

1. O princípio de Demeter. Eu criaria um método para encapsular todas essas chamadas.
2. Eu permitiria que a função imprime recebesse as mensagens via argumento.
3. O princípio de Liskov, uma vez que alguma classe que usar Calculadora talvez não poderá usar CalculadoraRápida
4. O princípio da responsabilidade única. Cada entidade (classe) é responsável por uma única tarefa. Por outro lado, um problema desta solução é a necessidade de várias entidades diferentes para uma tarefa.
5. Um exemplo interessante seria a criação de um blog pessoal que acesse um banco de dados via uma interface. Ou seja, estaríamos fazendo uso de Dependency Inversion Principle. Isso talvez fosse desnecessário em um contexto de blog pessoal visto que traria muitas complexidades, apenas para que a aplicação seja capaz de alterar de banco de dados facilmente, sendo que dificilmente surgiria essa necessidade dado a natureza do problema.