

- Princípio da Responsabilidade Única (SRP), a classe 'Product' é responsável por pelo produto e por calcular o desconto, indo contra a separação de responsabilidades. Eu separar colocaria calcular o desconto em uma classe separada, criando a classe 'CalcularDesconto', e manteria a classe 'Product' apenas com a representação dos produtos.
- Princípio da Responsabilidade Única (SRP), a classe 'Employer' está responsável por criar um funcionário e por calcular o salário e gerar um relatório, desobedecendo a divisão de responsabilidades. Assim, podemos separar as responsabilidades em classes separadas- 'CalcularSalario' 'ReportGenerator' e 'Employer', sendo que a classe 'Employer' vai criar um funcionário.
- Princípio de Liskov (LSP). 'EstudanteDePosGraduacao' está herdando de 'Estudante' e adicionando o método 'entregarTCC', desrespeitando a expectativa de comportamento definida pela classe pai. Temos que reescrever o código para que a classe 'EstudanteDePosGraduacao' não herde diretamente 'Estudante', e criando uma interface específica para os alunos de pós-graduação e lá terá a funcionalidade de entregarTCC.