

## Atividade Prototype e Immutable

Aluno: Eduardo Richetti Bonatto RA: 2369990

Aluno: André Luis Quiosi RA: 2369958

1. Faz sentido a classe *SimpleMotorState* ter o método *clone*? Explique

Acredito que não, já que não é possível modificar os valores dos atributos da classe.

2. Refatore a class *SimpleMotorState* para que ela não produza objetos imutáveis

```
public class SimpleMotorState implements MotorState {
    public Status currentStatus = Status.OFF;
    public float rpm = 0.0f;
    public float accelerationFraction = 0.0f;

    public SimpleMotorState() {}

    public SimpleMotorState(
        Status currentStatus,
        float rpm,
        float accelerationFraction
    ) {
        super();
        this.currentStatus = currentStatus;
        this.rpm = rpm;
        this.accelerationFraction = accelerationFraction;
    }

    @Override
    public Status currentStatus() {
        return currentStatus;
    }

    @Override
    public float rotationsPerMinute() {
        return rpm;
    }

    @Override
    public float accelerationFraction() {
        return accelerationFraction;
    }

    @Override
    public MotorState clone() {
        try {
            Object clone = super.clone();
            return (MotorState) clone;
        } catch (CloneNotSupportedException e) {
            e.printStackTrace();
            return null;
        }
    }

    private boolean areEquals(float valueA, float valueB) {
```

```

        final float difference = valueA - valueB;
        return Math.abs(difference) < 0.000_000_1;
    }

    @Override
    public boolean equals(Object obj) {
        SimpleMotorState other = (SimpleMotorState) obj;

        return (
            this.currentStatus() == other.currentStatus() &&
            areEquals(this.rotationsPerMinute(), other.rotationsPerMinute()) &&
            areEquals(this.accelerationFraction(), other.accelerationFraction())
        );
    }
}

```

3. Como é possível que *AbstractMotor* acesse os atributos de *SimpleMotorState*?

A classe *AbstractMotor* pode acessar os atributos de *SimpleMotorState* porque eles não foram declarados como privados, mas sim como públicos.

4. Onde ocorre o padrão de projeto Template Method? Explique

O padrão Template Method ocorre nos métodos *getPower()* e *determineRotationsPerMinute()* da classe *AbstractMotor*, já que eles são métodos abstratos, que devem ser implementados pelas subclasses.