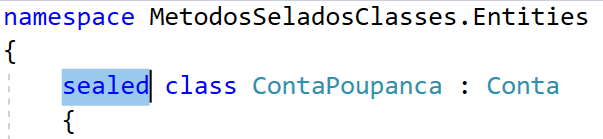
# Classes e métodos selados 16/12/2020

# Palavra chave: sealed (selado)

**Classe sealed**: evita que a classe seja herdada

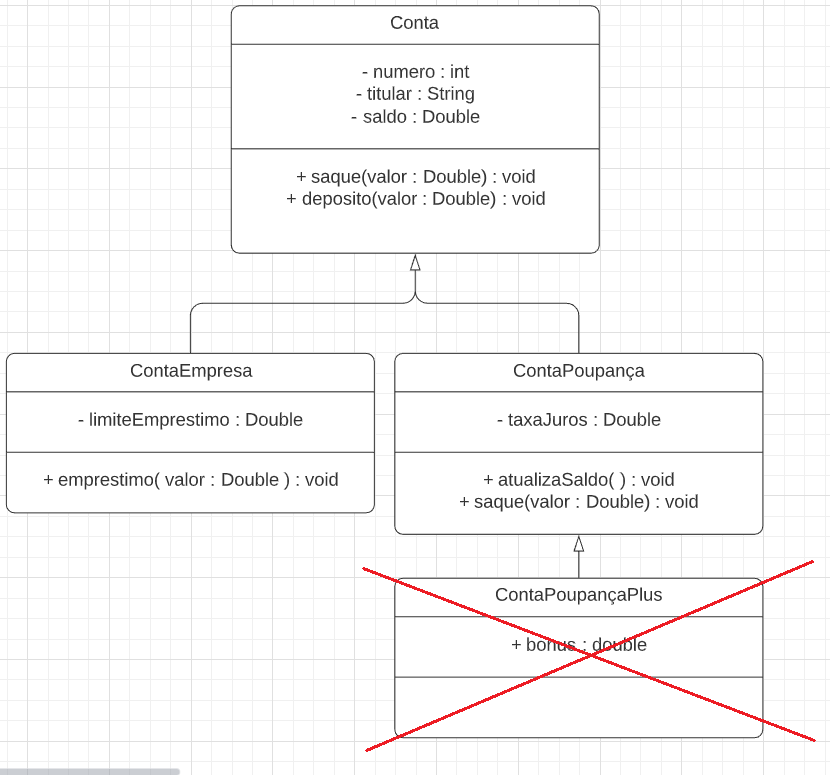
**Nota:** ainda é possível extender a funcionalidade de uma classe selada usando "extension methods"

Para selar uma classe, basta acrescentar a palavra reservada sealed na frente da classe.



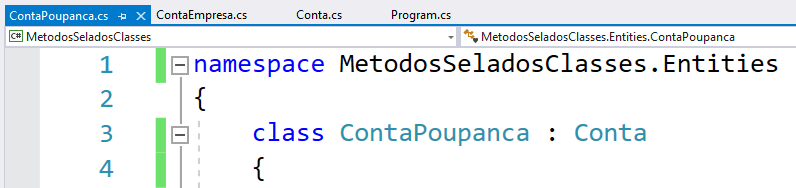
Usando o mesmo exemplo das contas bancárias.

Vamos supor que as regras de negócio não permitem que a classe ContaPoupanca possua uma subclasse. Assim como no projeto abaixo.

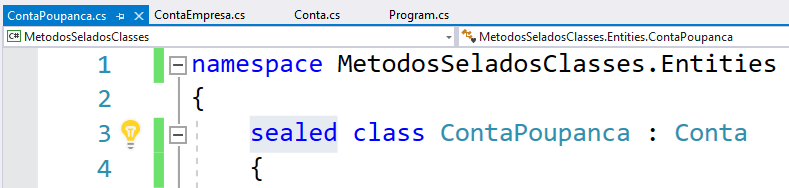


Implementando:

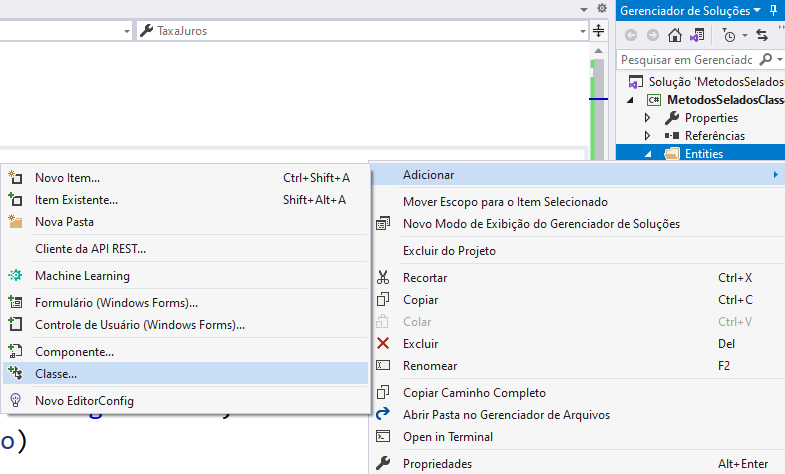
Vamos acrescentar a palavra chave **sealed** no início da classe ContaPoupanca. estava assim:



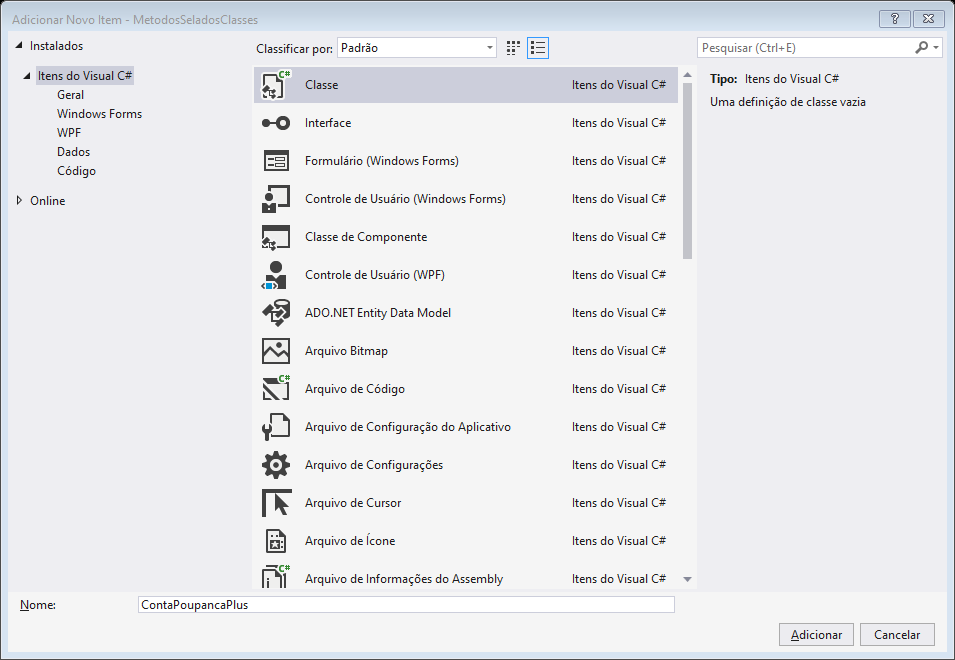
Agora:

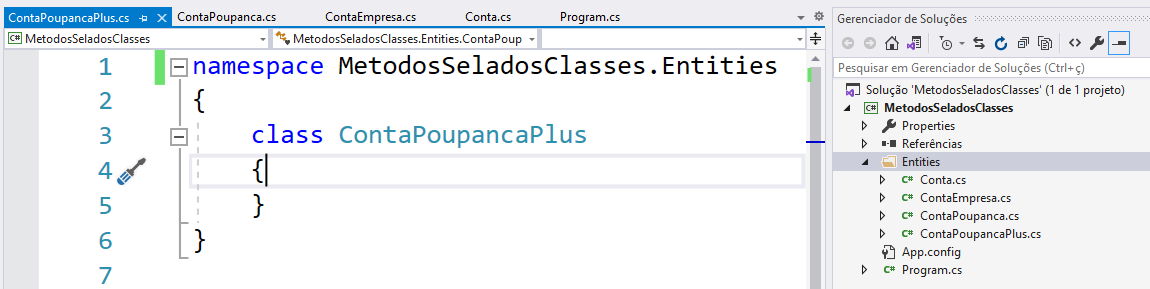


Vamos tentar acrescentar uma classe que na pasta Entities.

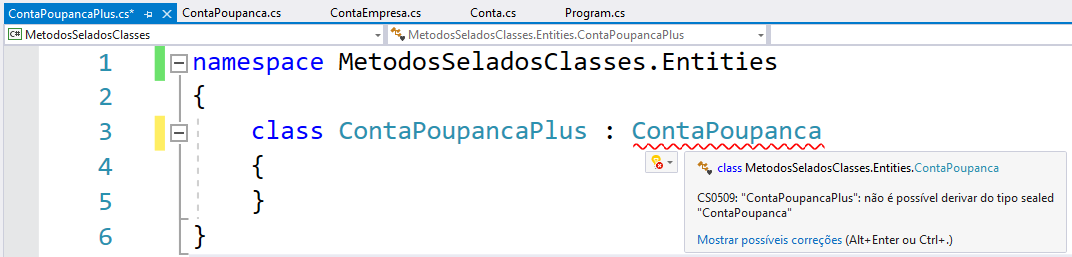


Vamos nomear a classe como ContaPoupancaPlus.



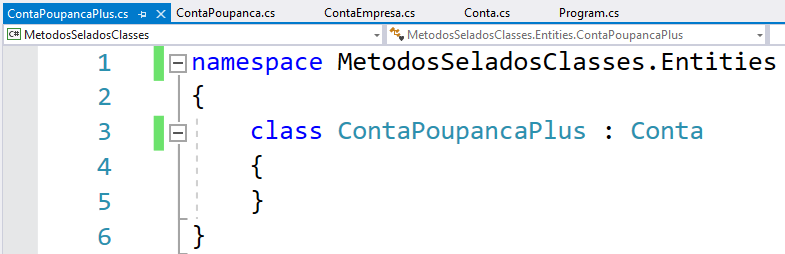


Vamos tentar herdar a Conta agora para ver.



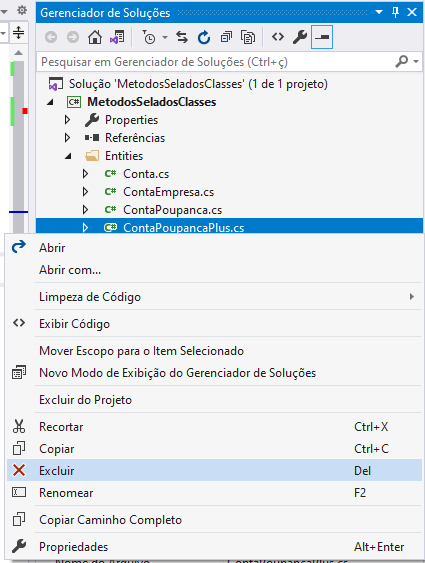
Não é possível derivar, pois a ContaPoupanca está selada.

Observe, que podemos herdar de classes não seladas normalmente.

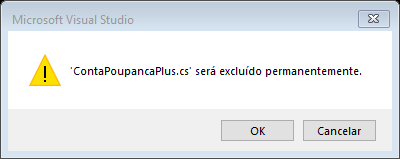


Ou seja, o sealed serve para proteger alguma regra de negócio onde uma classe não pode ser herdada.

Podemos excluir a classe, pois foi criada apenas como teste. Clique com o botão direito na classe ContaPoupancaPlus e em Excluir.



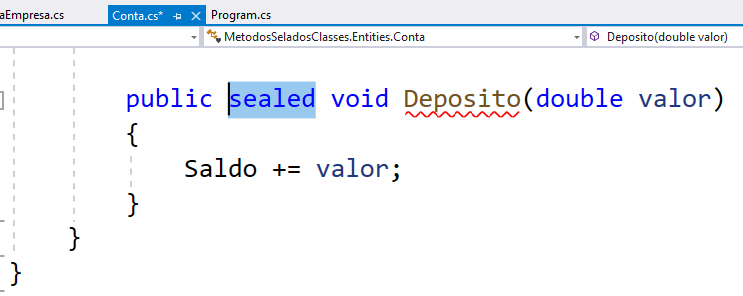
Confirme clicando em Ok



# Método selado

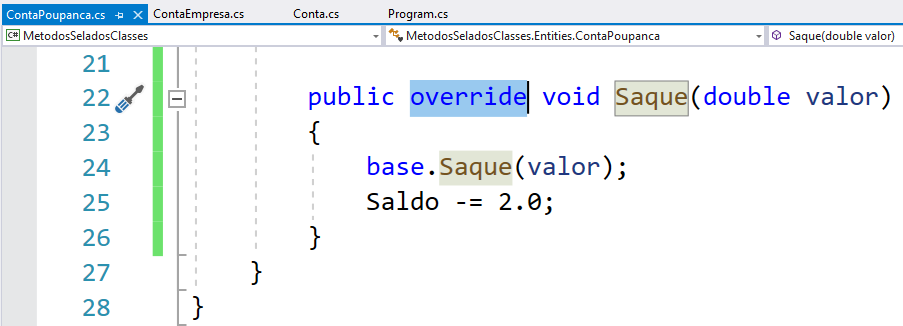
**Método**: evita que um método sobreposto possa ser sobreposto novamente.

Só pode ser aplicado a métodos sobrepostos. Exemplo:

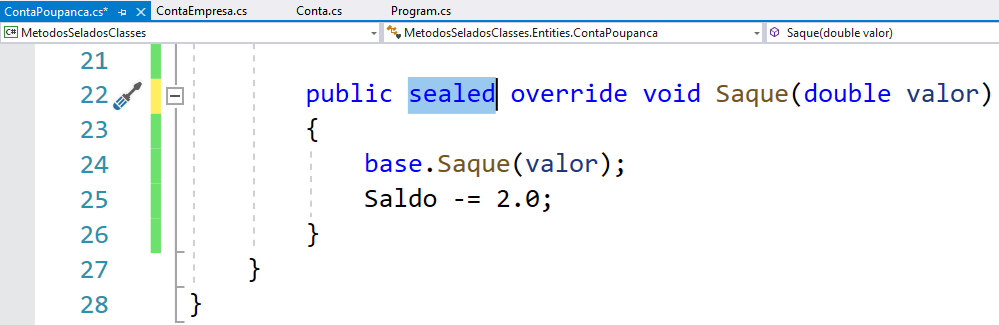


Como o método Deposito não foi sobreposto, ele não pode ser selado.

Já o método Saque da classe ContaPoupanca é uma sobreposição. Portanto, ele pode ser selado.

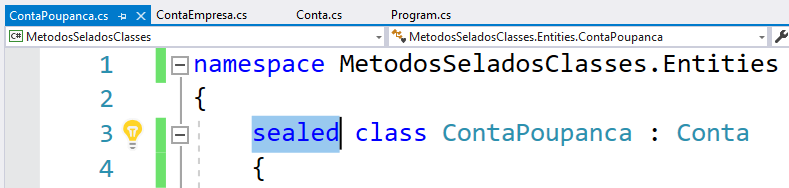


Sealed método Saque

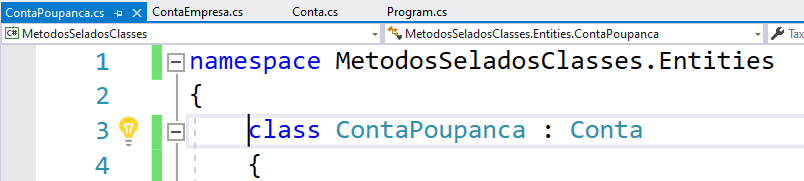


Estamos explicitando que esse método não pode ser sobreposto em uma outra classe.

Exemplo: Vamos retirar o sealed na classe ContaPoupanca

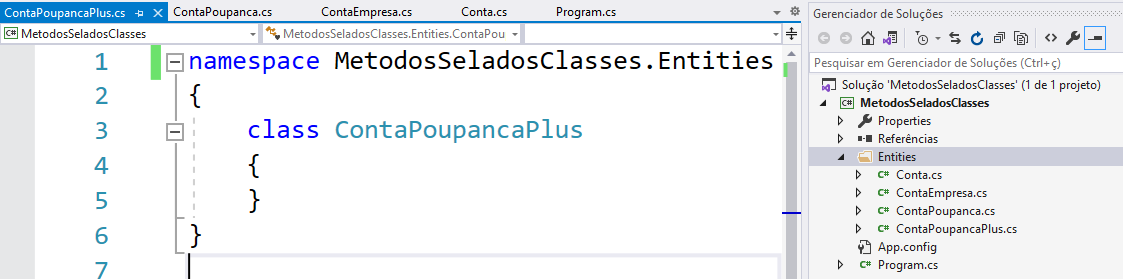


Retirado:

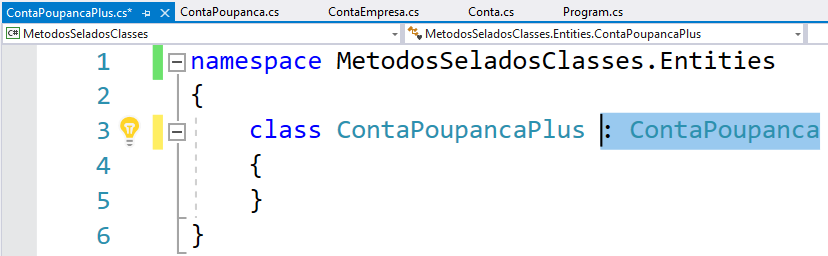


Estamos dizendo que a classe ContaPoupanca pode ser herdada, mas como o método Saque foi selado, ele não poderá.

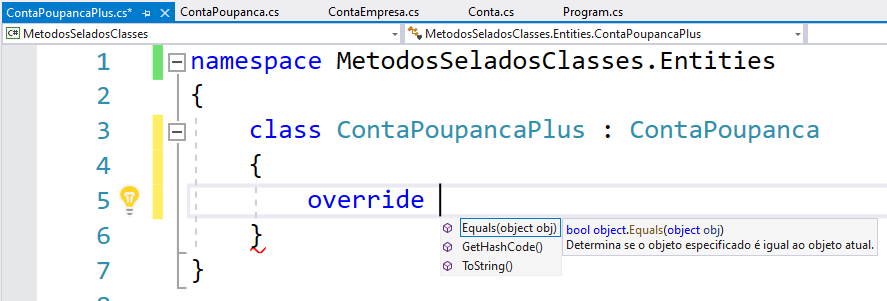
Vamos recriar aquela classe ContaPoupancaPlus para testar.



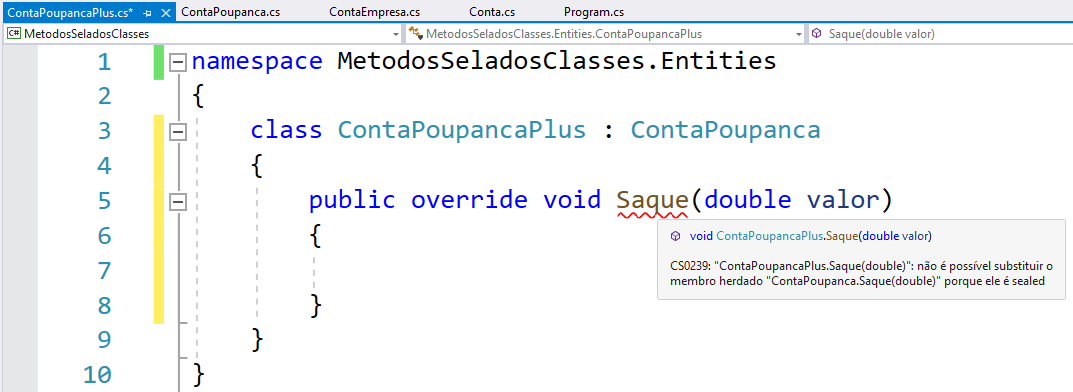
Vamos dizer que essa classe irá herdar na classe ContaPoupanca



Porém ao escrever override, não aparece o método Saque.



Mesmo tentando criar o método, o compilador não irá permitir sobrescrever um método selado.



# Conclusão

## Quais os motivos para se selar uma classe?

* Segurança: dependendo das regras do negócio, às vezes é desejável garantir que uma classe não seja herdada, ou que um método não seja sobreposto.
* Geralmente convém selar métodos sobrepostos, pois sobreposições múltiplas podem ser uma porta de entrada para inconsistências.
* Performance: atributos de tipo de uma classe selada são analisados de forma mais rápida em tempo de execução.

# GitHub

O código está na subpasta:

**\Aula\_08\_Classe\** **MetodosSeladosClasses**

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

# Referência:

Docs Microsoft – sealed (Referência de C#) – < https://docs.microsoft.com/pt-br/dotnet/csharp/language-reference/keywords/sealed > Acessado dia 15 de dezembro de 2020.

Tutorial W3Schools – Polimorfismo - < https://www.w3schools.com/cs/cs\_polymorphism.asp > Acessado dia 15 de dezembro de 2020.