

```
Account acc = new Account(1001, "Alex", 0.0);
BusinessAccount bacc = new BusinessAccount(1002, "Maria", 0.0, 500.0);

// UPCASTING

Account acc1 = bacc;
Account acc2 = new BusinessAccount(1003, "Bob", 0.0, 200.0);
Account acc3 = new SavingsAccount(1004, "Anna", 0.0, 0.01);

// DOWNCASTING

BusinessAccount acc4 = (BusinessAccount)acc2;
acc4.loan(100.0);

// BusinessAccount acc5 = (BusinessAccount)acc3;
if (acc3 instanceof BusinessAccount) {
    BusinessAccount acc5 = (BusinessAccount)acc3;
    acc5.loan(200.0);
    System.out.println("Loan!");
}

if (acc3 instanceof SavingsAccount) {
    SavingsAccount acc5 = (SavingsAccount)acc3;
    acc5.updateBalance();
    System.out.println("Update!");
}
```

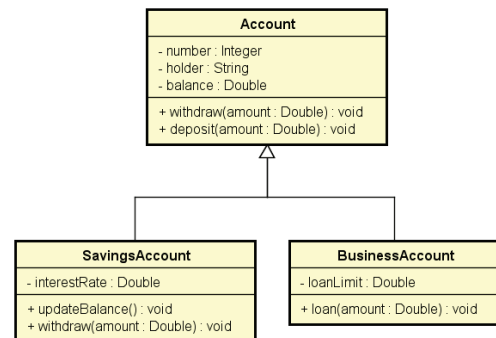
Sobreposição, palavra super,
anotação @Override

<http://educandoweb.com.br>

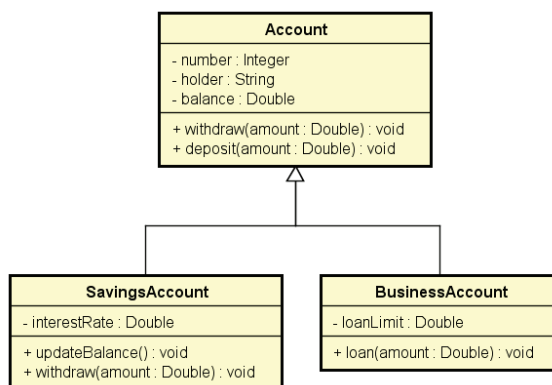
Prof. Dr. Nelio Alves

Sobreposição ou sobrescrita

- É a implementação de um método de uma superclasse na subclasse
- É fortemente recomendável usar a anotação `@Override` em um método sobrescrito
 - Facilita a leitura e compreensão do código
 - Avisamos ao compilador (boa prática)



Exemplo



Suponha que a operação de saque possui uma taxa no valor de 5.0. Entretanto, se a conta for do tipo poupança, esta taxa não deve ser cobrada.

Como resolver isso?

Resposta: sobrescrevendo o método `withdraw` na subclasse **SavingsAccount**

Account:

```
public void withdraw(double amount) {  
    balance -= amount + 5.0;  
}
```

SavingsAccount:

```
@Override  
public void withdraw(double amount) {  
    balance -= amount;  
}
```

Palavra super

É possível chamar a implementação da superclasse usando a palavra **super**.

Exemplo: suponha que, na classe BusinessAccount, a regra para saque seja realizar o saque normalmente da superclasse, e descontar mais 2.0.

```
@Override  
public void withdraw(double amount) {  
    super.withdraw(amount);  
    balance -= 2.0;  
}
```

Recordando: usando `super` em construtores

```
public class Account {  
    private Integer number;  
    private String holder;  
    private Double balance;  
  
    public Account(Integer number, String holder, Double balance) {  
        this.number = number;  
        this.holder = holder;  
        this.balance = balance;  
    }  
    (...)
```

```
public class BusinessAccount extends Account {  
    private double loanLimit;  
  
    public BusinessAccount(Integer number, String holder, Double balance, double loanLimit) {  
        super(number, holder, balance);  
        this.loanLimit = loanLimit;  
    }  
    (...)
```

Código fonte desta aula

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