

CAS – SENAC

2025

Atividade número: 5

Nome do aluno: Eduardo Teixeira da Silva

Turma: TADS 4ND

Atividade 1

Interface Estado

```
public interface Estado {  
  
    void abaixando();  
    void esperando();  
    void pulando();  
    void correndo();  
}
```

Classe Pulando

```
public class Pulando implements Estado{  
  
    private String nome = "Pulando";  
    Personagem personagem;  
  
    public Pulando(Personagem personagem) {  
        this.personagem = personagem;  
    }  
  
    @Override  
    public void abaixando() {  
        personagem.setEstado(new Abaixando(personagem));  
    }  
  
    @Override  
    public void esperando() {  
        personagem.setEstado(new Esperando(personagem));  
    }  
  
    @Override  
    public void pulando() {  
        System.out.println("Pulando...");  
    }  
  
    @Override  
    public void correndo() {  
        personagem.setEstado(new Correndo(personagem));  
    }  
  
    @Override  
    public String toString() {  
        return "Pulando{" + "nome=" + nome + '}';  
    }  
}
```

Classe Esperando

```
public class Esperando implements Estado{

    private String nome = "Esperando";
    Personagem personagem;

    public Esperando(Personagem personagem) {
        this.personagem = personagem;
    }

    @Override
    public void abaixando() {
        personagem.setEstado(new Abaixando(personagem));
    }

    @Override
    public void esperando() {
        System.out.println("Esperando...");
    }

    @Override
    public void pulando() {
        personagem.setEstado(new Pulando(personagem));
    }

    @Override
    public void correndo() {
        personagem.setEstado(new Correndo(personagem));
    }

    @Override
    public String toString() {
        return "Esperando{" + "nome=" + nome + '}';
    }

}
```

Classe Correndo

```
public class Correndo implements Estado{

    private String nome = "Correndo";
    Personagem personagem;

    public Correndo(Personagem personagem) {
        this.personagem = personagem;
    }

    @Override
    public void abaixando() {
        personagem.setEstado(new Abaixando(personagem));
    }

    @Override
    public void esperando() {
        personagem.setEstado(new Esperando(personagem));
    }

    @Override
    public void pulando() {
        personagem.setEstado(new Pulando(personagem));
    }

    @Override
    public void correndo() {
        System.out.println("Correndo...");
    }

    @Override
    public String toString() {
        return "Correndo{" + "nome=" + nome + " }";
    }

}
```

Classe Abaixando

```
public class Abaixando implements Estado{

    private String nome = "Abaixando";
    Personagem personagem;

    public Abaixando(Personagem personagem) {
    }

    @Override
    public void abaixando() {
        System.out.println("Abaixado...");
    }

    @Override
    public void esperando() {
        personagem.setEstado(new Esperando(personagem));
    }

    @Override
    public void pulando() {
        personagem.setEstado(new Pulando(personagem));
    }

    @Override
    public void correndo() {
        personagem.setEstado(new Correndo(personagem));
    }

    @Override
    public String toString() {
        return "Abaixando{" + "nome=" + nome + '}';
    }

}
```

Classe Personagem

```
public class Personagem {

    private String nome;
    private Estado estado = new Esperando(this);

    public Personagem(String nome) {
        this.nome = nome;
    }

    public void requisitarAbaixando() {
        estado.abaixando();
    }
    public void requisitarCorrendo() {
        estado.correndo();
    }
    public void requisitarEsperando() {
        estado.esperando();
    }
    public void requisitarPulando() {
        estado.pulando();
    }

    public String getNome() {
        return nome;
    }

    public void setNome(String nome) {
        this.nome = nome;
    }

    public Estado getEstado() {
        return estado;
    }

    public void setEstado(Estado estado) {
        this.estado = estado;
    }

    @Override
    public String toString() {
        return "Tarefa{" + "estado=" + estado + '}';
    }
}
```

Main

```
public class Main {  
    public static void main(String[] args) {  
  
        Personagem p1 = new Personagem("Julia");  
        p1.requisitarCorrendo();  
        System.out.println(p1.getEstado());  
        p1.requisitarPulando();  
        System.out.println(p1.getEstado());  
        p1.requisitarEsperando();  
        System.out.println(p1.getEstado());  
        p1.requisitarCorrendo();  
        System.out.println(p1.getEstado());  
        p1.requisitarAbaixando();  
        System.out.println(p1.getEstado());  
    }  
}
```

Console

```
run:  
Correndo{nome=Correndo}  
Pulando{nome=Pulando}  
Esperando{nome=Esperando}  
Correndo{nome=Correndo}  
Abaixando{nome=Abaixando}  
BUILD SUCCESSFUL (total time: 0 seconds)  
|
```

Atividade 2

Interface Pagamento

```
public interface Pagamento {  
  
    void pconfirmado();  
    void ppreparacao();  
    void pcancela();  
    void pEnvia();  
}
```

Classe PConfirmado

```
public class PConfirmado implements Pagamento {

    private String nome = "Pagamento Confirmado";
    private Processamento processamento;

    public PConfirmado(Processamento processamento) {
        this.processamento = processamento;
    }

    @Override
    public void pconfirmado() {
        System.out.println("Pagamento confirmado...");
    }

    @Override
    public void ppreparacao() {
        processamento.setPagamento(new PPreparacao(processamento));
    }

    @Override
    public void pcancela() {
        processamento.setPagamento(new PCancela(processamento));
    }

    @Override
    public void pEnvia() {
        processamento.setPagamento(new PEnvia(processamento));
    }

    @Override
    public String toString() {
        return "PConfirmado{" + "nome=" + nome + '}';
    }
}
```

Classe PPreparacao

```
public class PPreparacao implements Pagamento {

    private String nome = "Preparando pedido...";
    private Processamento processamento;

    public PPreparacao(Processamento processamento) {
        this.processamento = processamento;
    }

    @Override
    public void pconfirmado() {
        processamento.setPagamento(new PConfirmado(processamento));
    }

    @Override
    public void ppreparacao() {
        System.out.println("Preparando pedido...");
    }

    @Override
    public void pcancela() {
        processamento.setPagamento(new PCancela(processamento));
    }

    @Override
    public void pEnvia() {
        processamento.setPagamento(new PEnvia(processamento));
    }

    @Override
    public String toString() {
        return "PPreparacao{" + "nome=" + nome + '}';
    }
}
```


Classe PEnvia

```
public class PEnvia implements Pagamento {

    private String nome = "Pedido enviado...";
    private Processamento processamento;

    public PEnvia(Processamento processamento) {
        this.processamento = processamento;
    }

    @Override
    public void pconfirmado() {
        processamento.setPagamento(new PConfirmado(processamento));
    }

    @Override
    public void ppreparacao() {
        processamento.setPagamento(new PPreparacao(processamento));
    }

    @Override
    public void pcancela() {
        processamento.setPagamento(new PCancela(processamento));
    }

    @Override
    public void pEnvia() {
        System.out.println("Pedido enviado...");
    }

    @Override
    public String toString() {
        return "PEnvia{" + "nome=" + nome + '}';
    }
}
```

Classe PCancela

```
public class PCancela implements Pagamento {

    private String nome = "Cancelando Pedido...";
    private Processamento processamento;

    public PCancela(Processamento processamento) {
        this.processamento = processamento;
    }

    @Override
    public void pconfirmado() {
        processamento.setPagamento(new PConfirmado(processamento));
    }

    @Override
    public void ppreparacao() {
        processamento.setPagamento(new PPreparacao(processamento));
    }

    @Override
    public void pcancela() {
        System.out.println("Cancelando Pedido...");
    }

    @Override
    public void pEnvia() {
        processamento.setPagamento(new PEnvia(processamento));
    }

    @Override
    public String toString() {
        return "PCancela{" + "nome=" + nome + '}';
    }
}
```

Classe Processamento

```
public class Processamento {
    private String nome;
    private Pagamento pagamento = new PPreparacao(this);

    public Processamento(String nome) {
        this.nome = nome;
    }

    public void requisitarPconfirmado() {
        pagamento.pconfirmado();
    }

    public void requisitarCancelarPedido() {
        pagamento.pcancela();
    }

    public void requisitarPreparPedido() {
        pagamento.ppreparacao();
    }

    public void requisitarPenviado() {
        pagamento.pEnvia();
    }

    public String getNome() {
        return nome;
    }

    public void setNome(String nome) {
        this.nome = nome;
    }

    public Pagamento getPagamento() {
        return pagamento;
    }

    public void setPagamento(Pagamento pagamento) {
        this.pagamento = pagamento;
    }

    @Override
    public String toString() {
        return "Processamento{" + ", pagamento=" + pagamento + '}';
    }
}
```

Main

```
public class Main {
    public static void main(String[] args) {
        Processamento pl = new Processamento("Nubank");
        pl.requisitarPconfirmado();
        System.out.println(pl.getPagamento());
        pl.requisitarPreparPedido();
        System.out.println(pl.getPagamento());
        pl.requisitarPenviado();
        System.out.println(pl.getPagamento());
        pl.requisitarCancelarPedido();
        System.out.println(pl.getPagamento());
    }
}
```

Console

```
BUILD SUCCESSFUL (total time: 0 seconds)
PConfirmado{nome=Pagamento Confirmado}
PPreparacao{nome=Preparando pedido...}
PEnvia{nome=Pedido enviado...}
PCancela{nome=Cancelando Pedido...}
```

Atividade 3

Interface Estado

```
public interface Estado {

    public void atrasada();

    public void pendente();

    public void concluida();

}
```

Classe Atrasada

```
public class Atrasada implements Estado {  
  
    private String nome = "Atrasada";  
    private Tarefa tarefa;  
  
    public Atrasada(Tarefa tarefa) {  
        this.tarefa = tarefa;  
    }  
  
    @Override  
    public void atrasada() {  
        System.out.println("Já está atrasada");  
    }  
  
    @Override  
    public void pendente() {  
        System.out.println("Já está atrasada");  
    }  
  
    @Override  
    public void concluida() {  
        tarefa.setEstado(new Concluida(tarefa));  
    }  
  
    @Override  
    public String toString() {  
        return "Atrasada{" + "nome=" + nome + '}';  
    }  
}
```

Classe Concluida

```
public class Concluida implements Estado{
    private String nome = "Concluida";
    private Tarefa tarefa;

    public Concluida(Tarefa tarefa) {
        this.tarefa = tarefa;
    }

    @Override
    public void atrasada() {
        System.out.println("Já está concluída");
    }

    @Override
    public void pendente() {
        tarefa.setEstado(new Pendente(tarefa));
    }

    @Override
    public void concluida() {
        System.out.println("Já está concluída");
    }

    @Override
    public String toString() {
        return "Concluida{" + "nome=" + nome + '}';
    }
}
```

Classe Estado

```
public class Pendente implements Estado{
    private String nome="Pendente";
    private Tarefa tarefa;

    public Pendente(Tarefa tarefa){
        this.tarefa=tarefa;
    }

    @Override
    public void atrasada() {
        tarefa.setEstado(new Atrasada(tarefa));
    }

    @Override
    public void pendente() {
        System.out.println("Já está pendente");
    }

    @Override
    public void concluida() {
        tarefa.setEstado(new Concluida(tarefa));
    }

    @Override
    public String toString() {
        return "Pendente{" + "nome=" + nome + '}';
    }
}
```

Classe Tarefa

```
public class Tarefa {

    private static Tarefa instancia;
    private String nome;
    private String descricao;
    private Estado estado = new Pendente(this);
    private Tarefa(String nome, String descricao) {
        this.nome = nome;
        this.descricao = descricao;
    }

    public static Tarefa getInstancia(String nome, String descricao) {
        if (instancia == null) {
            instancia = new Tarefa(nome, descricao);
        }
        return instancia;
    }

    public void requisitarAtrasada() {
        estado.atrasada();
    }

    public void requisitarConcluida() {
        estado.concluida();
    }

    public void requisitarPendente() {
        estado.pendente();
    }

    public String getNome() {
        return nome;
    }

    public void setNome(String nome) {
        this.nome = nome;
    }

    public String getDescricao() {
        return descricao;
    }

    public void setDescricao(String descricao) {
        this.descricao = descricao;
    }

    public Estado getEstado() {
        return estado;
    }

    public void setEstado(Estado estado) {
        this.estado = estado;
    }

    @Override
    public String toString() {
        return "Tarefa{" + "estado=" + estado + '}';
    }
}
```


Classe UsaState (Main)

```
public class UsaState {  
  
    public static void main(String[] args) {  
        Tarefa tarefa = Tarefa.getInstancia("Gerson", "Implementar o pattern State com Singleton");  
        System.out.println(tarefa.getEstado());  
        tarefa.requisitarAtrasada();  
        System.out.println(tarefa.getEstado());  
        tarefa.requisitarConcluida();  
        System.out.println(tarefa.getEstado());  
    }  
}
```

Console

```
Build *  
Pendente{nome=Pendente}  
Atrasada{nome=Atrasada}  
Concluida{nome=Concluida}  
BUILD SUCCESSFUL (total time: 0 seconds)
```