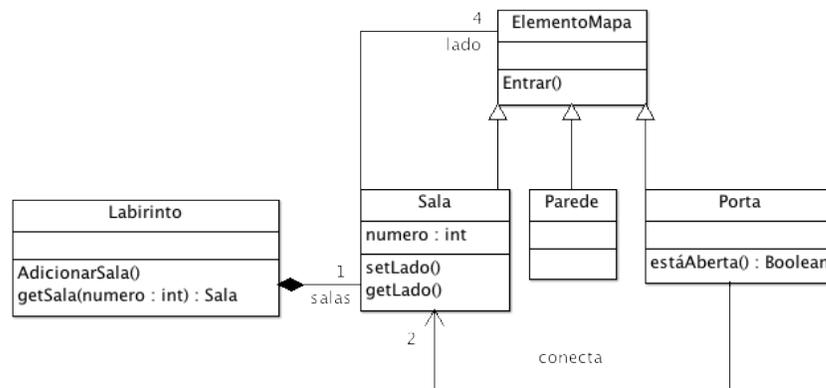


Instituto de Computação
Bacharelado em Sistemas de Informação
Disciplina: Tópicos em Projeto e Implementação de Sistemas I
Segunda lista de exercícios - 2014.1

- 1) Explique o conceito de padrões de projeto e o diferencie de *Frameworks* e APIs.
- 2) Compare o uso dos conceitos de herança e composição em projeto orientado a objetos.
- 3) Explique o conceito de delegação.
- 4) O método `Arrays.sort()` do Java é um exemplo do padrão Strategy. Explique por que.
- 5) Escreva uma classe que implemente uma representação para uma mensagem de correio eletrônico (*e-mail*). Uma mensagem de *e-mail* tem um conteúdo na forma de uma cadeia de caracteres. Suponha que o e-mail tenha que possuir novas características: primeiro, o e-mail deve ter um descritor da companhia adicionado ao final do conteúdo original; segundo, o e-mail deve permitir o uso de mensagens criptografadas. Proponha uma solução em que os diferentes tipos de e-mail possam ser utilizados em um sistema cliente.
- 6) Considere um Joystick, onde cada botão pode ser configurado para diferentes ações em um jogo. No jogo, o personagem principal pode estar a pé ou em um veículo. A pé o personagem pode avançar, virar à direita, virar à esquerda, recuar, saltar, pegar um objeto e efetuar uma ação que depende do contexto (uma ação genérica, como abrir uma porta ou dar um golpe). Quando em um veículo, as ações se restringem a entrar, sair, acelerar, frear, virar para esquerda, virar para a direita e dar ré. Descreva uma solução que proporcione o desacoplamento do tratamento de evento de pressionar cada um dos botões do joystick do conhecimento da ação e de quem realiza a ação. Considere um Joystick com 10 botões, com 4 botões em cruz para movimentos e 4 botões de ação e dois para ações especiais na parte frontal. Apesar dos botões serem classificados, assuma que seja possível fazer a configuração para cada botão arbitrariamente. Como você trataria uma ação nula?
- 7) Considere uma classe que representa a telemetria de um carro de Fórmula-1. O sistema de telemetria deve exibir, em duas interfaces de visualização, os dados correspondentes a velocidade, giro do motor, força de gravidade atuando sobre o piloto nas curvas, consumo de combustível e quilômetros rodados. A primeira interface é gráfica e apresenta os dados como gráficos de séries temporais; a segunda é um diálogo que simplesmente apresenta os valores instantâneos em forma textual. Apresente uma solução utilizando padrões para o problema em questão.

- 8) Considere a implementação de um labirinto que descreve o universo de um jogo. O labirinto é formado por um conjunto de salas. Cada sala conhece seus vizinhos que podem ser uma outra sala, uma parede ou uma porta para uma outra sala. Toda sala tem 4 vizinhos nas direções norte, sul, leste e oeste. Uma porta conecta sempre 2 salas. Um elemento do mapa representado por uma classe abstrata ElementoMapa é um uma abstração para todos os componentes de um labirinto.

Considere o seguinte diagrama de classes UML que descreve as relações entre as classes ElementoMapa, Sala, Parede, Porta e Labirinto.



- Apresente uma solução para o Labirinto tendo em mente a questão de como criar as instâncias do Labirinto.
- Refatore o seu código usando o padrão *Abstract Factory* para resolver o problema de instanciação do Labirinto.

Dica: estude a solução do livro Design Patterns de Eric Gamma et al.

- 9) Considere um editor de desenhos que permite o usuário desenhar elementos gráficos como linhas, polígonos e texto, em figuras e diagramas. A abstração chave é o conceito de

objeto gráfico, uma forma editável e capaz de se desenhar, representada por uma interface denominada Shape. A interface Shape propõe três métodos BoundingBox(), que retorna um retângulo que descreve a caixa envolvente da forma, CreateManipulator() que retorna uma instância de um objeto capaz de manipular a forma de modo apropriado e Draw(). Suponha ainda que o editor defina subclasses de Shape como PolygonShape e LineShape. O cliente do sistema, em um determinado momento, deseja que o editor seja capaz de exibir e manipular fontes de texto, uma TextShape. O código para TextShape é difícil de implementar. Por outro lado, existe uma API que disponibiliza uma classe denominada TextView para desenhar e editar textos. Infelizmente, TextView não atende a interface de Shape, já que propõe um interface diferente, com um método denominado GetExtent() ao invés de BoundingBox() e Display() ao invés de Draw(). Proponha uma solução para o problema de reuso de TextView usando o padrão de projeto apropriado. Forneça o diagrama de classes da sua solução.

10) Faça o diagrama de sequência UML que descreva o padrão MVC.

Dica: alguns exercícios foram baseados ou obtidos do livro Padrões de Projeto de Eric Gamma et al.