ACUMULADO PARADIGMAS DA PROGRAMAÇÃO

Questão 1: 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

O método utilizado para inicializar um objeto de uma classe no momento da sua criação é construtor. A resposta correta é a opção A.

Um construtor é um método especial que tem o mesmo nome da classe e é invocado automaticamente quando um objeto é criado. Ele é usado para inicializar os atributos da classe e pode receber parâmetros para inicialização personalizada. Os construtores poder ter diferentes assinaturas, permitindo a criação de múltiplos construtores para uma classe

A opção B (private) e a opção D (protected) são modificadores de acesso que definem o nível de visibilidade dos membros da classe. A opção C (void) é um tipo de retorno de um método que indica que ele não retorna nenhum valor. A opção E (super) é uma palavra-chave usada para chamar o construtor da classe pai a partir de uma classe filha.

Questão 2: Com relação a passagem de parâmetros, assinale a alternativa que apresenta as três formas de passagem de parâmetros para um subprograma

A) Por resultado, por inferência e por valor.

B) Por valor, por referência e por resultado.

C) Por meio da memória de variáveis e pelo disco.

D) Por meio da memória, por transferência e por disco.

E) Por variáveis fixas, propriedades e por referência.

A alternativa correta é a B) Por valor, por referência e por resultado. As três formas de passagem de parâmetros mais comuns em linguagens de programação são:

- Passagem por valor (por valor): o valor do parâmetro é copiado para a variável local do subprograma e, assim, qualquer alteração feita no parâmetro dentro do subprograma não afetará a variável original que foi passada como argumento.
- Passagem por referência (por referência): a referência ou endereço do parâmetro é passada para o subprograma, permitindo que o subprograma acesse e modifique o valor original da variável fora do subprograma.
- Passagem por resultado (por resultado): é semelhante à passagem por referência, mas a variável passada como argumento é inicializada com um valor padrão antes de ser passada e, ao final do subprograma, seu valor é copiado de volta para a variável original.

Questão 3: 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.

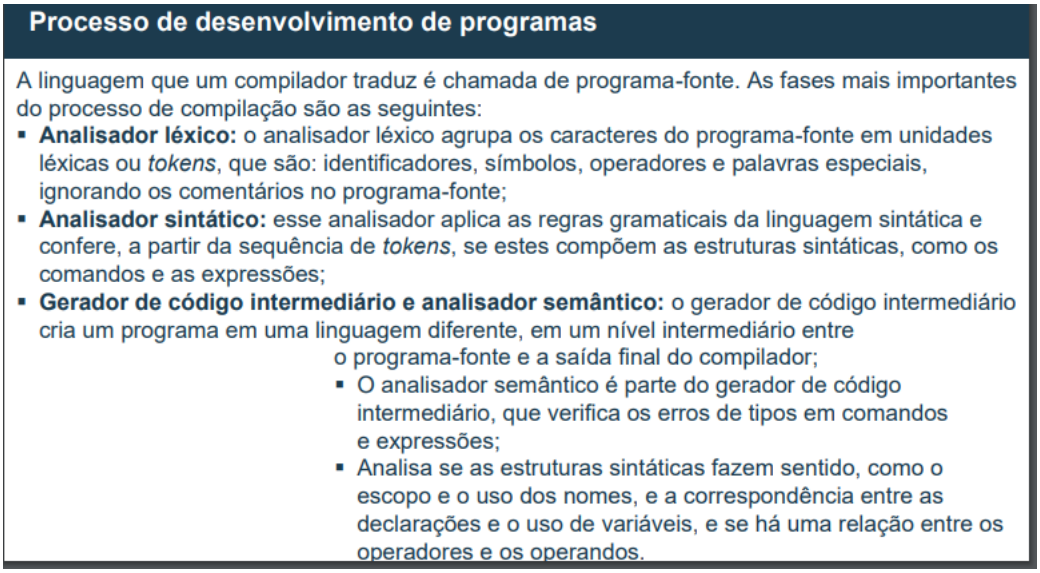
A) I, II e III

B) II e III

C) I, apenas

D) I e III, apenas

E) I e II, apenas



Processo de desenvolvimento de programas

A linguagem que um compilador traduz é chamada de programa-fonte. As fases mais importantes do processo de compilação são as seguintes:

- **Analisador léxico:** o analisador léxico agrupa os caracteres do programa-fonte em unidades léxicas ou *tokens*, que são: identificadores, símbolos, operadores e palavras especiais, ignorando os comentários no programa-fonte;
- **Analisador sintático:** esse analisador aplica as regras gramaticais da linguagem sintática e confere, 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;
  - O analisador semântico é parte do gerador de código intermediário, que verifica os erros de tipos em comandos e expressões;
  - Analisa se as estruturas sintáticas fazem sentido, como o escopo e o uso dos nomes, e a correspondência entre as declarações e o uso de variáveis, e se há uma relação entre os operadores e os operandos.

Questão 4: 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 5: 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 6: 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 7: 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 8: 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 ambas as classes?

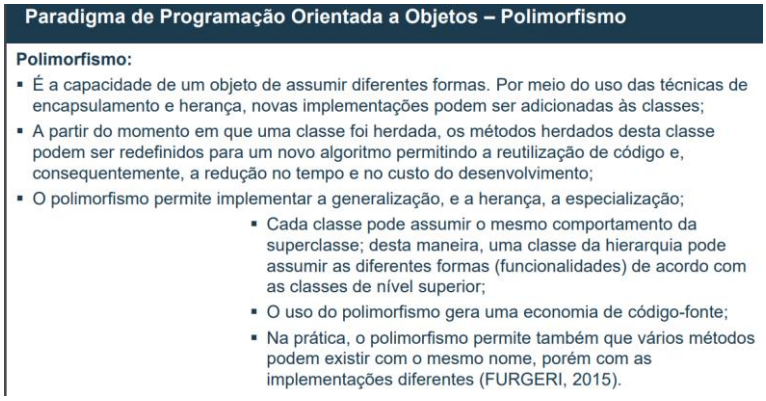
A) interface

B) private

C) void

D) polimorfismo

E) super



Questão 9: 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 10: Qual fase do processo de compilação agrupa os caracteres do programa fonte em unidades léxicas ou tokens, que são identificadores, símbolos, operadores e palavras especiais?

A) Analisador léxico

B) Interpretação pura

C) Analisador sintático

D) Gerador de código intermediário

E) Analisador semântico

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: 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 e III

B) II e III

C) I, apenas

D) I e III, apenas.

E) I e II, apenas.

Classe abstrata:

- A classe abstrata não permite ser instanciada, ou seja, não podem ser criados os objetos a partir dela. Os seus métodos são apenas declarados e não são definidos, ou seja, não têm implementação;

Interface

De acordo com Tucker e Noonan (2009, p. 582), "uma interface encapsula uma coleção de constantes e assinaturas de métodos abstratos. Uma interface não pode incluir variáveis, construtores ou métodos não abstratos".

Os objetos de uma classe são criados para representar entidades do mundo real ou abstrações que podem ser modeladas em um programa de computador. Para representar e resolver problemas do mundo real, os objetos de uma classe podem se utilizar de vários conceitos, como módulos, procedimentos, funções e variáveis, a fim de realizar tarefas específicas.

Questão 14: 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 hibrido.

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 15: 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 16: Com relação aos conceitos de paradigmas de programação e seus tipos, assinale a opção correta.

A) Em um programa orientado a objetos, toda a estrutura do programa é organizada em classes, que é uma declaração de tipo que encapsula constantes, variáveis e funções para manipulação dessas variáveis.

B) Em programação estruturada, as funções executam rotinas que não retornam resultados, enquanto os procedimentos executam algoritmos e retornam um resultado para a unidade chamadora.

C) Em um programa orientado a objetos, a sua estrutura se caracteriza pelo uso de funções e procedimentos e os seus dados são armazenados e acessados em variáveis fixas.

D) Na programação orientada a objetos, os programas são compostos por funções que implementam sentenças matemáticas da lógica de primeira ordem.

E) A programação orientada a eventos se caracteriza pelo uso excessivo de desvios incondicionais por meio do comando GOTO.

Questão 17: 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?

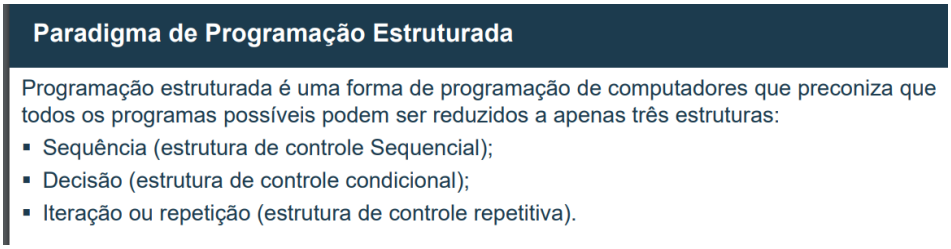
A) Paradigma funcional.

B) Paradigma estruturado.

C) Paradigma lógico.

D) Paradigma declarativo.

E) Paradigma orientado a objetos.



Questão 18: Especifica uma área de memória, associada a um nome, que pode armazenar valores de determinado tipo de dado. Essa descrição caracteriza uma:

A) Constante.

B) Instrução de atribuição.

C) Informação de comentário.

D) Variável

E) Instrução de controle

Questão 19: Qual a propriedade esperada para uma linguagem de programação (LP) que enfatiza a facilidade com que um código-fonte pode ser lido e entendido?

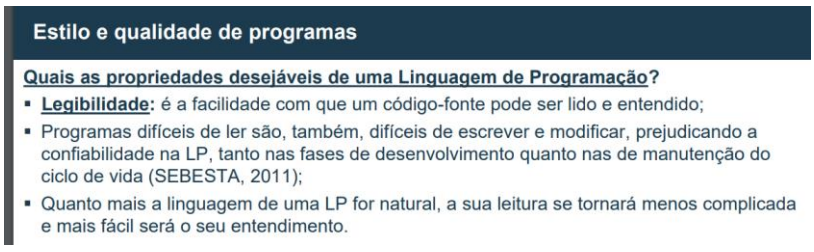
A) Legibilidade.

B) Simplicidade.

C) Redigibilidade.

D) Ortogonalidade

E) Portabilidade.



Questão 20: 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 21: Qual comando ou modificador de acesso define que um membro de uma classe pai pode ser acesado por membros 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

O modificador de acesso que define que um membro de uma classe pai pode ser acessado por membros da mesma classe, por membros de suas classes filhas e por membros de outras classes que estejam dentro do mesmo pacote é o modificador de acesso "protected".

Quando um membro de uma classe é declarado como protegido, ele pode ser acessado pelos membros da classe em que foi declarado, pelos membros de quaisquer classes que herdam essa classe e pelos membros de outras classes no mesmo pacote. No entanto, ele não pode ser acessado pelos membros de classes que não estão no mesmo pacote e que não herdam da classe em que o membro protegido foi declarado.

Questão 22: Um meio termo entre os compiladores e os interpretadores puros, ele realiza a tradução de linguagens de alto nível para uma linguagem intermediária projetada para facilitar a interpretação. 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 hibrido.

Processo de desenvolvimento de programas

Interpretação híbrida:

- A interpretação híbrida é um meio termo entre os compiladores e os interpretadores puros; ela realiza a tradução de linguagens de alto nível para uma linguagem intermediária projetada para facilitar a interpretação;
- Esses sistemas são mais rápidos do que a interpretação pura, porque as sentenças da linguagem-fonte são decodificadas apenas uma vez, e o programa interpreta o código intermediário, ao invés de traduzir o código da linguagem intermediária para a linguagem de máquina (SEBESTA, 2011).

Questão 23: 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.

Procedimento:

- Um procedimento é um subprograma que executa determinada tarefa e não retorna nenhuma informação para a 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.

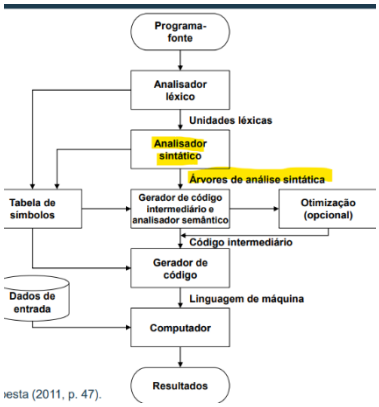
D) I e III, apenas.

E) I e II, apenas.

Na linguagem de programação Pascal, a modularização é implementada por meio dos procedimentos e das funções. Procedimentos são blocos de código que realizam uma tarefa específica e não retornam valores, enquanto funções são blocos de código que também realizam uma tarefa específica, mas retornam um valor para o chamador.

A utilização de módulos ou subprogramas permite uma melhor organização do código-fonte, pois cada módulo é responsável por resolver uma determinada função dentro do software. Esses módulos são geralmente criados para realizar tarefas específicas e podem ser reutilizados em diferentes partes do programa, o que ajuda a reduzir a redundância e aumentar a eficiência do código.

Questão 24: 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?



esta (2011, p. 47).

A) Analisador léxico.

B) Interpretação pura.

C) Analisador sintático.

D) Gerador de código intermediário.

E) Analisador semântico.

Questão 25: 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.

Al 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 programação e evitam depender de recursos específicos do sistema operacional ou da arquitetura de hardware tendem a ser mais portáteis.

Questão 26: As linguagens de programação fazem uso de identificadores e simbolos. Relacione os exemplos de lexemas a seguir (associe a coluna da direita à da esquerda) e depois escolha a sequência numérica correta:

1 – Operadores	( ) nomes de procedimentos, funções e variáveis
2 – Identificadores	( ) simbolos =, +, <=
3 – Palavras reservadas	( ) if, else, int

A) 123
B) 231

C) 321
D) 213
E) 312

Questão 27: 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 28: 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 29: 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:

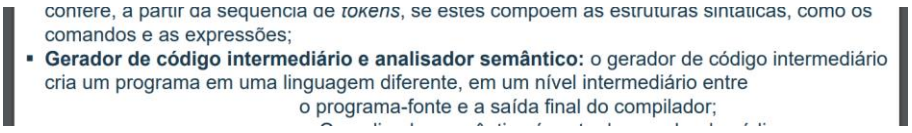
A) Analisador léxico.

B) Interpretação pura.

C) Analisador sintático.

D) Gerador de código intermediário.

E) Analisador semântico.


confere, a partir da sequência de *tokens*, se estes compoem as estruturas sintaticas, 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 30: 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

A ortogonalidade é uma propriedade de uma linguagem de programação que permite que seus recursos sejam usados de maneira consistente e previsível em todas as situações possíveis. Isso significa que cada recurso da linguagem deve ser independente dos outros recursos, o que permite que um programador combine esses recursos de maneiras diferentes para alcançar diferentes objetivos.

Questão 31: 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 hibrido.

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

A) Variaveis, 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.

Questão 33: 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 reuso de código.

II. Subprograma permite dividir o sistema em módulos aonde os dados e os procedimentos são encapsulados. Para utilizar 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 e III

B) II e III, apenas.

C) I, apenas.

D) I e III, apenas,

E) I e II, apenas.

À medida que os problemas se tornam maiores e mais complexos, sempre é possível simplificar a solução dividindo o programa em partes menores, seguindo o lema dividir para conquistar, chamadas de subprogramas ou módulos baseados em procedimentos e funções.

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

( ) Legibilidade.

( ) Arquitetura de sistema.

( ) Metodologias de desenvolvimento de software.

( ) Portabilidade.

( ) Simplicidade.

( ) Informações do sistema.

( ) Redigibilidade.

( ) Ortogonalidade.

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.

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.