



1. Em que cenários o padrão *Factory Method* é bastante utilizado? Justifique.
2. Quais as consequências de usar o padrão *Singleton*?
3. Em quais situações devemos usar o padrão *Adapter*?
4. ([www.inf.ufes.br/~vitorsouza/pt](http://www.inf.ufes.br/~vitorsouza/pt)) Considere uma aplicação com interface gráfica que é implementada para plataformas diferentes (Motif para UNIX e outros ambientes para Windows e MacOS). As inúmeras classes implementando os elementos gráficos (Botão, Janela, etc.) não podem ser definidas estaticamente no código cliente por meio do operador *new*. Precisamos de uma implementação diferente de cada elemento gráfico para cada ambiente. Até em um mesmo ambiente, gostaríamos de dar a opção ao usuário de implementar diferentes aparências (*look-and-feels*). O código cliente não sabe qual classe concreta utilizamos, ele conhece apenas sua interface.
  - a) Indique o padrão mais adequado a essa necessidade.
  - b) Apresente a estrutura de classes do padrão indicado representando o cenário descrito;
  - c) Implemente o código da solução.
5. O sistema de folha de pagamento de uma grande empresa faz controle de todos os tipos de funcionários, onde cada funcionário possui, como propriedades, matrícula, nome, sexo, data de nascimento, cargo e função, e as operações receber salário, bater ponto e trabalhar. A alta direção resolver aceitar estagiários em seus quadros, porém, a princípio, eles não fariam parte da folha de pagamento e receberiam por projetos avulsos. Assim, criou-se uma estrutura para estagiários com matrícula, nome, sexo e data de nascimento, e as operações receber bolsa, registrar entrada, registrar saída e estagiar. Meses depois, a direção resolveu incluir os estagiários no sistema da folha de pagamento, entretanto, o sistema não deve sofrer qualquer modificação para se adaptar a nova realidade. Pede-se:
  - a) Indique qual o padrão mais adequado a essa necessidade;
  - b) Apresente a estrutura de classes para solucionar esse problema dentro do padrão escolhido;
  - c) Implemente o código da solução.
6. Em um sistema bancário, é necessário fazer o *log* de todas as operações realizadas, para fins de auditoria, ou seja, precisa-se de uma infraestrutura onde um único objeto seja responsável pelo registro das operações realizadas em toda a aplicação, gravando a data/hora, o tipo de operação realizada e a matrícula do usuário que realizou a operação. A solução deve funcionar adequadamente mesmo que várias *threads* estejam executando simultaneamente.
  - a) Indique o padrão mais adequado a essa necessidade.
  - b) Apresente a estrutura de classes do padrão indicado representando o cenário descrito.
  - c) Complemente a classe *Logger* abaixo de tal modo que atenda as especificações acima.

```
class Logger {  
    public void log (Date dt, User user, Operation oper) {  
        // rotina de logging  
    }  
    public Log query (Id id) {  
        // rotina de consulta de log  
    }  
    public void report (Date dt) {  
        // rotina de emissão de relatório  
    }  
    ...  
}
```

7. (<http://www.dsc.ufcg.edu.br/~jacques/cursos/map/html/pat/template.htm>) Uma equipe de desenvolvimento se depara com a seguinte tarefa: escrever uma classe para controlar o login de usuários para as diversas aplicações da empresa. A solução genérica seria um *framework* de login, ou seja, um esqueleto de login que seja comum a qualquer tarefa de login, e eventuais recursos para permitir que cada aplicação customize o processo de login. Os passos genéricos de login são:
1. **Prompt** para o usuário fornecer sua identificação (ID) e senha;
  2. **Autenticação** do ID e senha;
  3. **Aviso visual** de progresso indicando que a autenticação está sendo realizada;
  4. Um **aviso de sucesso** ao resto da aplicação de que o login foi realizado.
- A lógica das etapas 1 e 3 não muda. A lógica das etapas 2 e 4 pode variar muito entre aplicações. Cada aplicação deverá prover seu próprio código para esses casos particulares. Pede-se:
- a) Indique o padrão mais adequado a essa necessidade.
  - b) Apresente a estrutura de classes do padrão indicado representando o cenário descrito;
  - c) Implemente o código da solução.
8. ([www.inf.ufes.br/~vitorsouza/pt](http://www.inf.ufes.br/~vitorsouza/pt)) Escreva um programa que conte até 10 e envie os números para uma ferramenta de log. Esta ferramenta de log deve permitir a escolha entre dois tipos de log: em arquivo (log.txt) ou diretamente no console. A escolha deve ser por um parâmetro passado ao programa ("arquivo" ou "console"). Observe que a classe de log não sabe, a priori, se deve ser criado um log em arquivo ou em console. Portanto, o design da solução deve ser flexível o suficiente para isso. Na realidade, a superclasse de log quer que delegar para suas subclasses a criação dos objetos reais (log de arquivo e log de console).
- a) Indique o padrão mais adequado a essa necessidade.
  - b) Apresente a estrutura de classes do padrão indicado representando o cenário descrito;
  - c) Implemente o código da solução.