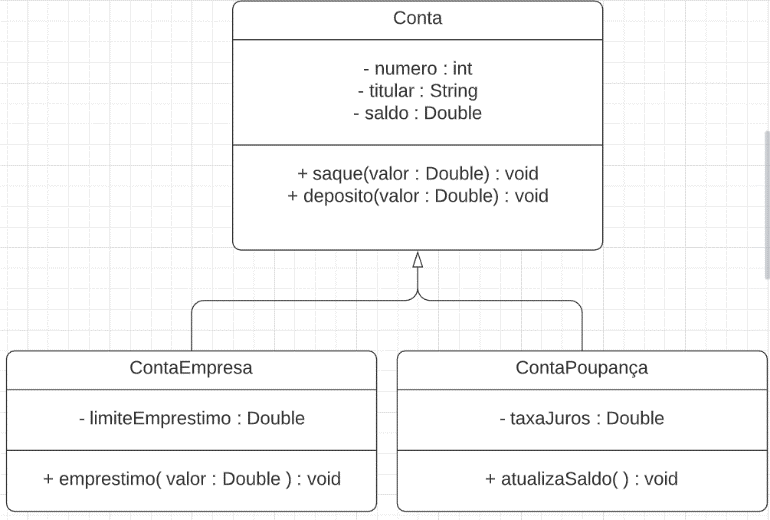


Vamos instanciar um objeto do tipo ContaEmpresa.



Agora sim. Vamos fazer um teste de UpCasting – Uma conversão da **subclasse** ContaEmpresa para uma **superclasse** Conta.



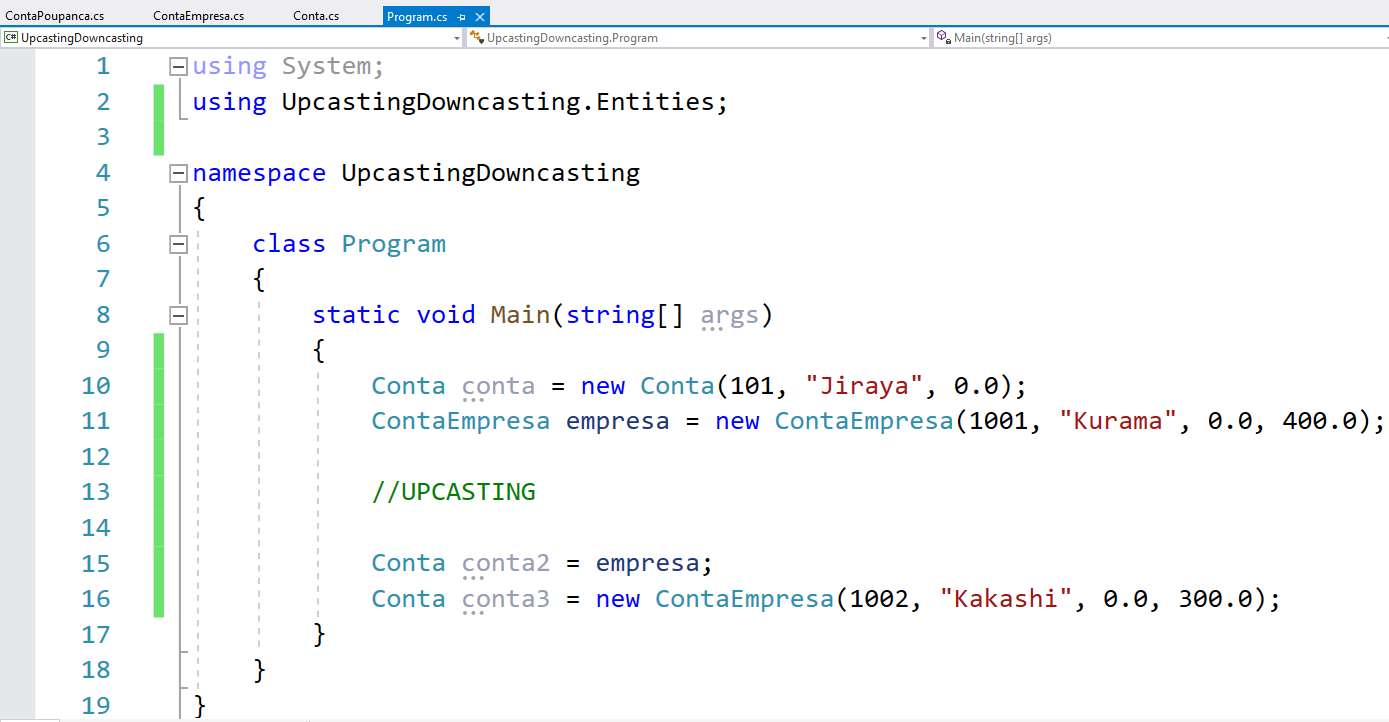
Por conta da condição “**é um**” e a **herança**.

Podemos dizer também que a ContaPoupanca **é uma** Conta.

Podemos dizer que a ContaEmpresa **é uma** Conta.

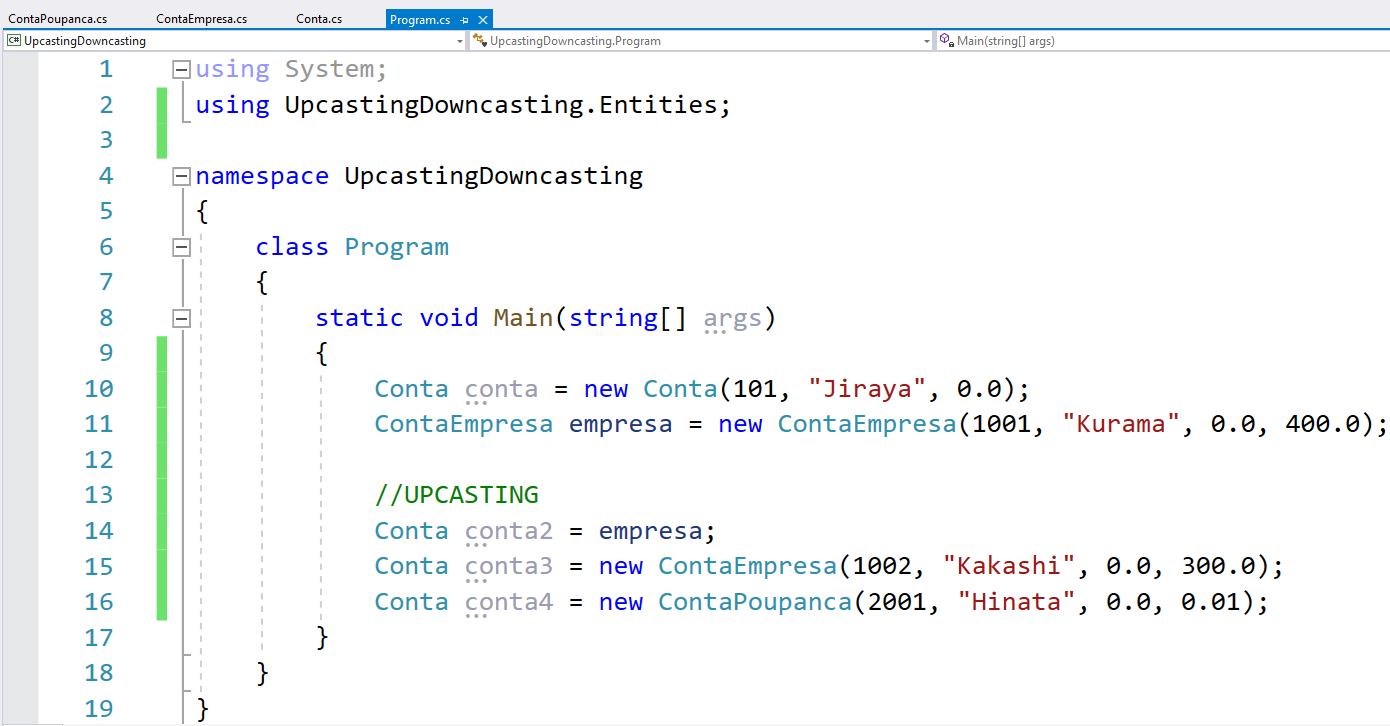
Por isso, podemos fazer a atribuição da linha 15 perfeitamente. Ou seja, uma variável de uma superclasse, aceita qualquer outra de suas subclasses.

Continuando.



A variável conta3 do tipo Conta está recebendo uma nova conta do tipo ContaEmpresa.

Outro exemplo.



# DownCasting

Converter uma **superclasse** em uma **subclasse**.

# GitHub

O código está na subpasta **\Aula\_08\_Classe\UpcastingDowncasting**

https://github.com/endroni/ProgramacaoDeAplicativos.git

# Referência:

Docs Microsoft – Herança em C# .NET – < https://docs.microsoft.com/pt-br/dotnet/csharp/tutorials/inheritance > Acessado dia 10 de dezembro de 2020.

Tutorial W3Schools – Inheritance - < https://www.w3schools.com/cs/cs\_inheritance.asp > Acessado dia 10 de dezembro de 2020.