# Roteiro de exercícios sobre Refatoração, disciplina Engenharia de Software e Sistemas

## Paulo Borba

## Centro de Informática

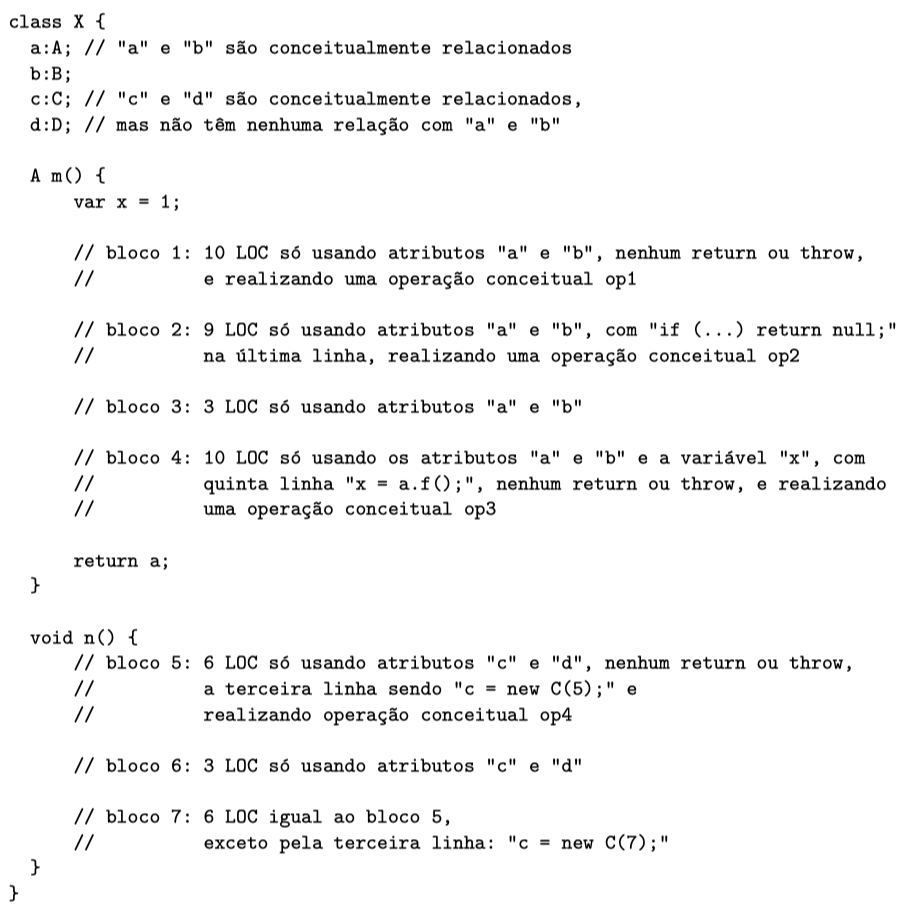
## Universidade Federal de Pernambuco

Adicione as respostas imediatamente após as perguntas, usando uma outra cor. Leia com calma e atenção as instruções, e reflita com cautela sobre as consequências de cada comando. As instruções abaixo assumem que toda a parte de instalação dos roteiros anteriores foram concluídas. É também assumido que todas as referência (manuais, tutoriais, etc.) mencionadas nos roteiros anteriores já foram consultadas, e que os alunos devem estar prontos para rapidamente consultá-las novamente durante o resto do curso, inclusive quando seguindo o roteiro abaixo.

## Aula 1

1. Qual refatoração do [catálogo de refatorações](https://www.refactoring.com/catalog/index.html) (também veja esse [outro catálogo](https://refactoring.guru/refactoring/techniques), que tem detalhes adicionais para algumas refatorações) poderia ser utilizada para retirar de uma classe Aluno informações como rua, número, cep, cidade, e estado? Como ficaria o código após a aplicação da refatoração (cole abaixo o novo código)? Quais as vantagens de aplicar essa refatoração nessa situação?
2. No *commit* "refatoracao do metodo gravar" do [teachingassistant](https://github.com/pauloborba/teachingassistant) (selecione o *branch* específico do período atual) o desenvolvedor alterou o método gravar. Que refatoração do [catálogo de refatorações](https://www.refactoring.com/catalog/index.html) foi utilizada nesse caso? Quais as vantagens de aplicar essa refatoração nessa situação? **(Opcional -->)** Escreva um teste (idealmente de classe) que evidencie que o comportamento foi preservado pela refatoração e cole o código logo abaixo. Execute esse teste (você vai precisar instalar as dependências e arquivos necessários para execução de testes de classe, já que no *commit* em questão ainda não temos isso) e cole abaixo o resultado da execução.
3. No *commit* "versao inicial do servidor, apos extracao de aluno para area comum", o método clone da classe Aluno foi modificado. Essa modificação pode ser considerada uma refatoração? Por que? Que refatorações do [catálogo de refatorações](https://www.refactoring.com/catalog/index.html) foram utilizadas nesse caso? Quais as vantagens de aplicar as mudanças nessa situação?
4. Compare as refatorações *Extract* e *Inline* *method*, explicando a relação entre elas e em que situações cada uma deve ser aplicada.
5. No *commit* "refatoracao dos testes de aceitacao", que refatorações do [catálogo de refatorações](https://www.refactoring.com/catalog/index.html) foram utilizadas nesse caso? Quais as vantagens de aplicar as mudanças nessa situação?

Aula 2

1. Compare as refatorações *Decompose Conditional* e *Consolidate Conditional Expression*, explicando a relação entre elas e em que situações cada uma deve ser aplicada.
2. Quais as vantagens de aplicar a refatoração *Encapsulate Field*?
3. Quais as vantagens de aplicar a refatoração *Replace Subclass with Fields*?
4. Responda as seguintes perguntas: (a) Quais os principais mau cheiros associados ao código a seguir? (b) Que refatorações você usaria para melhorar este código? (c) Qual o código resultante das refatorações? (d) Por que você acha que o comportamento do sistema seria preservado por essas refatorações? (e) Como você veriﬁcaria isso na prática?
5. Solicite, a um colega de turma, o *link* para a implementação de uma funcionalidade implementada por ele como resposta de um dos exercícios de implementação do projeto do roteiro anterior.
   1. Cole o *link* logo abaixo.
   2. Identifique, e liste logo abaixo, que refatorações deveriam ser aplicadas na funcionalidade implementada pelo seu colega.