

3. Herança e Polimorfismo

Nesta etapa, foi aplicada a herança utilizando a classe abstrata **Pessoa** como superclasse das classes **Visitante**, **Aluno** e **Professor**. Cada classe derivada herda os atributos e métodos definidos na superclasse, podendo adicionar funcionalidades específicas.

```
1 public class Aluno extends Pessoa{
2     String matricula;
3
4     Aluno(String nome, String cpf, int idade, String matricula){
5         super(nome, cpf, idade);
6         this.matricula = matricula;
7     }
8
9     public void pagarMensalidade(){
10        System.out.println(this.nome + " pagou a mensalidade!");
11    }
12
13    @Override
14    public String toString(){
15        return this.nome;
16    }
17 }
```

Na classe **Aluno**, foi incluído o atributo *matrícula* e o método **pagarMensalidade()**, responsável por simular o pagamento da mensalidade do aluno.

```
1 public class Bolsista extends Aluno {
2
3     Bolsista(String nome, String cpf, int idade, String matricula){
4         super(nome, cpf, idade, matricula);
5     }
6
7
8     @Override
9     public void pagarMensalidade() {
10        System.out.println(this.nome + " pagou a mensalidade com um desconto de 50%");
11    }
12 }
```

```

1  public class Regular extends Aluno {
2
3      Regular(String nome, String cpf, int idade, String matricula){
4          super(nome, cpf, idade, matricula);
5      }
6
7      @Override
8      public void pagarMensalidade() {
9          System.out.println(this.nome + " pagou a mensalidade integral!");
10     }
11 }

```

Já na classe **Professor**, foi adicionado o atributo *centro*, representando o departamento acadêmico, além do método **darAula()**, que simula a atuação profissional do docente.

```

1  public class Professor extends Pessoa{
2      String centro;
3
4      Professor(String nome, String cpf, int idade, String centro){
5          super(nome, cpf, idade);
6          this.centro = centro;
7      }
8
9      public void darAula(){
10         System.out.println(this.nome + " diz: Olá, turma. Vamos dar início à aula?");
11     }
12
13     @Override
14     public String toString(){
15         return this.nome + " (Centro: " + this.centro + ")";
16     }
17 }
18

```

Nas classes **Aluno** e **Professor**, também foi implementado o método **toString()**, possibilitando que, ao imprimir objetos dessas classes, sejam exibidas informações relevantes sobre o indivíduo, ao invés do endereço de memória do objeto. Isso contribui para uma melhor interpretação e visualização dos dados no console.

Classe Aluno

```

12
13     @Override
14     public String toString(){
15         return this.nome;
16     }
17 }

```

Classe Professor

```

12
13     @Override
14     public String toString(){
15         return this.nome + " (Centro: " + this.centro + ")";
16     }
17 }
18

```