

Programação Orientada a Objetos

Implementação de Relógio com um Atributo

Prof. Rui César das Neves, Prof. José Cordeiro

Departamento de Sistemas e Informática

Escola Superior de Tecnologia de Setúbal – Instituto Politécnico de Setúbal

2014/2015

Implementação de Relógio com um Atributo

```
public class Relogio {  
  
    /* Atributos */  
    private int segundosDesdeMeiaNoite;  
  
    /* Comportamento interno */  
    private int converterSegundosDesdeMeiaNoite(int horas, int minutos, int segundos) {  
        return horas * 3600 + minutos * 60 + segundos;  
    }  
  
    private void setSegundosDesdeMeiaNoite(int horas, int minutos, int segundos) {  
        segundosDesdeMeiaNoite = converterSegundosDesdeMeiaNoite(horas, minutos, segundos);  
    }  
  
    private int validar(int valor, int minimo, int maximo) {  
        if (valor < minimo) {  
            return minimo;  
        } else if (valor > maximo) {  
            return maximo;  
        } else {  
            return valor;  
        }  
    }  
}
```

Implementação de Relógio com um Atributo

```
/* Construtores */  
public Relogio() {  
    this(0, 0, 0);  
}  
  
public Relogio(int horas, int minutos) {  
    this(horas, minutos, 0);  
}  
  
public Relogio(int horas, int minutos, int segundos) {  
    setSegundosDesdeMeiaNoite(validar(horas, 0, 23),  
                               validar(minutos, 0, 59),  
                               validar(segundos, 0, 59));  
}
```

Implementação de Relógio com um Atributo

```
/* Inspectores ou Selectores */  
public int getHoras() {  
    return segundosDesdeMeiaNoite / 3600;  
}  
  
public int getMinutos() {  
    return (segundosDesdeMeiaNoite % 3600) / 60;  
}  
  
public int getSegundos() {  
    return segundosDesdeMeiaNoite % 60;  
}
```

Implementação de Relógio com um Atributo

```
/* Modificadores */  
public void setHoras(int horas) {  
    setSegundosDesdeMeiaNoite(validar(horas, 0, 23), getMinutos(), getSegundos());  
}  
  
public void setMinutos(int minutos) {  
    setSegundosDesdeMeiaNoite(getHoras(), validar(minutos, 0, 59), getSegundos());  
}  
  
public void setSegundos(int segundos) {  
    setSegundosDesdeMeiaNoite(getHoras(), getMinutos(), validar(segundos, 0, 59));  
}
```

Implementação de Relógio com um Atributo

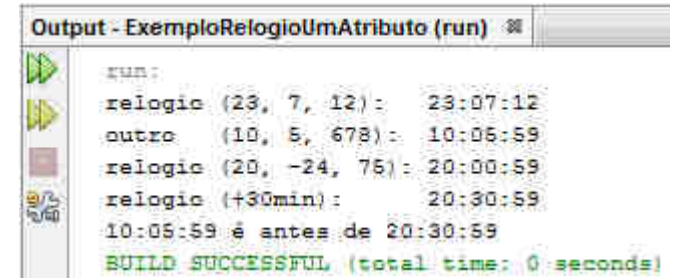
```
/* Comparador */
public boolean antes(Relogio relógio) {
    return this.segundosDesdeMeiaNoite < relógio.segundosDesdeMeiaNoite;
}

/* Transformador */
@Override
public String toString() {
    return String.format("%02d:%02d:%02d", getHoras(), getMinutos(), getSegundos());
}

/* Operação de Alto Nível
    Será substituído pelo passarDoTempo, quando se falar de Polimorfismo */
public void avancarMinutos(int maisMinutos) {
    segundosDesdeMeiaNoite += (maisMinutos * 60);
    segundosDesdeMeiaNoite %= 86400;
    //=(3600 * 23 + 60 * 59 + 59) + 1, preparar para uso de constantes (próxima semana)
}
}
```

Implementação de Relógio com um Atributo

```
public class ExemploRelogioUmAtributo {
    public static void main(String[] args) {
        Relogio relógio = new Relogio(23, 7, 12);
        Relogio outro = new Relogio(10, 5, 678);
        System.out.println("relogio (23, 7, 12): " + relógio);
        System.out.println("outro (10, 5, 678): " + outro);
        //Testar a validação
        relógio.setHoras(20);
        relógio.setMinutos(-24);
        relógio.setSegundos(75);
        System.out.println("relogio (20, -24, 75): " + relógio);
        //Testar avançar do tempo
        relógio.avancarMinutos(30);
        System.out.println("relogio (+30min): " + relógio);
        //Comparar relógios
        if (relógio.antes(outro)) {
            System.out.println(relógio + " é antes de " + outro);
        } else {
            System.out.println(outro + " é antes de " + relógio);
        }
    }
}
```



```
Output - ExemploRelogioUmAtributo (run)
run:
relogio (23, 7, 12): 23:07:12
outro (10, 5, 678): 10:05:59
relogio (20, -24, 75): 20:00:59
relogio (+30min): 20:30:59
10:05:59 é antes de 20:30:59
BUILD SUCCESSFUL (total time: 0 seconds)
```

Implementação de Relógio com um Atributo

```
/* Polimorfismo */
public void passarDoTempo() {
    /*
     * 1ª versão (antes do polimorfismo):
     segundosMeiaNoite++;
     segundosMeiaNoite %= 86400;
     //(3600 * 23 + 60 * 59 + 59) + 1, preparar para uso de constantes (próxima semana)
     */
    passarDoTempo(1);
}

public void passarDoTempo(int segundos) {
    segundosDesdeMeiaNoite += segundos;
    segundosDesdeMeiaNoite %= 86400;
    //(3600 * 23 + 60 * 59 + 59) + 1, preparar para uso de constantes (próxima semana)
}

public void passarDoTempo(int horas, int minutos, int segundos) {
    passarDoTempo(converterSegundosDesdeMeiaNoite(horas, minutos, segundos));
}
```