

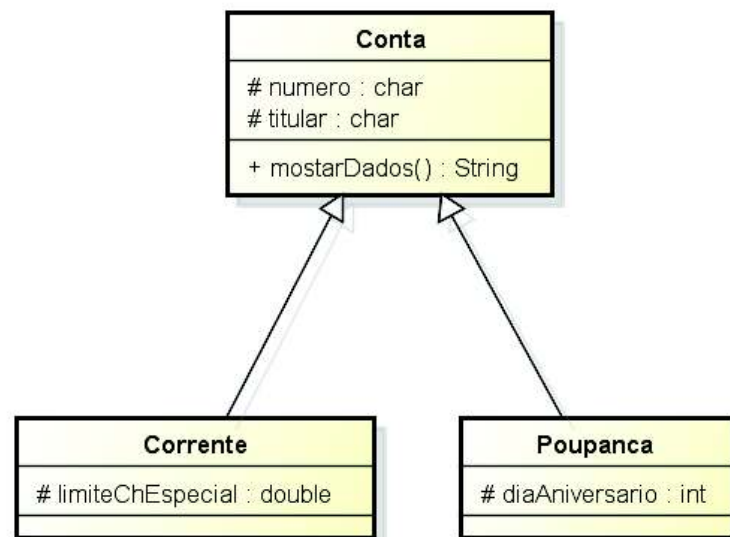
# Programação Orientada a Objetos

## HERANÇA

- A herança é um relacionamento pelo qual uma classe, chamada de **sub-classe**, herda todos comportamentos de outra classe, chamada de **super-classe**.

É permitido que a sub-classe possua **novos** métodos e atributos.

É também possível que a sub-classe **sobrescreva** métodos da super-classe (reescrevendo-os diretamente).

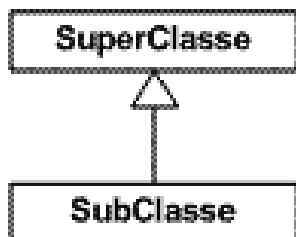
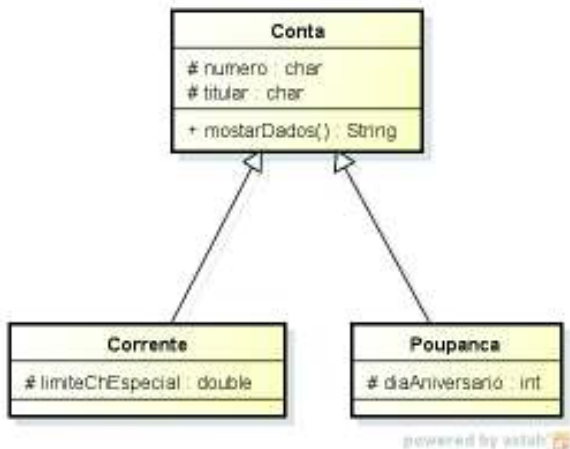


powered by astah

- O objetivo principal da herança é a **reutilização de software**, já que novas classes são criadas a partir de outras já existentes, herdando seus atributos e métodos.

# Programação Orientada a Objetos

## HERANÇA



Os atributos da superclasse  
devem ser protegidos.

```
public class Conta {

    protected String numero;
    protected String titular;

    public void setNumero(String n){
        numero = n;
    }

    public String getNumero(){
        return numero;
    }

    public void setTitular(String t){
        titular = t;
    }

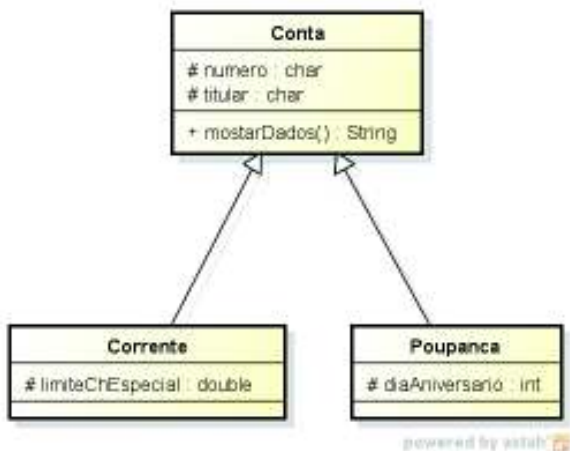
    public String getTitular(){
        return titular;
    }

    public String mostrarDados(){
        return numero + " - " + titular;
    }

}
```

# Programação Orientada a Objetos

## HERANÇA



```
public class Corrente extends Conta{

    protected double limiteChEspecial;

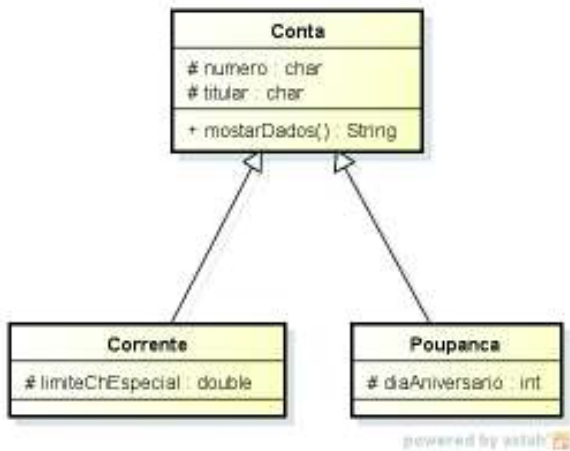
    public void setLimite(double l){
        limiteChEspecial = l;
    }

    public double getLimite(){
        return limiteChEspecial;
    }

}
```

# Programação Orientada a Objetos

## HERANÇA



```
public class Poupanca extends Conta {

    protected int diaAniversario;

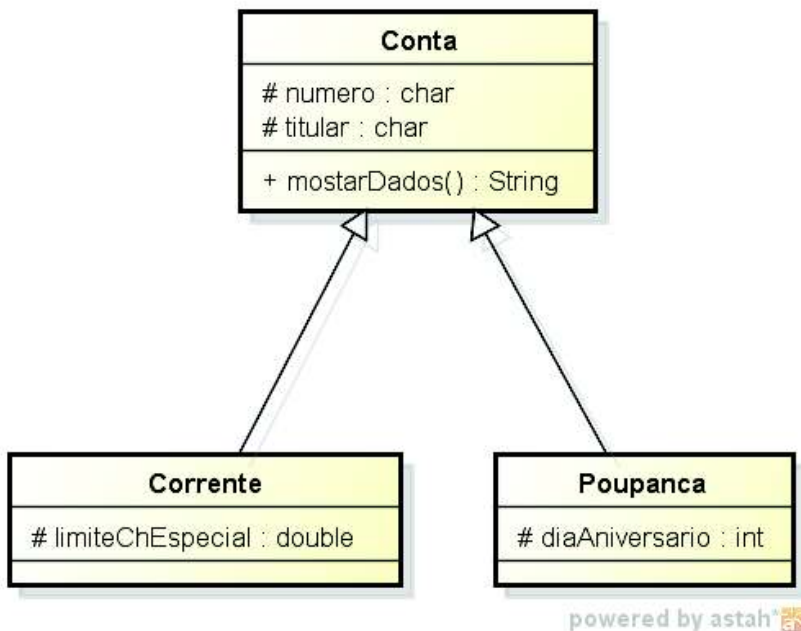
    public void setDia(int d){
        diaAniversario = d;
    }

    public int getDia(){
        return diaAniversario;
    }

}
```

# Programação Orientada a Objetos

HERANÇA



Implemente esse modelo.

Como mostrar os atributos específicos de **Corrente** e **Poupança**?

O método `mostrarDados` não deveria ser implementado em **Conta**!!

# Programação Orientada a Objetos

## **EXERCÍCIO 4.3 - PROJETO “Ex04” - PACOTE “Empresa”**

**Crie uma classe chamada Ingresso que possui um valor em reais e um método imprimeValor().**

**a. crie uma classe VIP, que herda Ingresso e possui um valor adicional. Crie um método que retorne o valor do ingresso VIP (com o adicional incluído).**

**b. crie uma classe Normal, que herda Ingresso e possui um método que imprime:"Ingresso Normal".**

**c. crie uma classe CamaroteInferior (que possui a localização do ingresso e métodos para acessar e imprimir esta localização) e uma classe CamaroteSuperior, que é mais cara (possui valor adicional). Esta última possui um método para retornar o valor do ingresso. Ambas as classes herdam a classe VIP.**

**d. Crie uma classe de Teste com o método main.**

# Programação Orientada a Objetos

## **EXERCÍCIO 4.4 - PROJETO “Ex04” - PACOTE “Empresa”**

**Crie a classe Imovel, que possui um endereço e um preço.**

**a. crie uma classe Novo, que herda Imovel e possui um adicional no preço. Crie métodos de acesso e impressão deste valor adicional.**

**b. crie uma classe Velho, que herda Imovel e possui um desconto no preço. Crie métodos de acesso e impressão para este desconto.**

**c. Crie uma classe de Teste com o método main.**