

PARADIGMAS DE LINGUAGUENS

Questão 1: Qual recurso da programação orientada a objetos (POO), por meio da especificação de uma classe, e não a maneira como vai ser implementada (codificada), permite que classes diferentes utilizem métodos comuns em ambas as classes?

- A) interface
- B) private
- C) void
- D) polimorfismo
- E) super

A interface define apenas a especificação de uma classe, e não a maneira como ela será implementada (codificada). Ela funciona de maneira semelhante a classes abstratas, mas não permite a implementação de nenhum método, pois apenas declara um ou mais métodos (comportamento) que a classe que implementa a interface deve ter (FURGERI, 2015).

Questão 2: Qual fase do processo de compilação obtém as unidades léxicas do analisador léxico e as utiliza para construir estruturas hierárquicas chamadas de árvores de análise?

- A) Analisador léxico.
- B) Interpretação pura.
- C) Analisador sintático.
- D) Gerador de código intermediário.
- E) Analisador semântico.



Questão 3: Analise as seguintes afirmativas sobre o paradigma de programação orientada a objetos e assinale a alternativa correta:

- I. A classe abstrata não permite ser instanciada, ou seja, não podem ser criados objetos a partir dela. Os seus métodos são apenas declarados e não são definidos, ou seja, não têm implementação.
- II. Uma interface encapsula uma coleção de constantes e assinaturas de métodos abstratos. Uma interface não pode incluir variáveis, construtores ou métodos não abstratos.
- III. Os objetos de uma classe se utilizam de conceitos como módulos, procedimentos, funções e variáveis para representar e resolver problemas do mundo real.

- A) I, II, III.
- B) II e III, apenas.
- C) I, apenas.
- D) I e III, apenas.
- E) I e II, apenas.

Questão 4: Todo programa ou código-fonte, independentemente de em qual linguagem de programação seja desenvolvido, antes de ser executado, deve ser convertido para a linguagem de máquina.

Qual o método em que um programa-fonte escrito em uma linguagem de alto nível é traduzido para código executável, em uma versão compatível com a linguagem de máquina, a qual pode ser executada diretamente no computador?

- A) Compilação.
- B) Interpretação pura.
- C) Interação.
- D) Compilação pura.
- E) Sistema híbrido.

Compilação ou tradução:

- Na compilação ou tradução, a conversão do programa-fonte escrito em uma linguagem de alto nível é traduzida para o código executável, em uma versão compatível com a linguagem de máquina, a qual pode ser executada, diretamente, no computador;

Questão 5: No processo de compilação, ele cria um programa em uma linguagem diferente, em um nível intermediário entre o programa fonte e a saída final do compilador. Estamos nos referindo a:

- A) Analisador léxico.
- B) Interpretação pura.
- C) Analisador sintático.
- D) Gerador de código intermediário.
- E) Analisador semântico.

contém, a partir da sequência de *tokens*, se estes compõem as estruturas sintáticas, como os comandos e as expressões;

- Gerador de código intermediário e analisador semântico:** o gerador de código intermediário cria um programa em uma linguagem diferente, em um nível intermediário entre o programa-fonte e a saída final do compilador;

Questão 6: No processo de compilação, ele é responsável por verificar erros difíceis de serem detectados durante a análise sintática, como erros de tipos em comandos e expressões. Analisa se as estruturas sintáticas fazem sentido como escopo e uso dos nomes e correspondência entre declarações e uso de variáveis e se há relação entre operadores e os operandos.

- A) Analisador léxico.
- B) Interpretação pura.
- C) Analisador sintático.
- D) Gerador de código intermediário.
- E) Analisador semântico.**

Questão 7: Qual conceito expressa a capacidade que objetos de diferentes classes relacionadas por implementação de interface ou herança têm de responderem de formas diferentes a uma mesma chamada de método?

- A) interface
- B) private
- C) void
- D) polimorfismo**
- E) super

Questão 8: Qual classe não pode acessar ou herdar os atributos e métodos declarados como private de sua classe pai ou superclasse?

- A) classe filha ou subclasse**
- B) private
- C) void
- D) protected
- E) super

A classe filha ou subclasse não pode acessar ou herdar os atributos e métodos declarados como private de sua classe pai ou superclasse. Portanto, a alternativa correta é:

A) classe filha ou subclasse.

O modificador de acesso **'private'** é o mais restritivo de todos, restringindo o acesso aos membros da classe apenas aos métodos dessa mesma classe. Isso significa que nenhum membro da classe filha ou subclasse pode acessar diretamente os membros **'private'** da classe pai ou superclasse. Para que uma classe filha ou subclasse possa acessar esses membros, é necessário utilizar os modificadores de acesso **'protected'** ou **'public'**.

Questão 9: Com relação a subprogramas, analise as seguintes afirmativas e assinale a alternativa correta:

- I.** Subprograma é uma forma que as LPs utilizam para implementar o conceito de dividir para conquistar, com a intenção de facilitar a resolução de problemas computacionais, melhorar a compreensão do programa e viabilizar o reúso de código.
- II.** Subprograma permite dividir o sistema em módulos aonde os dados e os procedimentos são encapsulados. Para realizar essa tarefa, utiliza-se o conceito de abstração, onde separa-se o que de fato importa no contexto do problema.
- III.** Subprogramas são representados nas LPs por abstrações de procedimentos, estes realizam as ações necessárias quando são chamados a partir do programa principal.

- A) I, II, III.**
- B) II e III, apenas.
- C) I, apenas.
- D) I e III, apenas.
- E) I e II, apenas.

O uso de subprogramas permite que um sistema seja dividido em módulos ou componentes menores e mais gerenciáveis, em que dados e procedimentos são encapsulados. Essa técnica ajuda a tornar o sistema mais organizado e modular, facilitando a manutenção e a evolução do código.

Além disso, a abstração é um conceito-chave na programação orientada a objetos, que se baseia no uso de objetos para representar entidades do mundo real e seus comportamentos. A abstração envolve separar o que é relevante e importante para o problema em questão, e deixar de lado o que não é importante.

Questão 10: Qual tipo de classe declara atributos e comportamentos comuns de várias classes em uma hierarquia de classes?

- A) construtor
- B) private
- C) abstrata**
- D) protected
- E) super

O tipo de classe que declara atributos e comportamentos comuns de várias classes em uma hierarquia de classes é a classe abstrata (alternativa C).

Uma classe abstrata é uma classe que não pode ser instanciada, e serve como um modelo para outras classes. Ela pode definir atributos e métodos, mas não precisa implementá-los. As classes que herdam da classe abstrata devem implementar seus métodos abstratos.

Usar uma classe abstrata é uma forma de estabelecer um contrato com as classes que a herdam, garantindo que elas possuem certos atributos e comportamentos em comum. Isso pode ajudar a evitar repetições de código e facilitar a manutenção do sistema.

Questão 11: É considerado um padrão de solução de problemas relacionados a determinada categoria de programas e linguagens de programação. Trata-se de:

- A) Conceito de desenvolvimento de sistemas.
- B) Técnicas de programação.
- C) Paradigma de programação.**
- D) Paradigma de conhecimento
- E) Conceito de conhecimento.

Questão 12: Durante o processo de compilação, essa tabela serve como uma base de dados onde estão armazenadas informações de tipo e atributos de cada um dos nomes definidos pelo usuário no programa.

- A) Tabela de dados.
- B) Tabela de símbolos.**
- C) Tabela sintática.
- D) Tabela intermediária.
- E) Tabela semântica.

Processo de desenvolvimento de programas

- Para o processo de compilação, a tabela de símbolos serve como uma base de dados onde estão armazenadas as informações de tipo e os atributos de cada um dos nomes definidos pelo usuário no programa.
- Essa informação é inserida na tabela pelos analisadores sintático e léxico, e usada pelo analisador semântico e pelo gerador de código.

Questão 13: O processo de compilação é dividido em várias fases: análise léxica, análise sintática e análise semântica. Avalie as afirmativas a seguir sobre essas fases e assinale a alternativa correta.

I - A análise léxica agrupa os caracteres do programa fonte em unidades léxicas ou tokens, que são identificadores, símbolos, operadores e palavras especiais, ignorando comentários no programa fonte.

II - A análise sintática obtém as unidades léxicas do analisador léxico e as utiliza para construir estruturas hierárquicas chamadas de árvores de análise. Esse analisador aplica regras gramaticais da linguagem sintática e confere a sequência de tokens se estes compõem estruturas sintáticas como comandos e expressões.

III - A análise semântica analisa se as estruturas sintáticas fazem sentido como escopo e uso dos nomes correspondência entre declarações e uso de variáveis e se há relação entre operadores e os operandos.

- A) I, II e III**
- B) II e III
- C) I, apenas
- D) I e III, apenas
- E) I e II, apenas

Questão 14: Qual comando ou modificador de acesso define que um método ou atributo declarado como público pode ser livremente utilizado por objetos de qualquer outra classe?

- A) public**
- B) private
- C) void
- D) protected
- E) super

Questão 15: Avalie as seguintes afirmativas sobre os conceitos básicos de programação de computadores e assinale a alternativa correta:

I - Os tipos de operadores podem ser unários (1 operando), binários (2 operandos) ou ternários (3 operandos).

II - Existem diversos tipos de expressões, elas são classificadas de acordo com os seus tipos de operandos, a natureza da operação realizada e o tipo de resultado produzido. Por exemplo, os operadores aritméticos mais simples em Java são: soma (+), subtração (-), divisão (/) e multiplicação (*).

III - As expressões relacionais têm como propósito comparar ou relacionar os valores de seus operandos geralmente dois operandos e um operador relacional. O resultado dessa expressão é booleano: true (um) ou false (zero).

- A) I, II e III**
- B) II e III
- C) I, apenas
- D) I e III, apenas
- E) I e II, apenas

Questão 16: Relacione os paradigmas de linguagem às suas respectivas características ou descrições. Depois assinale a sequência numérica correta.

- | | |
|------------------------------|------------------------------------------------------------------------------|
| (1) Programação estruturada. | () Possui hierarquia de classes e trabalha com coleções de objetos. |
| (2) Orientada a objetos. | () Sua programação é estruturada através de eventos. |
| (3) Orientada a eventos | () Características centrais de sua estrutura: sequência decisão e repetição |

A) 123

B) 312

C) 321

D) 213

E) 231

Questão 17: Qual a propriedade esperada para uma linguagem de programação (LP) que enfatiza a facilidade com que um código-fonte pode ser lido e entendido?

A) Legibilidade.

B) Simplicidade.

C) Redigibilidade.

D) Ortogonalidade

E) Portabilidade.

Estilo e qualidade de programas

Quais as propriedades desejáveis de uma Linguagem de Programação?

- **Legibilidade:** é a facilidade com que um código-fonte pode ser lido e entendido;
- Programas difíceis de ler são, também, difíceis de escrever e modificar, prejudicando a confiabilidade na LP, tanto nas fases de desenvolvimento quanto nas de manutenção do ciclo de vida (SEBESTA, 2011);
- Quanto mais a linguagem de uma LP for natural, a sua leitura se tornará menos complicada e mais fácil será o seu entendimento.

Questão 18: Analise as seguintes afirmativas relacionadas a conceitos básicos sobre programação e assinale a alternativa correta:

I. Procedimentos executam um conjunto de instruções e retornam valor para o programa principal ou unidade chamadora.

II. Em cada linguagem de programação os subprogramas ou módulos são implementados de maneira diferentes. Na linguagem Pascal, a modularização é implementada por meio dos procedimentos e das funções.

III. A utilização de módulos ou subprogramas permitem uma melhor organização do código-fonte, pois cada módulo é responsável por resolver uma determinada função dentro do software.

A) I, II, III.

B) II e III, apenas.

C) I, apenas.

D) I e III, apenas.

E) I e II, apenas.

Questão 19: O paradigma de programação orientado a eventos é caracterizado por:

I. O usuário, para interagir com essas interfaces, faz uso de toques em botões, digita textos e seleciona diversas opções nos menus da aplicação. ✓

II. É um modelo de desenvolvimento de software onde partes do programa são executadas em momentos inesperados e não se controla a sequência na qual ocorrem os eventos de entrada de dados. ✓

III. Não permite que o usuário realize operações durante a execução do programa.

A) I, II e III

B) II e III, apenas.

C) I, apenas

D) I e III, apenas.

E) I e II, apenas.

Questão 20: Com relação aos conceitos de POO, qual comando permite chamar formalmente o construtor da superclasse quando um programa cria objetos da classe filha ou subclasse?

A) Protected

B) Private

C) Void

- D) Final
E) Super

Questão 21: É um recurso da programação estruturada que permite escrever trechos de códigos da linguagem que retorna um resultado em um tipo de dado determinado pelo programador.

- A) Funções**
B) Scriptlets
C) Laços
D) Comentários
E) Procedimento

Função:

- A diferença entre a função e o procedimento está no fato de que uma função sempre deve retornar um resultado para a unidade chamadora;
- As funções são constituídas por uma sequência de instruções para executar uma tarefa específica, porém, no final, retornam um valor ou resultado para o programa principal;
- Por exemplo, uma função pode fazer um cálculo e retornar o seu resultado.

Questão 22: Com relação as linguagens de programação, qual afirmação esta **incorreta**?

- A) Na LP Pascal os operadores lógicos têm maior precedência sobre os relacionais. Para alterar a ordem de precedência deve se usar parênteses.
- B) Os operadores relacionais verificam se o valor ou o resultado da expressão logica à esquerda e igual ou diferente da direita retornando um valor booleano (true/false)
- C) Nos projetos de software, a interface e muito útil para que os desenvolvedores sejam obrigados a seguir um padrão de projeto como a de nomenclatura de métodos, pois as classes são obrigadas a implementar os métodos definidos na interface.
- D) Os mecanismos de herança não permitem o compartilhamento ou reaproveitamento de métodos definidos nas classes**
- E) Tipos de dados são uma coleção de valores de dados um conjunto de operações que podem ser realizadas sobre eles

Questão 23: O que determina um local ou contexto em que uma variável pode ser utilizada ou referenciada como um identificador em um programa?

- A) Variável fracamente tipada.
B) Escopo de visibilidade.
C) Variável fortemente tipada.
D) Variável dinamicamente tipada.
E) Variável fracamente tipada.

Estilo e qualidade de programas

Definição do escopo de visibilidade das variáveis:

- Escopo é a característica que determina um local onde uma variável pode ser utilizada ou referenciada como um identificador em um programa;

Questão 24: O desenvolvedor de software ao programar um computador codifica uma série de instruções. Qual paradigma permite que essas instruções sejam executadas de forma sequencial, condicional e repetitiva?

- A) Paradigma funcional.
B) Paradigma estruturado.
C) Paradigma lógico.
D) Paradigma declarativo.
E) Paradigma orientado a objetos.

Paradigma de Programação Estruturada

Programação estruturada é uma forma de programação de computadores que preconiza que todos os programas possíveis podem ser reduzidos a apenas três estruturas:

- Sequência (estrutura de controle Sequencial);
- Decisão (estrutura de controle condicional);
- Iteração ou repetição (estrutura de controle repetitiva).

Questão 25: Uma LP deve propiciar ao desenvolvedor a facilidade de se concentrar nos algoritmos centrais de programa para resolução do problema de negócio e não se preocupar com aspectos não relevantes como detalhes de implementação. Estamos definindo a propriedade de uma LP:

- A) Legibilidade
B) Simplicidade.
C) Redigibilidade.
D) Ortogonalidade.
E) Portabilidade.

Questão 26: Qual comando ou modificador de acesso define que, com o uso deste modificador propriedades e métodos declarados, só são acessíveis por métodos da classe que foram declarados?

- A) Interface
- B) Private**
- C) Void
- D) Polimorfismo
- E) Super

Questão 27: Influenciada pelo grau de padronização da LP, é a capacidade de um programa se comportar de maneira similar, independentemente da arquitetura computacional, hardware ou sistema operacional sobre o qual estão sendo executados. Estamos definindo:

- A) Legibilidade:
- B) Simplicidade.
- C) Redigibilidade
- D) Ortogonalidade.
- E) Portabilidade.**

Estamos definindo Portabilidade.

A portabilidade é a capacidade de um programa se comportar de maneira similar em diferentes sistemas computacionais, independentemente da arquitetura de hardware, sistema operacional ou ambiente em que estão sendo executados. Um programa portátil pode ser facilmente transferido de um sistema para outro, sem a necessidade de reescrever o código ou fazer grandes modificações.

O grau de portabilidade de um programa é influenciado pelo grau de padronização da linguagem de programação utilizada, bem como pela maneira como o programa foi escrito e compilado. Programas que utilizam apenas recursos padronizados da linguagem de

Questão 28: São programas de computador que utilizam a CPU para traduzir um programa-fonte, um comando de cada vez, de uma linguagem de programação, geralmente de alto nível e executa diretamente o programa. Estamos nos referindo ao método de conversão:

- A) Compilação.
- B) Interpretação pura.**
- C) Interação.
- D) Compilação pura.
- E) Sistema híbrido.

Processo de desenvolvimento de programas

Interpretação:

- Esse processo de conversão do código-fonte utiliza a CPU para traduzir um programa-fonte, de uma LP interpretada, geralmente, de alto nível e executa, diretamente, o programa;
- Esse processo de *software* fornece uma máquina virtual para a linguagem (SEBESTA, 2011).

Questão 29: Qual a propriedade esperada para uma linguagem de programação (LP) que minimiza a relação de dependência entre componentes, garantindo que, ao se alterar um componente, essa alteração não afetara outros componentes, trazendo mais confiabilidade para o desenvolvedor?

- A) Legibilidade
- B) Simplicidade
- C) Redigibilidade
- D) Ortogonalidade**
- E) Portabilidade

- **Ortogonalidade:** é quando não existe uma relação de dependência entre componentes. Se um componente for alterado, essa alteração não vai afetar outros. A LP deve permitir a combinação de conceitos básicos sem que se produzam efeitos atípicos, o ideal é que ela tenha muita ortogonalidade. A falta de ortogonalidade gera exceções às regras da linguagem, fato que gera falta de confiança no desenvolvedor. Por exemplo, um caso clássico de falta de ortogonalidade na

Questão 30: Utilizando (V) para verdadeiro e (F) para falso, assinale quais das alternativas abaixo são propriedades esperadas para uma LP. Em seguida indique qual a alternativa contém a sequência correta.

- () Legibilidade.
- () Arquitetura de sistema.
- () Metodologias de desenvolvimento de software.
- () Portabilidade.
- () Simplicidade.
- () Informações do sistema.
- () Redigibilidade.
- () Ortogonalidade.

- A) V-V-F-V-V-F-V-V.
- B) V-F-F-V-V-F-V-V.**
- C) V-F-V-V-V-F-V-V.
- D) V-F-F-V-V-F-F-V.
- E) V-F-F-V-V-V-F-V.

As propriedades esperadas para uma LP são:

- Legibilidade: a linguagem deve ser fácil de ler e entender;
- Portabilidade: o código escrito em uma plataforma deve ser capaz de ser executado em outras plataformas sem muitas modificações;
- Simplicidade: a linguagem deve ser fácil de usar e entender;
- Redigibilidade: a linguagem deve ser fácil de escrever e manter;
- Ortogonalidade: a linguagem deve possuir poucas regras que se apliquem a muitas situações diferentes.

Questão 31: As linguagens de programação fazem uso de identificadores e símbolos. Relacione os exemplos de lexemas a seguir (associe a coluna da direita à da esquerda) e depois escolha a sequência numérica correta:

- | | |
|------------------------|------------------------------------------------|
| 1- Operadores | () nome de procedimentos, funções e variáveis |
| 2- Identificadores | () símbolos =, +, <= |
| 3- Palavras reservadas | () if, else, int |

- 2- Identificadores () símbolos =, +, <=
- 3- Palavras reservadas () if, else, int

- A) 123
- B) 231
- C) 321
- D) 213**
- E) 312

- **Operação:** soma, subtração, multiplicação e divisão.
- **Operador:** pode ser relacional =, <=, >= e <>; ou lógico (or e not, and).
- **Palavras reservadas da linguagem e seus tipos de dados:** if, else, int, boolean, char.

Questão 32: As linguagens de programação imperativas utilizam os seguintes comandos:

- A) Variáveis, atribuições e laços de iteração.
- B) Funções, atribuições e laços de iteração.
- C) Atribuição, seleção e repetição.**
- D) Funções, definições e laços de iteração.
- E) Variável, atribuição e definição.

Como vimos anteriormente, as linguagens **imperativas** simulam o funcionamento da máquina de Von Neuman, que é o fundamento da arquitetura dos computadores, e executam comandos de atribuição de valores a variáveis e estruturas que controlam o fluxo de execução do programa. A linguagem imperativa utiliza os seguintes comandos:

- **Atribuição:** possibilita que os valores de variáveis sejam atualizados.
- **Seleção:** permite a escolha de diferentes caminhos no fluxo de controle do programa.
- **Repetição:** proporciona que determinados trechos de código possam ser executados repetidamente.

Questão 33: Qual método é utilizado para inicializar um objeto de uma classe no momento da sua criação?

- A) Construtor**
- B) Private
- C) Void
- D) Protected
- E) Super

Questão 34: Qual comando ou modificador de acesso define que um membro de uma classe pai pode ser acessado por outro membro da mesma classe, por membros de suas classes filhas e por membros de outras classes que estejam dentro do mesmo pacote?

- A) Interface
- B) Private
- C) Void
- D) Protected**
- E) Super

Questão :

Questão :

Questão :

Questão :

Questão :