

1.
 - a. **Manutenção**, ou seja, facilita a manutenção do código ao encapsular detalhes de implementação, permitindo alterações internas sem afetar outras partes do sistema.
 - b. **Reutilização**, promovendo a reutilização de código ao fornecer interfaces claras, tornando as classes mais adaptáveis a diferentes contextos
 - c. **Redução de complexidade**, a legibilidade do código se torna mais direta, uma vez que se oculta informações distratoras e desnecessárias
2. Ele estará melhorando o **acoplamento** do código, uma vez que resolverá o bug que está ligado ao relacionamento das classes A e B. Por outro lado, ele estará piorando a **coesão** do código, uma vez que as classes tornam-se mais dependentes entre si.
3. Como teremos muitas classes simples, leia-se, independentes, teremos uma baixa **coesão**. A partir disto, pode-se deduzir que o sistema possui um **acoplamento** alto, o que aumenta diretamente a complexidade da manutenção de certas partes do sistema.
4.
 - a. **Acoplamento aceitável**: ocorre quando classes possuem somente dependências necessárias, via uma interface confiável e estável.
 - b. **Acoplamento ruim**: ocorre quando existem fortes dependências entre classes, onde uma das classes possui uma interface instável ou a dependência não é feita através de uma interface.
 - c. **Acoplamento estrutural**: ocorre quando uma classe depende explicitamente de outra, podendo ser um acoplamento bom ou ruim.
 - d. **Acoplamento evolutivo (ou lógico)**: ocorre quando uma classe não depende explicitamente de outra, porém alterações em uma classe podem afetar a outra. Este acoplamento é sempre ruim.
5.
 - a. **Aceitável**: Uma classe A que precisa de informações em um modelo e uma classe B que é responsável por alterar e prover estas informações. Desta forma, A pode usar B para buscar estes dados sem causar inconsistências no modelo.

- b. **Ruim:** Uma classe A que precisa de informações em um modelo, que acessa esse modelo diretamente e uma classe B que faz alterações no modelo sem A ter ciência disto. Desta forma, o modelo pode ser alterado e A continuará tentando buscar pelas informações erroneamente.
- 6. Sim é possível, porém este acoplamento será **ruim** em todos os casos.
- 7. Ele tem um problema de **coesão**, pois somente o método main fica responsável por resolver todo o problema (isto considerando um problema grande e complexo). Por outro lado, torna-se impossível ter problemas de acoplamento no código, justamente por não ter diferentes objetos para se relacionarem, ou seja, não há problemas de acoplamento, pois não existe acoplamento.