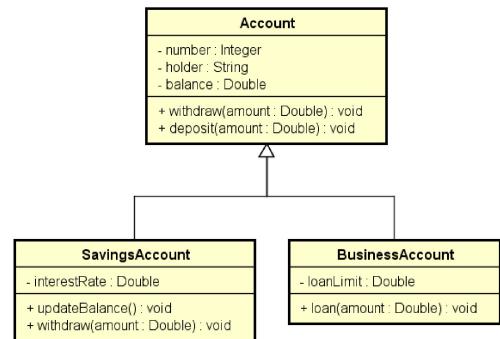
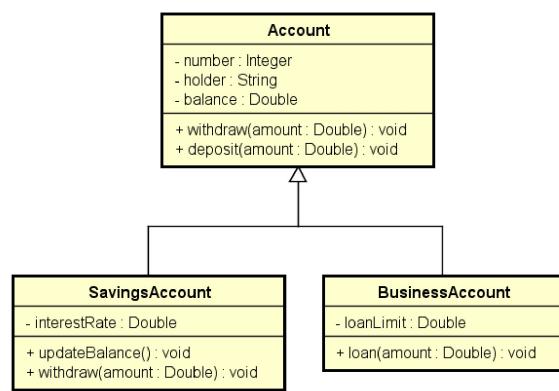


## Sobreposição ou sobrescrita

- É a implementação de um método de uma superclasse na subclasse
- Para que um método comum (não abstrato) possa ser sobreposto, deve ser incluído nele o prefixo "**virtual**"
- Ao sobrescrever um método, devemos incluir nele o prefixo "**override**"



## 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 withdraw na subclasse SavingsAccount

**Account:**

```
public virtual void Withdraw(double amount) {  
    Balance -= amount + 5.0;  
}
```

**SavingsAccount:**

```
public override void Withdraw(double amount) {  
    Balance -= amount;  
}
```

## Palavra base

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

Exemplo: suponha que a regra para saque para conta poupança seja realizar o saque normalmente da superclasse (Account), e depois descontar mais 2.0.

```
public override void Withdraw(double amount) {  
    base.Withdraw(amount);  
    Balance -= 2.0;  
}
```

## Recordando: usando base em construtores

```
class BusinessAccount : Account
{
    public double LoanLimit { get; set; }

    public BusinessAccount()
    {
    }

    public BusinessAccount(int number, string holder, double balance, double loanLimit)
        : base(number, holder, balance)
    {
        LoanLimit = loanLimit;
    }

    (...)
```

## Código fonte desta aula

<https://github.com/acenelio/inheritance3-csharp>

# Classes e métodos selados

<http://educandoweb.com.br>

Prof. Dr. Nelio Alves

## Classes e métodos selados

- Palavra chave: sealed
- **Classe:** evita que a classe seja herdada
  - Nota: ainda é possível extender a funcionalidade de uma classe selada usando "extension methods"

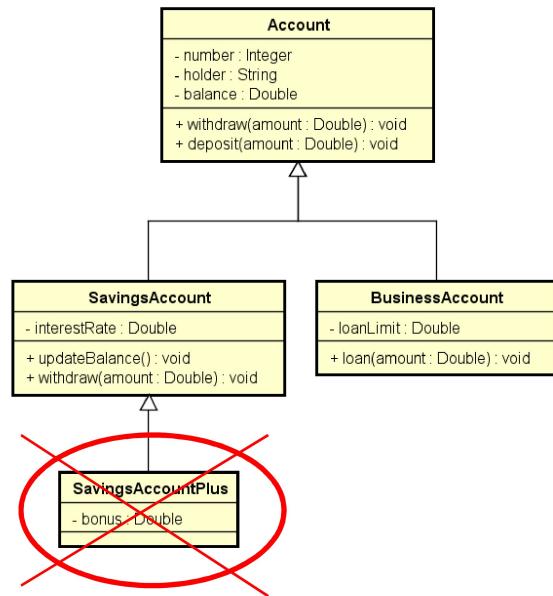
```
namespace Course {  
    sealed class SavingsAccount {
```

- **Método:** evita que um método sobreposto possa ser sobreposto novamente
  - Só pode ser aplicado a métodos sobrepostos

## Exemplo - Classe selada

Suponha que você queira evitar que sejam criadas subclasses de SavingsAccount

```
namespace Course {  
    sealed class SavingsAccount {  
        (...)
```



## Exemplo - método selado

Suponha que você não queira que o método Withdraw de SavingsAccount seja sobreposto novamente

```
public sealed override void Withdraw(double amount)  
{  
    base.Withdraw(amount);  
    Balance -= 2.0;  
}
```

## Pra quê?

- 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.
  - Exemplo clássico: string

## Introdução ao polimorfismo

<http://educandoweb.com.br>

Prof. Dr. Nelio Alves