

# Classes Abstratas

Prof. Pedro Pongelupe



PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS  
Departamento de Ciência da Computação

# Sumário

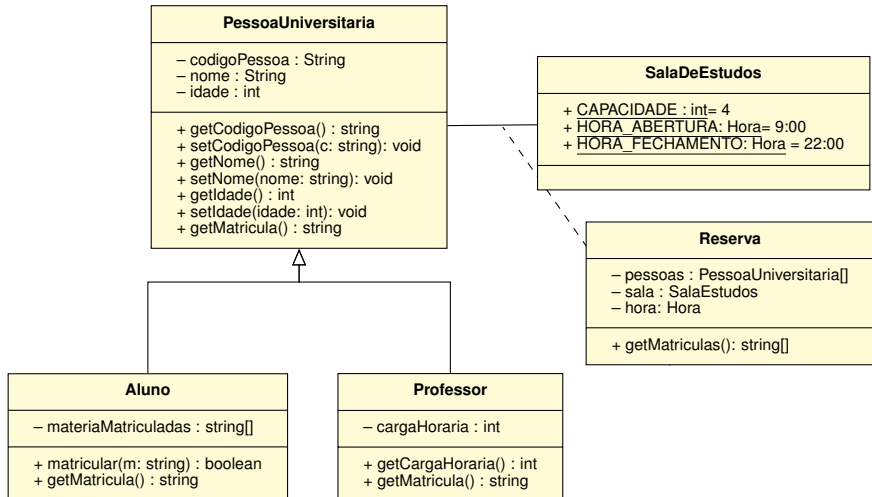
## 1 Polimorfismo

## 2 Classe Abstrata

- Pensando a classe FuncionarioAdministrativo
- Classe PessoaUniversitaria abstrata



# Exemplo: Reserva Sala de Estudos polimórfica



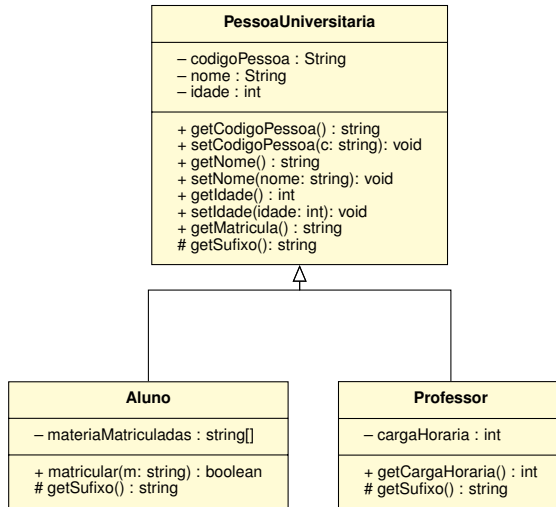


# Novas regras para geração da matrícula

Para todas as especialização, as existentes e novas, deverão **obrigatoriamente** adicionar um prefixo ao código pessoa para gerar a matrícula.

- Aluno: AL{código pessoa}
- Professor: PF{código pessoa}

# Utilizando Polimorfismo para resolver o novo requisito





# Implementação

```
public class PessoaUniversitaria {  
    ...  
    public String getMatricula() {  
        return getSufixo()  
            .concat(codigoPessoa);  
    }  
  
    public String getSufixo() {  
        return "";  
    }  
}
```

```
public class Aluno {  
    ...  
    @Override  
    protected String getSufixo() {  
        return "AL";  
    }  
    ...  
}  
  
public class Professor {  
    ...  
    @Override  
    protected String getSufixo() {  
        return "PR";  
    }  
    ...  
}
```



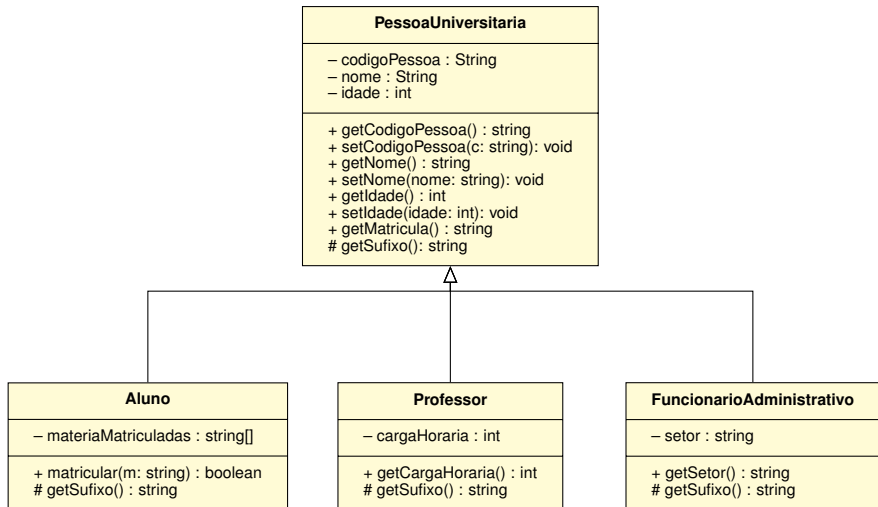
# Conclusões relativas ao uso da herança

- Polimorfismo
  - É a característica que permite que diferentes objetos respondam a mesma mensagem cada um a sua maneira.
  - Uma referência para a superclasse só pode acessar os métodos previstos na interface da superclasse, porém, o Java automaticamente ativa a implementação correspondente no objeto apontado.
- O comando **instanceof** verifica se um determinado objeto é de uma instância de uma certa classe

```
if (pessoa instanceof Aluno)
    System.out.println("Aluno");
```



# Pensando a classe FuncionarioAdministrativo





## Exemplo no Eclipse



# Problemas com a Implementação

- Ainda faz sentido existir instâncias de PessoaUniversitaria?
- As novas especializações de PessoaUniversitaria não são obrigadas a implementar o método *getSufixo()* : *string*

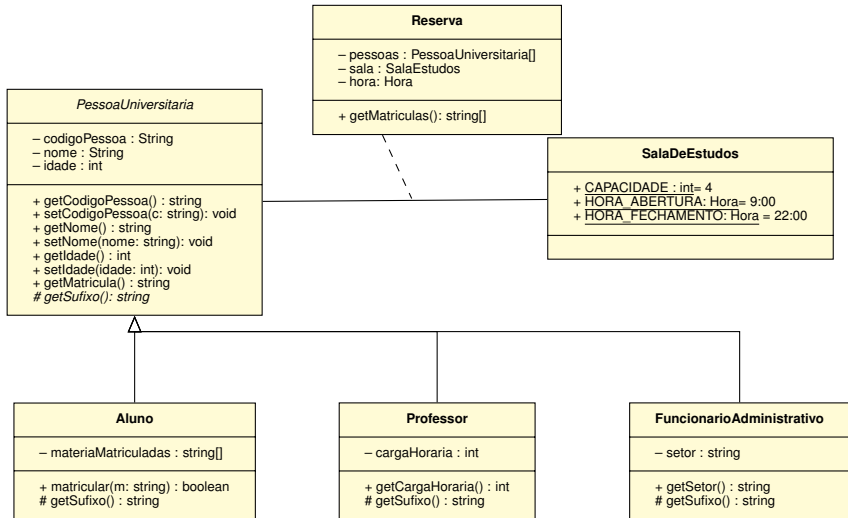


# Classes abstratas

- Classes abstratas permitem que se definam métodos sem implementação que devem ser redefinidos em classes derivadas.
- Classes abstratas podem ou não ter métodos abstratos.
- Classes abstratas não podem ser instanciadas.
- As classes derivadas de classes abstratas herdam todos os métodos, incluindo os abstratos.
- As classes derivadas de classes abstratas são abstratas até que implementem os métodos abstratos.
- Em Java: palavra-chave **abstract**.



# Exemplo: Reserva Sala de Estudos polimórfica





# Classe PessoaUniversitaria abstrata

```
public abstract class PessoaUniversitaria {  
    private String codigoPessoa;  
    private String nome;  
    private int idade;  
  
    public PessoaUniversitaria(String codigoPessoa, String nome,  
int idade) {  
        this.codigoPessoa = codigoPessoa;  
        this.nome = nome;  
        this.idade = idade;  
    }  
    ...  
  
    public String getMatricula() {  
        return getSufixo().concat(codigoPessoa);  
    }  
  
    protected abstract String getSufixo();  
}
```



# Obrigado!!

Muito obrigado pela atenção! Alguma dúvida? Bora praticar!!!

*"Quem não se movimenta, não sente as correntes que o prendem."*