

# DA1MCTA018-13SA - Programação Orientada a Objetos - Paulo Henrique Pisani - 2021.2

[Painel](#) / [Meus cursos](#) / [POO - DA1MCTA018-13SA - 2021.2](#) / [Herança, polimorfismo](#) / [\[EP\] Vínculo com a Universidade](#)

Descrição

[Visualizar envios](#)

## [EP] Vínculo com a Universidade

**Data de entrega:** segunda, 28 Jun 2021, 23:59

**Arquivos requeridos:** Pessoa.java, Professor.java, Aluno.java ([Baixar](#))

**Tipo de trabalho:** Trabalho individual

## Vínculo com a Universidade

**Autor:** Guilherme Aldeia. POO - QS/2020

### Tarefa

Nesse exercício, você criará três classes pertencentes ao **pacote universidade**: **Pessoa**, **Professor** e **Aluno**.

Classe **Pessoa**:

- Construtor sem parâmetros: `Pessoa()`;
- Métodos:
  - `void setNomeSobrenome(String nome, String sobrenome)`: recebe duas strings contendo nome e sobrenome, e guarda os valores em atributos na instância da classe.
  - `String getEmail()`: retorna uma string com email no formato "<nome>.<sobrenome>@dominio.generico.com.br".
  - `String getVinculo()`: retorna uma string com o vínculo da pessoa com a UFABC (nesta classe `Pessoa`, retorna "Sem vinculo com a UFABC")

Subclasses **Professor** e **Aluno**: estendem a classe `Pessoa` (herdam a classe `Pessoa`). Ambas as classes devem possuir o construtor sem parâmetros. Essas subclasses devem sobrepor os métodos `getEmail()` e `getVinculo()` da classe `Pessoa`. O email de um professor tem o formato "<nome>.<sobrenome>@ufabc.edu.br", enquanto que o de um aluno tem o formato "<nome>.<sobrenome>@aluno.ufabc.edu.br". O método `getVinculo()` deve retornar, respectivamente, "Professor da UFABC" e "Aluno da UFABC" nas classes `Professor` e `Aluno`.

**Submeta apenas as classes `Pessoa`, `Professor` e `Aluno`. As classes devem ser públicas e estar no pacote `universidade` (incluir `package universidade`).**

**Importante:** O programa principal já existe no sistema de correção automática. As classes submetidas não pode realizar impressão de dados ou utilizar `import`. Todas as classes devem possuir o construtor sem parâmetros, que será usado pelo sistema de correção automática para instanciar as classes.

### Casos de teste

Formato dos casos de teste (esse é o formato do que aparece ao avaliar o programa no sistema de correção automática; o código submetido não deve realizar leitura de dados ou impressão, isso já é realizado pelo sistema de correção).

As classes serão instanciadas da seguinte forma pelo sistema de correção (observe que o tipo da variável é sempre `Pessoa`, isso é possível, pois as classes `Aluno` e `Professor` são subclasses de `Pessoa`):

```
Pessoa pe = new Pessoa();
pe.setNomeSobrenome(nomePessoa, sobrenomePessoa);
Pessoa al = new Aluno();
al.setNomeSobrenome(nomeAluno, sobrenomeAluno);
Pessoa pr = new Professor();
pr.setNomeSobrenome(nomeProfessor, sobrenomeProfessor);
```

Entrada:

NomePessoa SobrenomePessoa

NomeAluno SobrenomeAluno

NomeProfessor SobrenomeProfessor

Saída:

Saída dos métodos getEmail() e getVinculo() para uma instância de Pessoa

Saída dos métodos getEmail() e getVinculo() para uma instância de Aluno

Saída dos métodos getEmail() e getVinculo() para uma instância de Professor

◀ [Vídeo] Ocultar atributos

Seguir para...

[EP] Veículos ▶

VPL



Este é o Ambiente Virtual de Aprendizagem da UFABC para apoio ao ensino presencial e semipresencial. Esta plataforma permite que os usuários (educadores/alunos) possam criar cursos, gerenciá-los e participar de maneira colaborativa.

Informação

- Conheça a UFABC
- Conheça o NTI
- Conheça o Netel

Contato

Av. dos Estados, 5001. Bairro Bangu - Santo André /SP – Brasil. CEP 09210-580.

Siga-nos

