Laboratório de Programação Orientada por Objetos (MIEIC)

8/6/2012

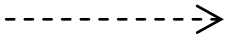
Mini-teste de escolha múltipla - Sem consulta - 60 Minutos

Assinala com um "X" a resposta correta a cada uma das perguntas seguintes. Cada pergunta só tem uma resposta correta. Respostas certas valem 1 ponto. Respostas erradas descontam 0.25 pontos.

Nome: _____

Diagramas de Classes UML

1. Nos diagramas de classes UML, o seguinte símbolo representa que tipo de relação entre classes?



	a. associação navegável
	b. concretização (*realization)*
	c. generalização
	d. dependência

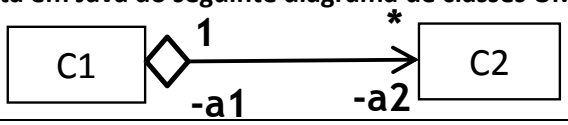
2. Nos diagramas de classes UML, a relação de concretização (*realization*) permite relacionar que tipos de elementos?

	a. objetos e classes
	b. classes concretas e classes abstratas
	c. classes e interfaces
	d. classes não genéricas e classes genéricas

3. Nos diagramas de classes UML, membros estáticos (atributos e operações) são representados:

	a. em itálico
	b. com sublinhado
	c. com um estereótipo
	d. com "/" antes do nome

4. Qual a tradução mais correta em Java do seguinte diagrama de classes UML:



	a. class C1 { private Set<C2> a2; } class C2 { }
	b. class C1 { private C2 a2; } class C2 { private HashSet<C1> a1; }
	c. class C1 { private List<C2> a2; } class C2 { private C1 a1; }
	d. class C1 { private List<C2> a2; class C2 { } }

Diagramas de Sequência UML

5. Os diagramas de sequência UML são mais apropriados para:

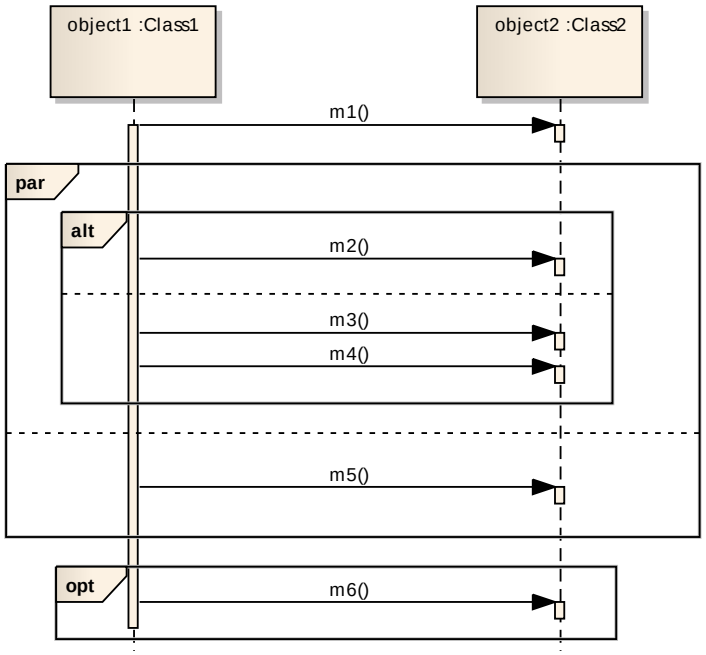
	a. Mostrar os passos envolvidos num algoritmo.
	b. Mostrar o ciclo de vida de objetos.
	c. Mostrar interações entre objetos num determinado contexto.
	d. Mostrar uma configuração de objetos e ligações entre objetos.

6. Que tipos de elementos podem ter linhas de vida num diagrama de sequência?

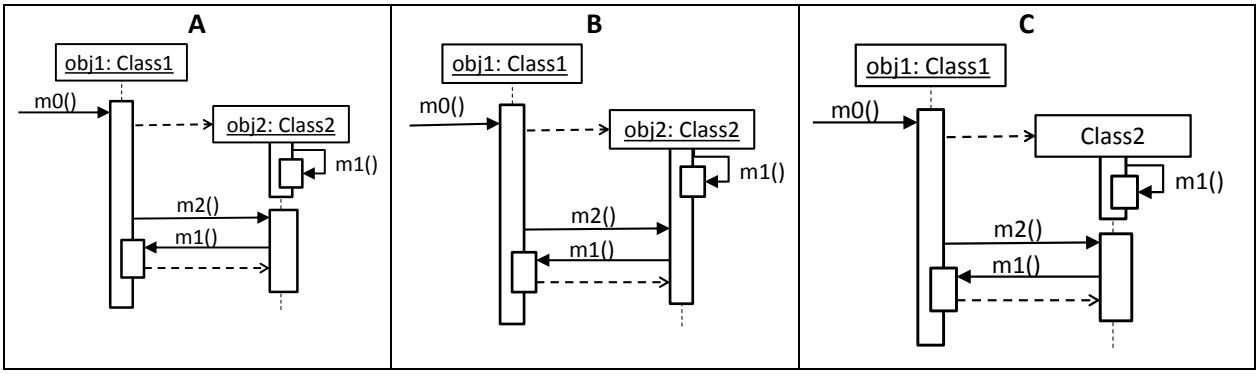
	a. Atores, classes, instâncias de atores, instâncias de classes e casos de utilização.
	b. Atores, classes, instâncias de atores e instâncias de classes.
	c. Qualquer elemento UML.
	d.. Apenas classes e instâncias de classes.

7. Qual das sequências de mensagens <u>não é</u> permitida pelo seguinte diagrama de sequência UML?

	a. m1 m5 m2
	b. m1 m3 m5 m4
	c. m1 m3 m5 m6
	d. m1 m2 m5 m6



8. Quais dos seguintes diagramas de sequência UML são sintaticamente válidos? (Nota: A mensagem *m0()* é válida)



	a. Nenhum
	b. Todos
	c. Só A
	d. Só A e C

Diagramas de Estados UML

9. Nos diagramas de estados UML, as transições podem ser etiquetadas com que elementos?

	a. evento, condição de guarda e ação
	b. evento, condição de guarda e atividade
	c. nome, evento, pré-condição e pós-condição
	d. evento e condição inicial

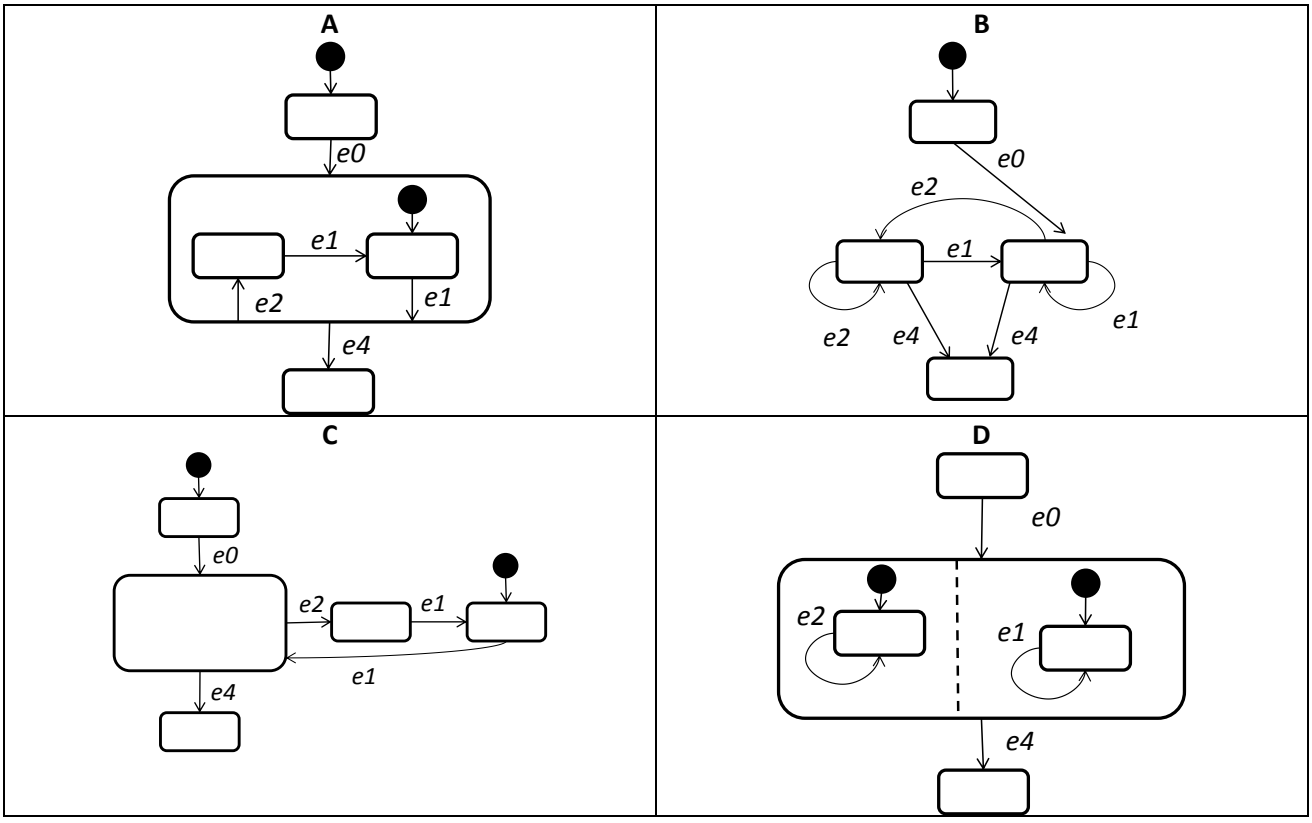
10. Nos diagramas de estados UML, estados compostos (com subestados sequenciais) são úteis para:

	a. evitar a explosão combinatório de estados
	b. evitar a explosão combinatório de transições
	c. modelar comportamentos repetitivos
	d. modelar eventos temporais

11. Nos diagramas de estados UML, quais dos seguintes tipos de eventos são aceites?

	a. chamada de operações, entrada em estado, saída de estado, início e fim
	b. chamada de operações, retorno de operações, eventos temporais e sinais
	c. chamada de operações, exceções, e *listeners*
	d. chamada de operações e eventos temporais

12. Quais dos seguintes diagramas de estados UML são equivalentes, no sentido de aceitarem as mesmas sequências de eventos?



	a. A e D
	b. A e C
	c. A, B e D
	d. A, B e C

Padrões de Desenho

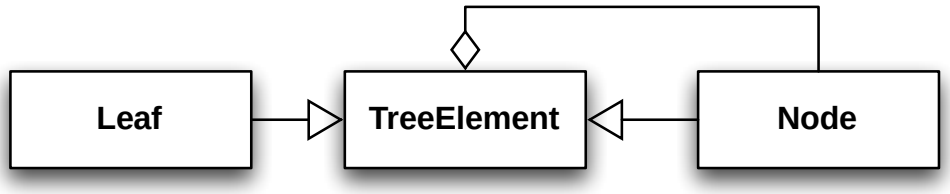
13. ***"Defines a family of algorithms, encapsulates each one, and make them interchangeable. It lets the algorithm vary independently from clients that use it."*** **Qual o padrão de desenho que tem este propósito (*intent*)?**

	a. COMPOSITE
	b. STRATEGY
	c. TEMPLATE METHOD
	d. BUILDER

14. ***"Defines an interface for creating objects, but let subclasses to decide which class to instantiate."*** **Qual o padrão de desenho que tem este propósito (*intent*)?**

	a. BUILDER
	b. ITERATOR
	c. OBSERVER
	d. FACTORY METHOD

15. O diagrama abaixo reflecte que padrão de desenho?



	a. BUILDER
	b. TEMPLATE METHOD
	c. COMPOSITE
	d. FACTORY METHOD

16. Durante o desenvolvimento do primeiro projecto guiado, para se proceder ao desenho do labirinto e respectivos elementos gráficos, criava-se uma subclasse de JPanel e fazia-se *override* ao método *paintComponent()*. Qual o padrão usado pelo SWING que permite este tipo de mecanismo?

	a. COMPOSITE
	b. TEMPLATE METHOD
	c. FACTORY METHOD
	d. OBSERVER

Refactoring e *Code Smells*

17. Quando <u>não</u> se deve fazer *refactoring* a uma parte específica de código?

	a. quando se está a adicionar uma nova funcionalidade
	b. quando se está a resolver um bug ou anomalia
	c. quando se está a analisar esse código e a tentar perceber como funciona
	d. quando se está a meio duma implementação e o código ainda não funciona

18. Que *refactoring* considera ser o mais útil aplicar para melhorar o seguinte código?

```
double getSpeed() {
  switch (_type) {
    case EUROPEAN:
      return getBaseSpeed();
    case AFRICAN:
      return getBaseSpeed() - getLoadFactor() * _numberOfCoconuts;
    case NORWEGIAN_BLUE:
      return (_isNailed) ? 0 : getBaseSpeed(_voltage);
  } throw new RuntimeException ("Should be unreachable");
}
```

	a. Extract Method
	b. Switch Statements
	c. Replace Conditional with Polymorphism
	d. Duplicated code

19. Que *code smells* podem ser resolvidos pelo *refactoring Extract Method*?

	a. Long Method e Switch Statements
	b. Duplicated Code e Comments
	c. Long Parameter List e Comments
	d. Large Class e Data Clumps

20. Que refactoring sugere utilizar para melhorar o seguinte código?

```
public int _type;
```

	a. Pull Up Field
	b. Extract Method
	c. Encapsulate Field
	d. Public Field