1. O que nos motiva utilizar o padrão *Iterator*?

2. Descreva os participantes do padrão *Proxy*.

3. Qual a aplicabilidade do padrão *State*?

4. (http://brizeno.wordpress.com/2011/11/21/mao-na-massa-state/) Uma empresa de criação de games deseja criar um jogo com o personagem Mario Bros. Durante o jogo, acontecem várias trocas de estado com o Mario, por exemplo, ao pegar uma flor de fogo, o Mario pode crescer, se estiver pequeno, e ficar com a habilidade de soltar bolas de fogo. Desenvolvendo um pouco mais o raciocínio, temos um conjunto grande de possíveis estados, e cada transição depende de qual é o estado atual do personagem. Ao pegar uma flor de fogo, por exemplo, podem acontecer quatro ações diferentes, dependendo de qual é o estado atual do Mario:
   - Se Mario pequeno -> Mario grande e Mario fogo
   - Se Mario grande -> Mario fogo
   - Se Mario fogo -> Mario ganha 1000 pontos
   - Se Mario capa -> Mario fogo

   Todas estas condições devem ser checadas para realizar esta única troca de estado. O jogo deve contemplar os seguintes estados: Mario pequeno, Mario grande, Mario flor e Mario capa. As seguintes ações são possíveis: Pegar Cogumelo, Pegar Flor, Pegar Pena e Levar Dano. Abaixo seguem os resultados da execução de cada ação dependendo do estado do personagem:

   Pegar Cogumelo:
   - Se Mario pequeno -> Mario grande
   - Se Mario grande -> 1000 pontos
   - Se Mario fogo -> 1000 pontos
   - Se Mario capa -> 1000 pontos

   Pegar Flor:
   - Se Mario pequeno -> Mario grande e Mario fogo
   - Se Mario grande -> Mario fogo
   - Se Mario fogo -> 1000 pontos
   - Se Mario capa -> Mario fogo

   Pegar Pena:
   - Se Mario pequeno -> Mario grande e Mario capa
   - Se Mario grande -> Mario capa
   - Se Mario fogo -> Mario fogo
   - Se Mario capa -> 1000 pontos

   Levar Dano:
   - Se Mario pequeno -> Mario morto
   - Se Mario grande -> Mario pequeno
   - Se Mario fogo -> Mario grande
   - Se Mario capa -> Mario grande

   A empresa não quer investir tempo numa solução tradicional que adote, por exemplo, comandos condicionais grandes para as trocas de estado. 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.

5. Uma ferramenta utiliza diversos botões em sua interface, como “visualizar impressão” e “incluir imagem”, por exemplo, que ficam visíveis assim que a interface da ferramenta é exibida ao usuário. Porém, por questões de performance, as funcionalidades associadas a tais botões não são carregadas no momento da visualização, e sim quando o botão é pressionado. 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. Uma empresa de TI adota uma estrutura padronizada para acessar sequencialmente elementos de cadeias de objetos, chamada SequenceAccess. Todos os tipos de cadeias seguem o mesmo formato de acesso definindo em SequenceAccess, com métodos como primeiro(), para retornar o primeiro elemento, último() para retornar o último, próximo() para retornar o próximo e um teste temPróximo() para verificar se há um próximo elemento na cadeia, de modo que todos as cadeias de dados são acessadas da mesma maneira pelos usuários. Foi adquirido recentemente um *framework* de acesso a dados chamado DataAll. Deseja-se adequar os métodos de acesso a dados do novo *framework* ao padrão de acesso da empresa, permitindo acessar sequencialmente os elementos armazenados em banco de dados, de modo que os usuários acessem tais dados utilizando a mesma estrutura padrão da empresa, sem tomar conhecimento do *framework* por trás dos acessos. A classe DataAll conta com os seguintes métodos de acesso aos dados: getFirst(), getLast(), getNext() e hasNext(). 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.