|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Logotipo, nome da empresa  Descrição gerada automaticamente | | **Curso: Análise e Desenvolvimento de Sistemas**  **Professor: Michel Gonçalves da Silva**  **Exercícios**  **Ano e Semestre Letivo – 2024/2** | | | | | | |
| **Nota da  Prova** |  | | **Nota dos Trabalhos** |  | **Nota  Final** |  | **Nota Revisada** |  |

|  |  |  |
| --- | --- | --- |
| **Nome do aluno:** | **RA:** | **Turma:** |
| **Assinatura do aluno:** | **Data do Exercício:** | |

**QUESTÕES**

**1** - Durante o desenvolvimento de um software utilizando a programação orientada a objetos (POO), uma equipe deseja criar um sistema no qual as entidades do mundo real sejam representadas como objetos. Qual conceito da POO permite a criação de classes que podem ser estendidas e modificadas por outras classes, promovendo a reutilização de código?

Alternativas

**A**

Encapsulamento.

**B**

Herança.

**C**

Polimorfismo.

**D**

Abstração.

**2** – Sobre os conceitos aplicados em Orientação a Objetos, escolha a opção correta:

**A**

Herança representa a capacidade de uma única referência invocar métodos diferentes, dependendo do seu encapsulamento.

**B**

Encapsulamento identifica o princípio de ocultar os detalhes internos de uma classe e expor apenas o que é necessário por meio de interfaces públicas.

**C**

Polimorfismo é o mecanismo pelo qual uma classe (subclasse) herda características e comportamentos de outra classe (superclasse).

**D**

Composição relaciona classes em uma hierarquia, em que uma subclasse herda atributos e métodos de uma superclasse e é utilizada quando há um relacionamento “é-um” (*is-a*).

**E**

Abstração é o mecanismo de ocultar o código fonte e comportamentos de outra classe (superclasse).

**3** - Tendo em vista que arquitetura e linguagem orientada a objetos (OO) são conceitos fundamentais no desenvolvimento de *software*, assinale a opção correta.

**A**

A arquitetura de *software* define a estrutura geral de um sistema, incluindo seus componentes e as interações entre eles, servindo como um projeto para construir e manter um sistema de informação, embora não haja garantia de que esse projeto atenda aos requisitos de desempenho e escalabilidade.

**B**

A manutenibilidade é um benefício da integração de arquitetura em camadas e OO, pois objetos podem ser reutilizados em diferentes partes do sistema ou mesmo em diferentes projetos.

**C**

Polimorfismo na programação OO permite que uma classe derive de outra, reutilizando e estendendo funcionalidades.

**D**

Linguagens orientadas a objetos são projetadas para modelar problemas do mundo real por meio de objetos; os objetos encapsulam dados e comportamentos, promovendo a reutilização e a modularidade do código.

**E**

Como exemplo da arquitetura de *software* em um contexto orientado a objetos, tem-se a arquitetura em camadas e OO, que conta com as camadas de: apresentação, lógica, acesso a dados, segurança, interoperabilidade.

**4**- Com relação aos conceitos da programação orientada a objetos, avalie as afirmativas a seguir.  
  
I. Herança e polimorfismo são pilares da programação orientada a objeto.  
II. Uma classe contém os atributos (dados ou propriedades) do objeto e as funções (métodos) que podem ser executadas por ele.  
III. Um objeto é uma instância de uma classe.  
Está correto o que se afirma em

**A**

I, apenas.

**B**

II, apenas.

**C**

I e II, apenas.

**D**

II e III, apenas.

**E**

I, II e III.

**5** – Qual das seguintes opções é uma vantagem da análise orientada a objetos sobre a análise estruturada?

Alternativas

**A**

Maior clareza na modelagem do fluxo de dados.

**B**

Melhor suporte para reuso de software.

**C**

Simplicidade na representação de processos.

**D**

Facilidade na geração de diagramas de fluxo de dados.

**6** –

Em programação orientada a objetos, o emprego do conceito de encapsulamento implica que

Alternativas

**A**

os dados declarados em uma subclasse podem ser acessados em sua superclasse.

**B**

a definição de uma interface deve ser separada de sua implementação.

**C**

a implementação de uma interface deve ser contida em uma única classe.

**D**

os dados de um objeto devem ser todos definidos em uma mesma classe.

**7** – Sobre os conceitos aplicados em Orientação a Objetos, escolha a opção correta:

**A**

polimorfismo é a capacidade de uma classe de possuir várias instâncias simultaneamente.

**B**

métodos protegidos de uma classe não podem ser acessados por classes-filhas.

**C**

as classes-filhas podem acessar os métodos privados de uma classe-pai.

**D**

a hierarquia composta por uma classe que define métodos e atributos utilizados por classes derivadas é denominada *herança*.

**E**

classes abstratas entregam a implementação padrão de comportamentos a classes derivadas, de modo que estas não possam reimplementá-los.

**8** - Dentro da programação orientada a objetos (POO), o conceito de herança desempenha um papel fundamental na estruturação de código. Assim, qual das seguintes alternativas melhor descreve o conceito de herança?

Alternativas

**A**

Herança refere-se à prática de duplicar código, promovendo a reutilização de métodos e atributos idênticos em diferentes classes.

**B**

Herança é a restrição da visibilidade de certos métodos e atributos a uma classe específica, garantindo encapsulamento e segurança no acesso aos membros da classe.

**C**

Herança consiste na criação de instâncias de uma classe em outra classe, promovendo a flexibilidade na implementação de métodos e facilitando a comunicação entre objetos.

**D**

Herança é a capacidade de uma classe adquirir propriedades e comportamentos de outra, permitindo a criação de uma hierarquia de classes e a reutilização de código.

9 - Na programação orientada a objetos, o conceito de encapsulamento desempenha um papel fundamental na organização do código. Entre as opções abaixo, qual melhor descreve a principal finalidade do encapsulamento, considerando a separação dos aspectos externos e internos de um objeto?

Alternativas

**A**

Facilitar a manutenção do código, tornando o programa altamente dependente de pequenas modificações, garantindo flexibilidade.

**B**

Expor os detalhes internos da implementação de um objeto, promovendo uma compreensão mais profunda e direta do programa.

**C**

Integrar os aspectos externos e internos de um objeto, simplificando a interação entre diferentes objetos e aumentando a complexidade do código.

**D**

Ocultar os detalhes internos de um objeto, separando-os dos aspectos externos, e impedir a propagação de grandes efeitos causados por pequenas modificações no programa, promovendo a robustez e a manutenibilidade do código.

**10** –

No que se refere à orientação a objetos, insere-se o conceito de métodos, sendo correto afirmar que estes

Alternativas

**A**

implementam, exclusivamente, a geração de mensagens ocasionadas por erros de execução.

**B**

implementam o comportamento (ações) oferecidas por uma classe.

**C**

representam o conjunto de dados e valores que definem a estrutura de dados de uma classe.

**D**

são utilizados exclusivamente em situações de exceção na implementação das classes.

**E**

já não são mais empregados na orientação a objetos.