

Exercícios CI1163 – Design de Software

- 1) O que é um padrão de software ?
- 2) O padrão Information Expert define princípios para a atribuição de responsabilidades aos objetos.
 - a) qual é o princípio central do padrão?
 - b) que benefícios tenho ao aplicar o padrão?
- 3) Considere as classes **Aluno**, **Turma** e **Matricula**, em um sistema de gerenciamento acadêmico. Uma *Turma* poderá ter várias *Matriculas*, cada uma referente a um único *Aluno*. Um *Aluno* poderá ter várias matrículas. Uma *Matricula* armazenará pelo menos a nota do *Aluno*. O sistema poderá calcular a nota de cada aluno. Defina o diagrama UML deste sistema, e explique como poderia usar o padrão *Information Expert* para especificar o método de cálculo da nota.
- 4) Considere um sistema de gestão de bibliotecas, contendo as classes *Livro*, *Usuario* e *Emprestimo*. Quem deveria ser o responsável pelo cálculo de multa de devolução ? E pela devolução do livro ? Explique o padrão seguido.
- 5) No caso do sistema acadêmico apresentado no exercício (3), quem deve ser o responsável pela criação das matrículas? Explique o padrão utilizado na escolha.
- 6) Porque devemos priorizar “alta coesão” e “baixo acoplamento” na aplicação de um padrão GRASP ?
- 7) O que é acoplamento de dados ? Cite um exemplo.
- 8) Cite um exemplo um acoplamento de dado global.
- 9) Que tipo de objeto geralmente deve ser responsável por receber eventos do sistema? Isto é definido em qual padrão ?
- 10) Porque não é adequado um objeto *Controller* realizar ações do domínio da aplicação?
- 11) Explique um exemplo de solução, usando o padrão *Pure Fabrication*, para armazenamento de dados em um repositório. Porque não usar estritamente o padrão *Information Expert* ?
- 12) Defina o código orientado a objeto para o exercício (3).
- 13) Explique uma vantagem em usar um gerador de código para criar código objeto a partir de um diagrama UML.
- 14) Dado o JSON do exercício 13, quais são os elementos específicos da linguagem e quais são os elementos variáveis?
- 15) Defina, sob a forma de um exemplo, uma nova linguagem textual para projetar um diagrama de classes UML. A linguagem deverá ter, no mínimo, classes, atributos e associações.
- 16) O padrão *Composite* define a estrutura de objetos relacionados em árvore. Defina um diagrama UML para representar uma relação de composição em um sistema de arquivos, contendo *Diretorios* e *Arquivos*.
- 17) O diagrama abaixo representa os elementos para implementação do padrão Singleton. Este padrão permite que apenas uma instância de uma determinada classe seja instanciada, setando seu construtor para nulo e criando um método (estático ou de instância) para recuperação dessa instância. Defina uma classe concreta seguindo este padrão, para instanciação de um *Log*.



- 18) O princípio SRP (Single Responsibility Principle) afirma que uma classe deve ter UM, e somente UM, motivo para mudar. Explique a afirmação acima
- 19) Porque o princípio SRP pode ser considerado mais restritivo que o padrão GRASP Especialista?
- 20) O código OO abaixo não segue o padrão SRP. Modifique este código para que siga o padrão. Novas classes poderão ser criadas, mas métodos e atributos não poderão ser excluídos.

```
class Turma {
    float calculaMedia(){/*...*/}
    List getAlunos(){/*...*/}
    int getTotalAlunos(){/*...*/}
    aluno add(Aluno aluno){/*...*/}
    void delete(Aluno aluno){/*...*/}
    void imprimeDados(){/*...*/}
    void mostraDados(){/*...*/}
    void load(){/*...*/}
    void save(){/*...*/}
    void update(){/*...*/}
    void delete(){/*...*/}
}
```
- 20) O código OO abaixo não segue o padrão SRP. Modifique este código para que siga o padrão. Novas classes poderão ser criadas, mas métodos e atributos não poderão ser excluídos.
- 21) Explique o princípio de inversão de dependência (DIP – Dependency Inversion Principle).
- 22) Dado o código abaixo:

```
class LembraSenha {
    private Connection dbConn;
    LembraSenha() {
        dbConnection = new MySQLConnection();
    }
}
```
- a) Explique porque ele não segue o princípio DIP.
- b) Adapte o código para seguir o princípio DIP.