Programação Orientada a Objetos

Prof. Dr. Alan Souza

alan.souza@unama.br

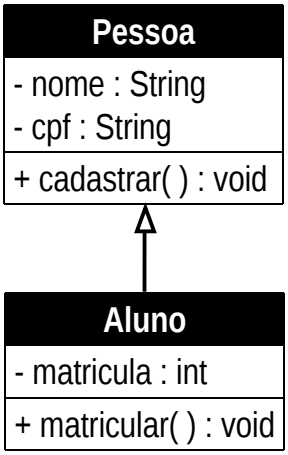
2020

Herança

- Relacionamento entre duas classes;
- Para saber se o relacionamento de herança faz sentido, pergunta-se "É-UM":
  - Exs: ***Todo*** Aluno ***é uma*** Pessoa, ***Toda*** Casa ***é um*** Imóvel.
- Quando a classe A herda da classe B, então os atributos e os métodos acessíveis da classe B poderão ser usados na classe A sem a necessidade de um objeto para o acesso;
- A sub-classse (filha) pode usar o objeto **super** para referenciar a super classe (mãe).
- Em Java, utiliza-se a palavra reservada **extends** para programar o relacionamento de herança entre duas classes.

Herança

- Diagrama de classes (UML - *Unified Model Language*)

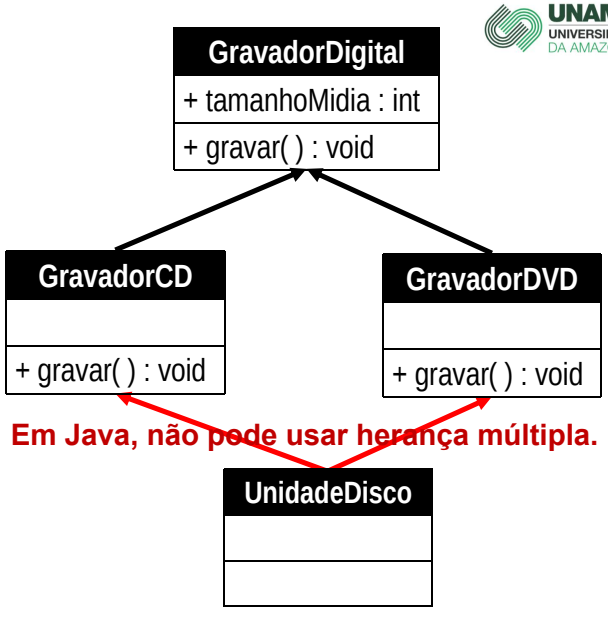


Modificadores de acesso (símbolos)	
-	private
+	public
#	protected
~	default

Herança

- Qual o problema da herança múltipla (herdar de mais de uma classe ao mesmo tempo)?
- R: Losango Mortal!

*O que aconteceria se fosse criado um objeto da classe UnidadeDisco e o método **gravar** fosse chamado?*
R: um impasse/embaraço.



Herança - Polimorfismo

- Sobrescrita & Sobrecarga.
- **Sobrescrita** (*Override*): quando um método da classe filha muda o comportamento da classe mãe, ou seja, sobrescreve-o;
- **Sobrecarga** (*Overload*): quando vários métodos com o mesmo nome ou vários construtores diferentes possuem número de parâmetros de entrada diferente.

Herança - Polimorfismo - Ex. Sobrescrita - ProjetoBanco

```
public class Transacao {
    //atributos e métodos...
    public void emitirComprovante( ){
        Sout("Desc.: " + this.getDescricao( ));
        Sout("Data: " + this.getData( ));
        Sout("Valor: " + this.getValor( ));
    }
}
public class Transferencia extends Transacao {
    public void emitirComprovante( ) {
      super.emitirComprovante( );
      Sout("Conta origem: " + this.getContaOrigem( ));
      Sout("Conta destino: " + this.getContaDestino( ));
    }
}
```

```
public class PagamentoBoleto extends Transacao {
    public void emitirComprovante( ) {
      super.emitirComprovante( );
      Sout("Linha dig.: " + this.getLinhaDigitavel( ));
      Sout("Data venc: " + this.getDataVencimento( ));
      Sout("Cedente: " + this.getCedente( ));
    }
}
```

← **Sobrescrita**

Herança - Polimorfismo - Ex. Sobrecarga - ProjetoBanco

```
public class Transferencia extends Transacao {
    private String contaOrigem;
    private String contaDestino;
    public Transferencia( ) {
    }
    public Transferencia(String contaOrigem, String contaDestino) {
        this.contaOrigem = contaOrigem;
        this.contaDestino = contaDestino;
    }
}
```

Socrecarga

Herança

Exercício do tipo TODO

- Acesse **https://github.com/amarcel/unama2020p1**
- Baixe os projetos-exercícios **Exerc07_Janelas, Exerc08_Janelas, Exerc09_Janelas,** da pasta **POO - EXERCICIOS - TODO**
- Abra os projetos no Netbeans;
- Faça os TODOs de cada projeto, começando pelo **Exerc07_Janelas**;