# Sobreposição – Palavra Virtual – Override – Base

**16/12/2020**

# Sobreposição ou sobrescrita

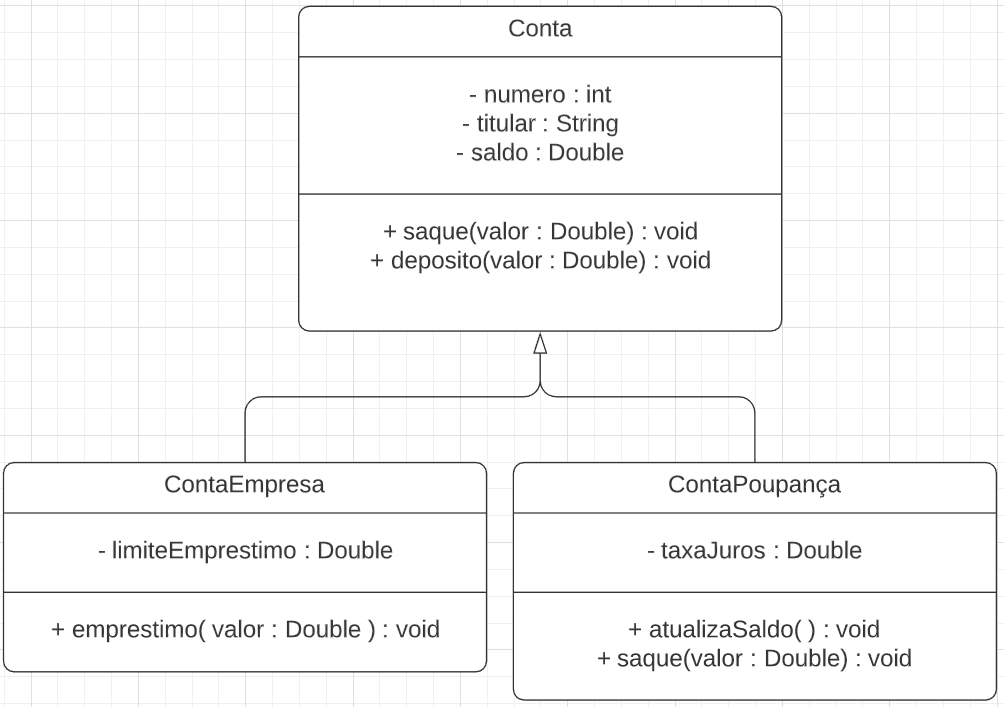
É a implementação de um método de uma superclasse na subclasse.

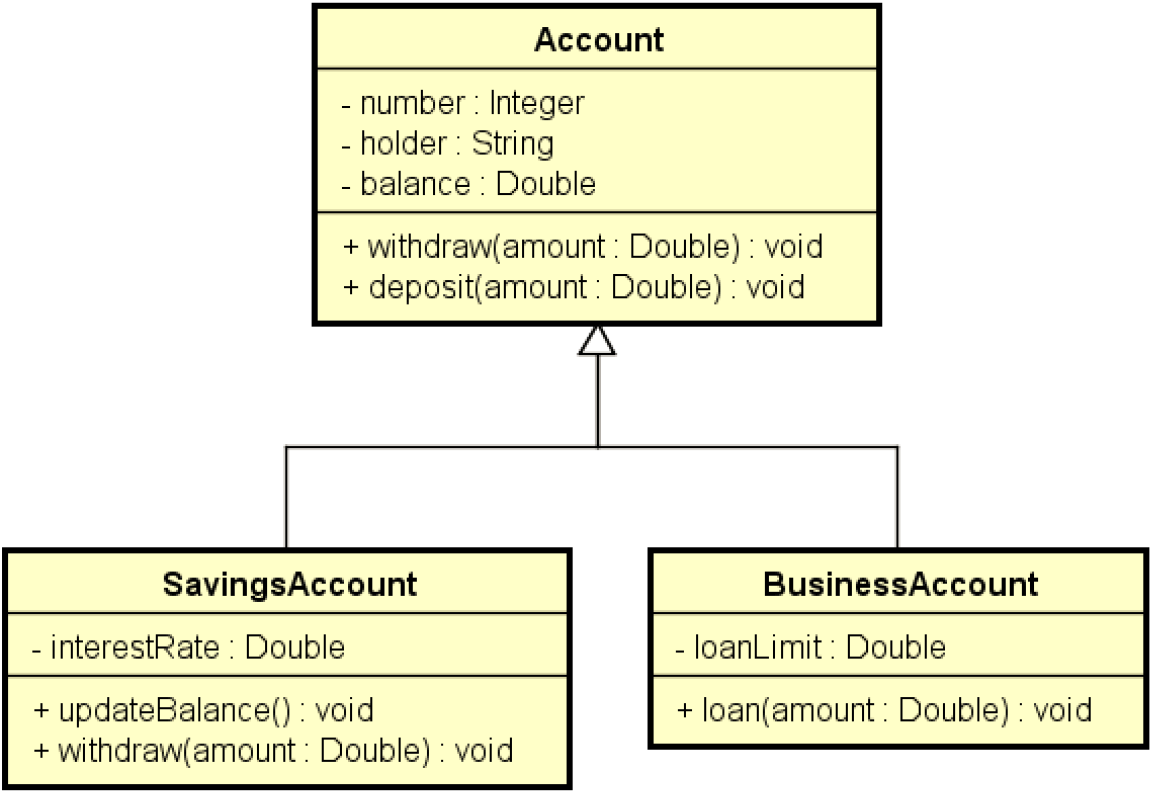
* Nesse exemplo, a superclasse é a conta e as subclasses são as ContaEmpresa e ContaPoupanca

Para que um método comum (não abstrato) possa ser sobreposto (sobrescrito), deve ser incluído nele o prefixo "**virtual**"

Ao sobrescrever um método, devemos incluir nele o prefixo "**override**"

* Até então, temos utilizado o override para implementar o método ToString.
  + O método ToString pertence à classe Object. E a classe Object é uma superclasse de todas as outras classes. Quando nós alteramos o ToString com o override, estamos sobrescrevendo um método da superclasse Object.





## Exemplo:

Suponha as seguintes regras para saque:

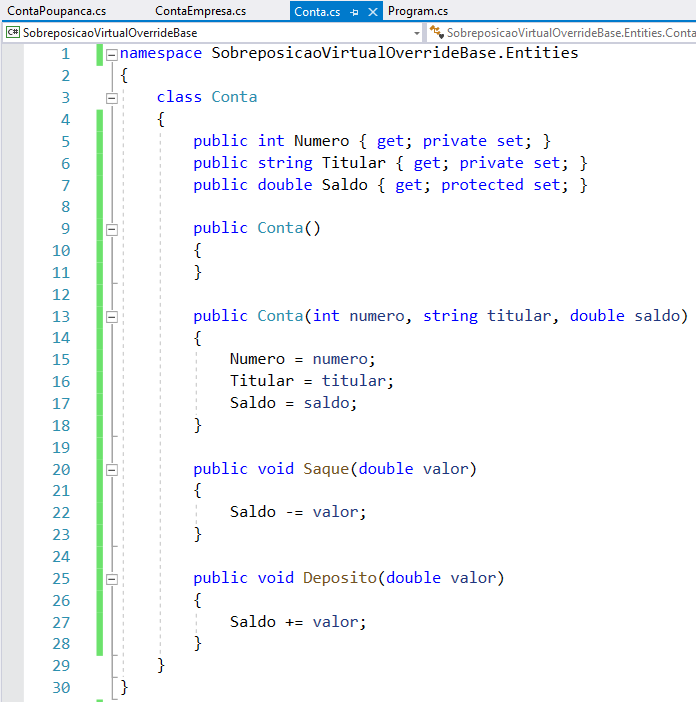
Conta comum: é cobrada uma taxa no valor de 5.00.

Conta poupança: não é cobrada taxa.

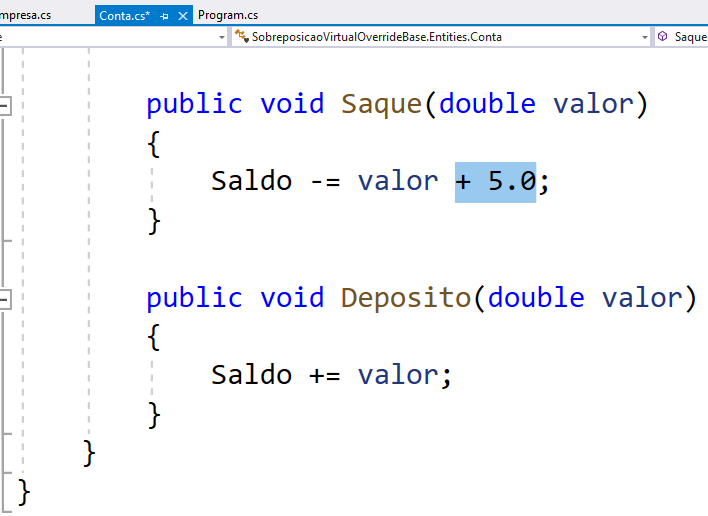
Como resolver isso?

Resposta: sobrescrevendo o método **saque** na subclasse **ContaPoupança.**

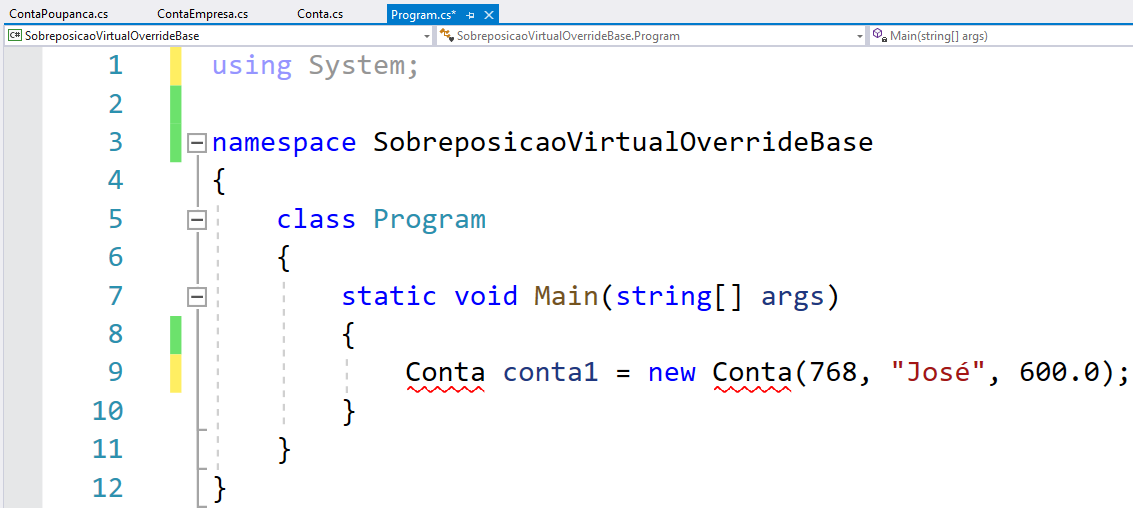
A classe Conta estava assim:



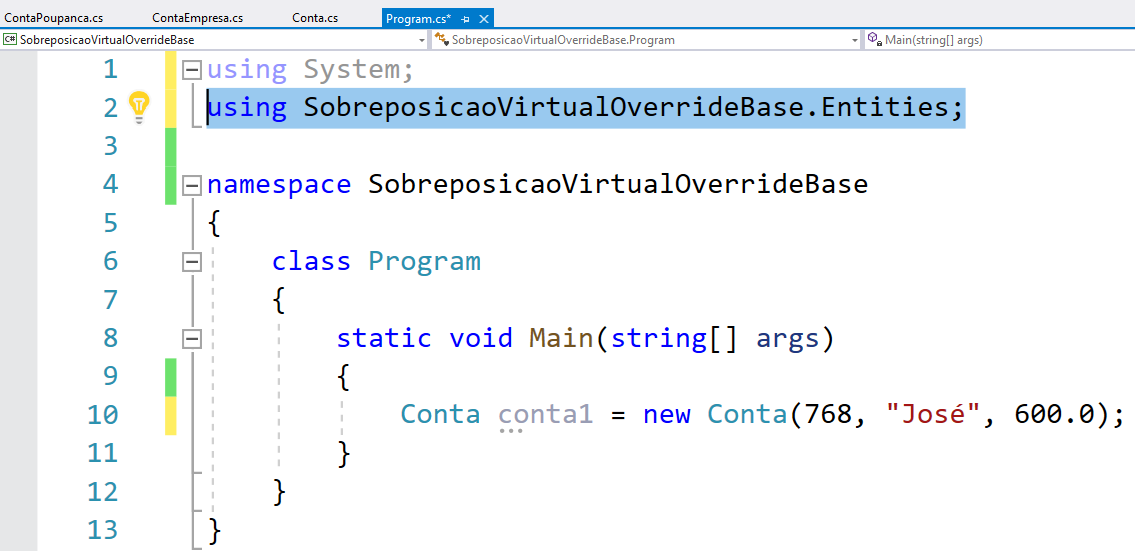
Vamos adicionar uma taxa de 5,00 ao saque.



Agora vamos fazer um teste. No programa principal, vamos apagar o código anterior e criar uma variável conta1 e iniciar com alguns valores.

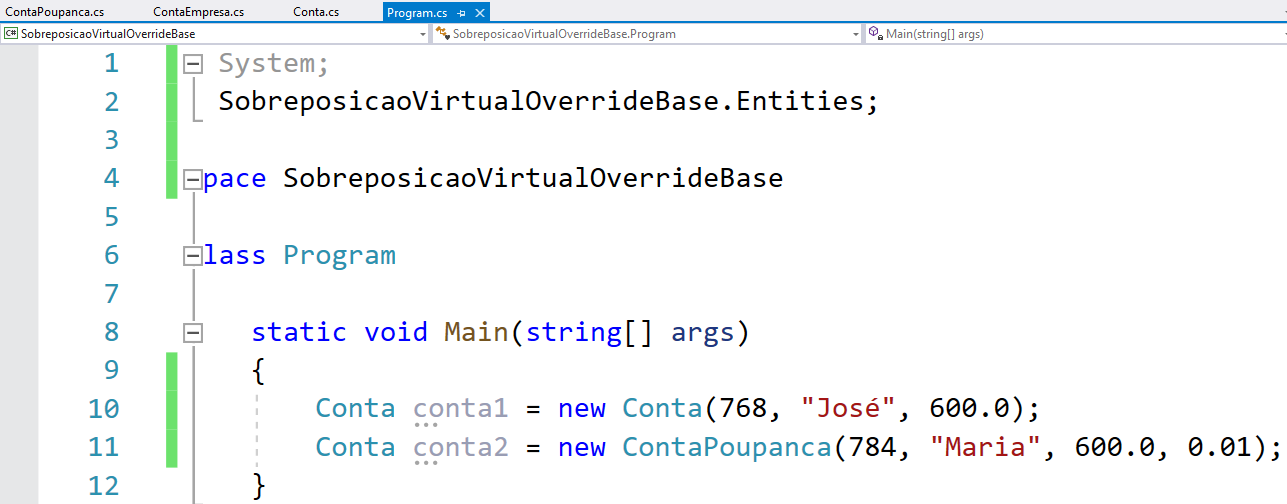


Observem que eu não consigo utilizar as contas se não referencia-las com o using.

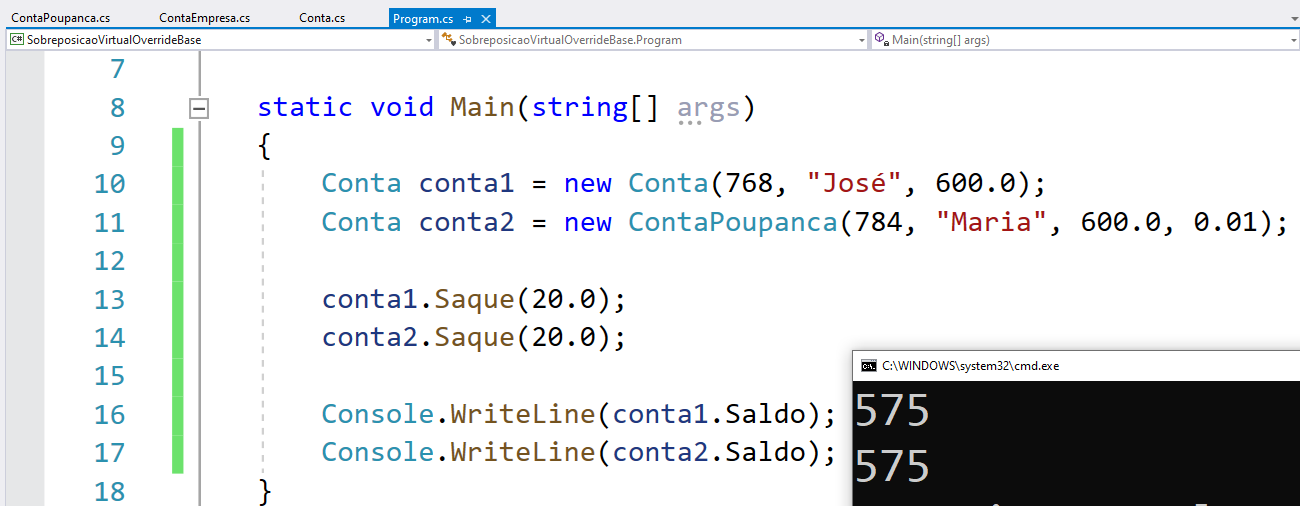


Agora sim.

Vamos fazer um UpCasting para instanciar uma ContaPoupanca com uma Conta normal e atribuir alguns valores também.



Agora vamos fazer um saque nas duas contas e mostrar o novo saldo.

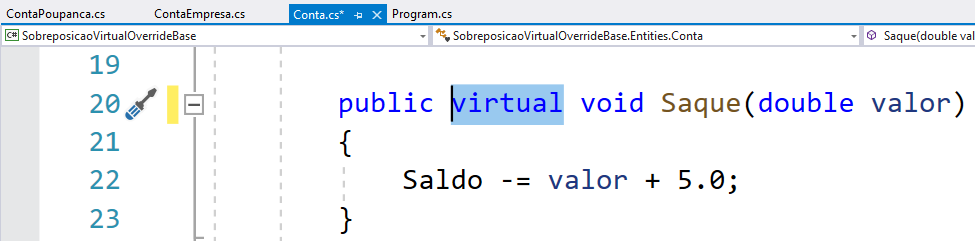


Observe que o saque de 20.0 foi feito com a taxas de 5.0 nas duas contas. Mesmo implementando o ajuste apenas na Conta superclasse.

# virtual

## Regra de negócio

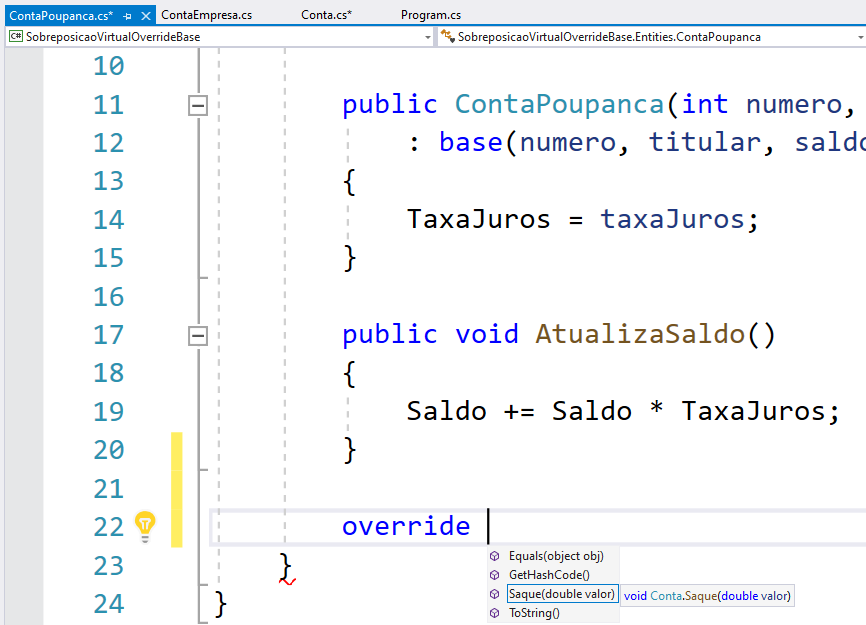
Agora precisamos implementar a regra de negócio, onde a cobrança da taxa de 5.0 só será cobrada na Conta e não na ContaPoupanca. Para fazer isso, insira no método Saque a palavra **virtual**.



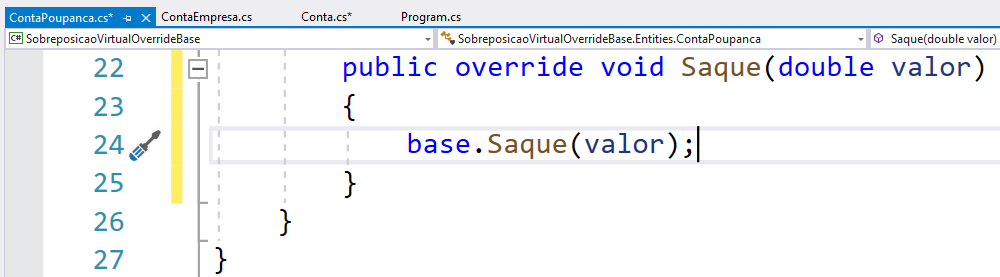
E na classe ContaPoupanca vamos sobrescrever o método Saque da superclasse com uma implementação própria.

# override

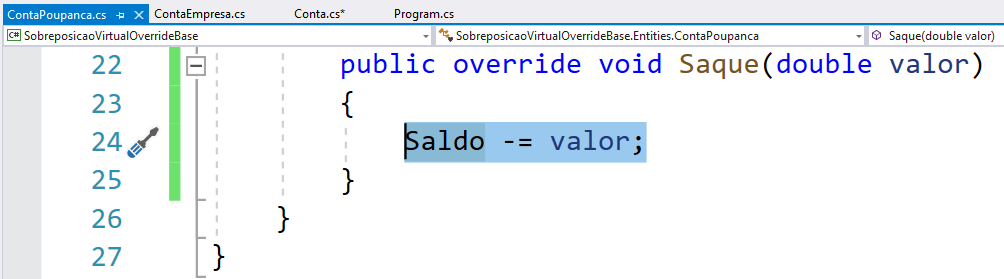
Na classe ContaPoupanca, digite override e pressione a tecla espaço. Algumas opções serão oferecidas pelo Visual Studio.



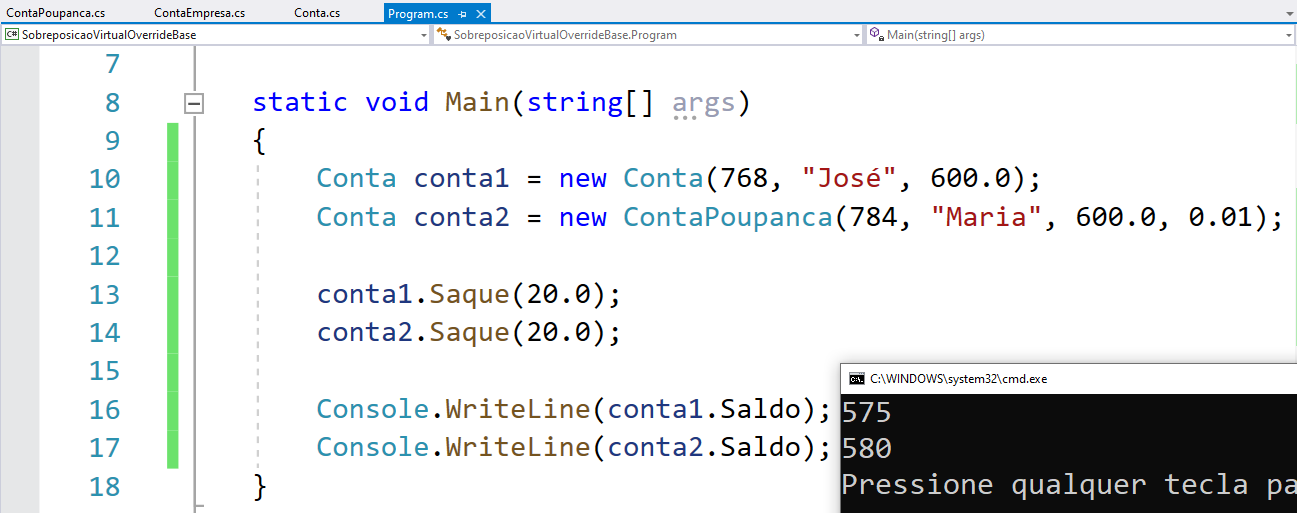
Todos os métodos marcados como virtual, serão oferecidas para serem utilizadas aqui. Clique na opção Saque.



Vamos implementar o método, sem a cobrança da taxa.



Testando o resultado do saque com Ctrl + F5.



Sobreposição então, é quando criamos um método em uma classe que será sobreposta um outro método em uma outra classe.

# base

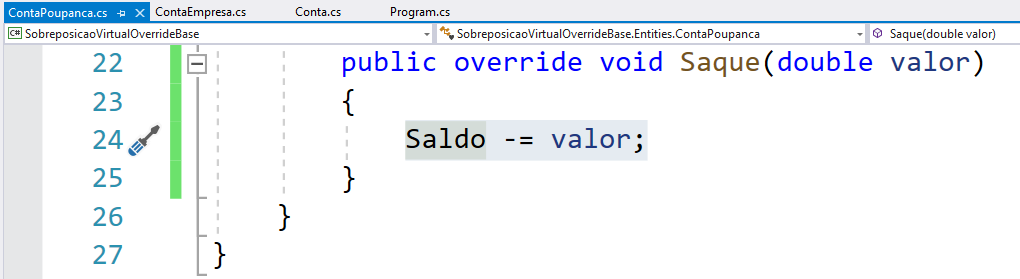
## Palavra base

É possível chamar a implementação da superclasse usando a palavra base.

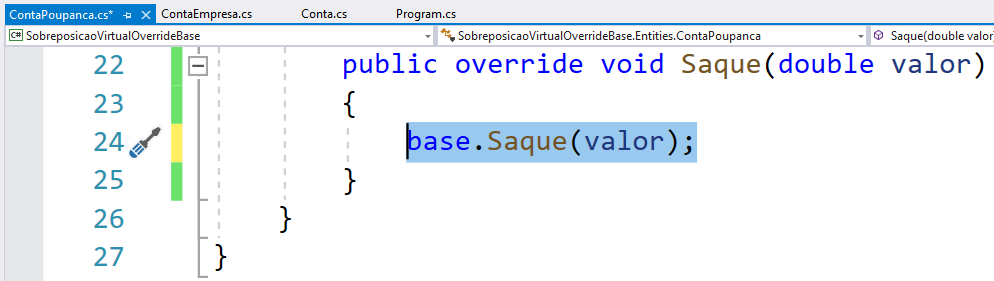
Exemplo: suponha que a regra para saque para ContaPoupanca seja realizar o saque normalmente da superclasse (Conta), e depois descontar mais 2.0. Vamos aproveitar a implementação da superclasse e ainda criar outra regra.

Exemplo:

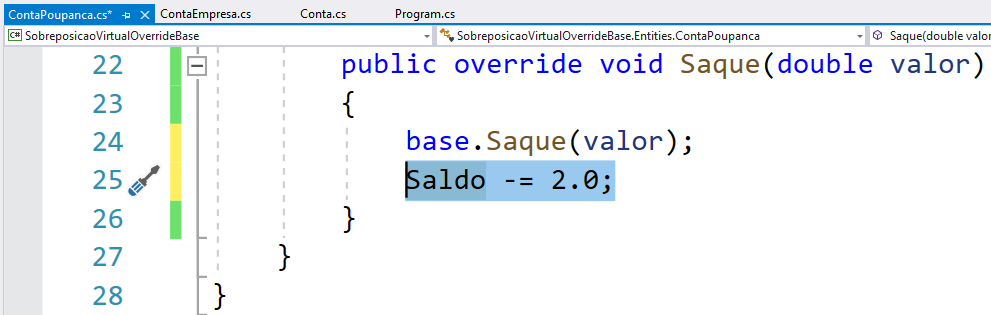
Apague o saque da ContaPoupanca.



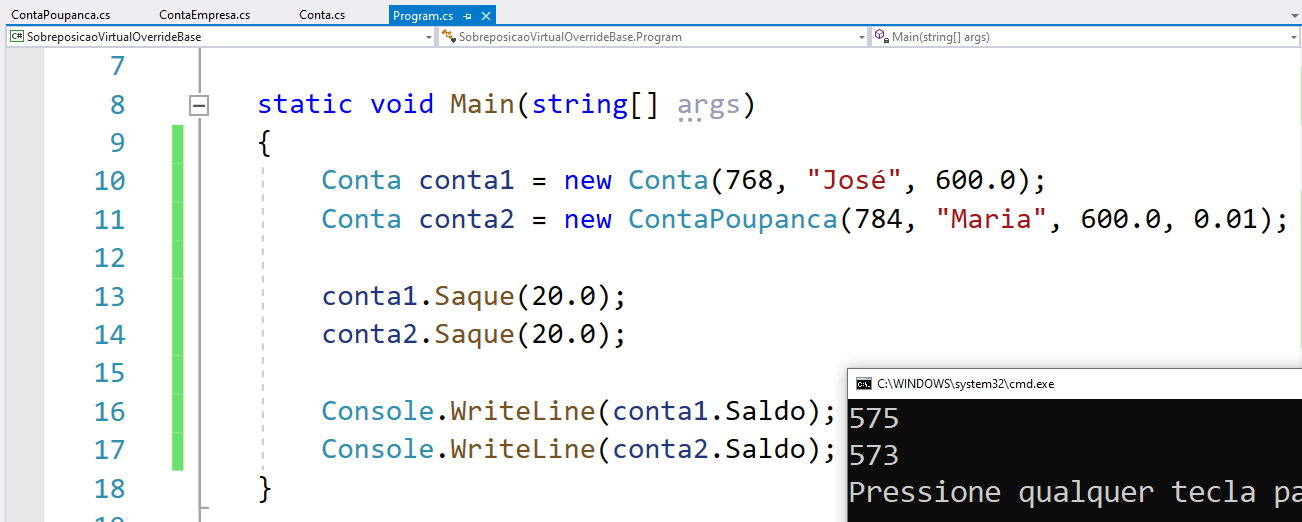
Com a palavra base vamos chamar a implementação da superclasse Conta.



Agora, vamos aplicar a regra da taxa de 2.0.

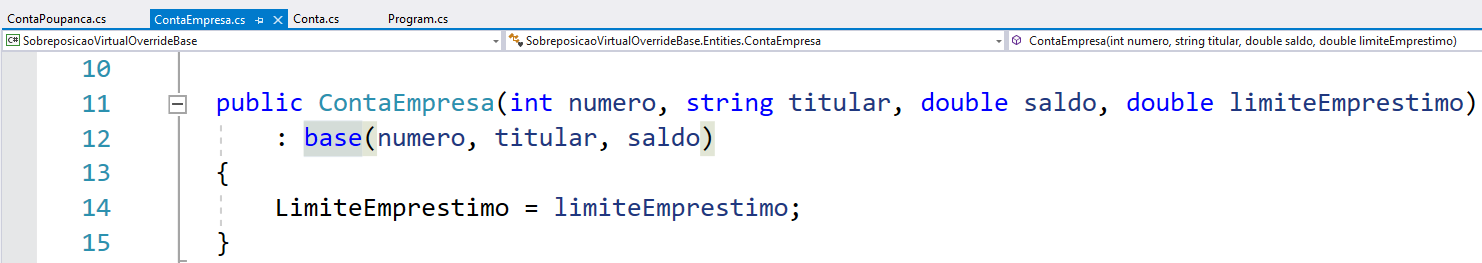


Testando.



Ok, foi sacado os 20.0 de cada conta. Na Conta comum, foi retirado a taxa de 5.0, na ContaPoupanca foi retirado além dos 5.0 da superclasse, os 2.0 da subclasse.

Recordando: A palavra base já foi utilizada para reaproveitar os construtores de uma superclasse em uma subclasse.



# GitHub

O código está na subpasta:

**\Aula\_08\_Classe\** **SobreposicaoVirtualOverrideBase**

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

# Referência:

Docs Microsoft – Virtual (Referência de C#) – < https://docs.microsoft.com/pt-br/dotnet/csharp/language-reference/keywords/virtual > Acessado dia 14 de dezembro de 2020.

Tutorial W3Schools – Polimorfismo - < https://www.w3schools.com/cs/cs\_polymorphism.asp > Acessado dia 14 de dezembro de 2020.