



Fundação CECIERJ - Vice Presidência de Educação Superior a Distância

Curso de Tecnologia em Sistemas de Computação

Disciplina de Arquitetura e Projeto de Sistemas

Gabarito da AP3 2º semestre de 2017.

Nome –

Assinatura –

Observações:

1. Prova sem consulta e sem uso de dispositivos (e.g., celular, calculadora, etc.).
 2. Use caneta para preencher o seu nome e assinar nas folhas de questões e nas folhas de respostas.
 3. Você pode usar lápis para responder as questões.
 4. Ao final da prova devolva as folhas de questões e as de respostas.
 5. Todas as respostas devem ser transcritas nas folhas de respostas. As respostas nas folhas de questões não serão corrigidas.
-

Questão 1 (2 pontos)

Indique o estilo arquitetural mais indicado para cada situação a seguir:

- a) Um sistema operacional, que separe os serviços disponíveis ao usuário dos serviços intermediários, e os serviços intermediários dos serviços mais próximos ao hardware, onde cada tipo de serviço só pode se comunicar com os serviços diretamente inferiores.
- b) Um compilador, que tem módulos responsáveis por análise léxica, análise sintática, análise semântica, otimização e geração de código, onde cada um desses módulos é executado em sequência e o resultado de um é entrada para o seguinte.

Resposta: a → Camadas; b → Pipes & Filters.

Questão 2 (3 pontos)

Relacione cada problema da coluna da esquerda com um e somente um padrão da coluna da direita que é o principal indicado para tratar o problema.

- | | |
|--|-------------------------|
| (a) Quem deveria ser responsável por tratar um evento de sistema? | (1) Padrão Polymorphism |
| (b) Em um sistema com centenas de classes, como selecionamos quais responsabilidades devem estar em quais classes? | (2) Padrão Creator |

- | | |
|---|-----------------------------------|
| (c) Como posso evitar o acoplamento direto? | (3) Padrão Information Expert |
| (d) Como tratar alternativas em função do tipo da classe? | (4) Padrão Low Coupling |
| (e) A quem atribuir uma responsabilidade quando todas as opções ferem os princípios de acoplamento baixo e coesão alta? | (5) Padrão Controller |
| (f) Como assegurar que uma classe tenha uma e somente uma instância e fornecer um ponto global de acesso para ela? | (6) Padrão High Cohesion |
| | (7) Padrão Pure Fabrication |
| | (8) Padrão Indirection |
| | (9) Nenhum dos padrões anteriores |

Resposta: a → 5; b → 3; c → 8; d → 1; e → 7; f → 9

Questão 3 (4 pontos)

Assinale verdadeiro ou falso às seguintes afirmativas, justificando.

- (a) Um filtro se comunica com um outro através de uma interface de software.
- (b) Uma camada pode se comunicar apenas com a(s) camada(s) que está ligada imediatamente abaixo.
- (c) Processos distribuídos se comunicam por mensagens.
- (d) Um sistema com processos distribuídos é restrito exclusivamente pela sincronização das mensagens.
- (e) Numa arquitetura orientada a objetos, a visão lógica e a visão de desenvolvimento devem tratar dos mesmos pacotes.
- (f) Numa arquitetura orientada a objetos, apesar de possuírem dados em comum, as visões são independentes.
- (g) Uma classe que agrupe as funcionalidades de um *framework* é desejada.
- (h) *Frameworks* caixa-preta facilitam a reutilização.

Resposta:

- (a) FALSO. Filtros são unidades independentes que não se comunicam.
- (b) FALSO. Esse tipo de restrição deve ser definida no projeto da arquitetura.
- (c) VERDADEIRO. A definição de um componente é um processo que envia e recebe mensagens.
- (d) FALSO. Há restrições topológicas e da corretude do roteamento.
- (e) FALSO. A visão de desenvolvimento pode incluir novos pacotes para tratar funcionalidades de baixo nível.
- (f) FALSO. A visão de cenários integra as outras visões e valida as demais de forma a garantir a corretude do modelo.
- (g) FALSO. Classes devem ser coesas, evitando agrupar estruturas independentes.
- (h) VERDADEIRO. É necessário conhecer apenas as interfaces para uso, sem detalhes de implementação.

Questão 2 (1 ponto)

Dado que um compilador é organizado em “etapas de compilação” independentes, qual o estilo arquitetural mais adequado à sua implementação? Justifique.

Resposta:

Como as etapas são independentes, o uso de filtros se mostra adequado, uma vez que ele pressupõe que unidades independentes processam uma entrada e o resultado é passado adiante para que a(s) próxima(s) unidade(s) efetue seu processamento.