

PERGUNTAS

Sobre Compiladores:

1. Um compilador é um programa de computador (ou um grupo de programas) que, a partir de um código-fonte, escrito em uma linguagem máquina, cria um programa semanticamente equivalente, porém escrito em outra linguagem de alto nível, código objeto. Fonte: “Compilador” (2022).
a) Errado ☒ (X)
b) Certo ☐ ()
2. Classicamente, um compilador traduz um programa de uma linguagem textual facilmente entendida por um ser humano para uma linguagem de máquina, específica para um processador e sistema operacional. Fonte: “Compilador” (2022).
a) Errado ☐ ()
b) Certo ☒ (X)
3. Um compilador traduz um programa de uma linguagem de máquina para uma linguagem textual facilmente entendida por um ser humano. Fonte: “Compilador” (2022).
a) Errado ☒ (X)
b) Certo ☐ ()
4. Com relação a linguagens de programação e compiladores, julgue os itens subsequentes. Um algoritmo computacional escrito em linguagem de programação pode ser completamente executado sem gerar nenhuma saída. Fonte: Concursos (2022)..
a) Errado ☒ (X)
b) Certo ☐ ()
5. Linguagem compilada é uma linguagem de programação em que o código-fonte, nessa linguagem, é executado diretamente pelo sistema operacional ou pelo processador, após ser traduzido através de um processo chamado compilação, usando um programa de computador chamado compilador, para uma linguagem de baixo nível, como linguagem de montagem “Assembly” ou código de máquina. Fonte: “Linguagem compilada” (2022).
a) Errado ☐ ()
b) Certo ☒ (X)
6. Linguagem interpretada é uma linguagem de programação em que o código-fonte, nessa linguagem, é executado diretamente pelo sistema operacional ou pelo processador, após passar pelo interpretador é gerado um arquivo executável. Fonte: “Linguagem interpretada” (2022).
a) Errado ☒ (X)
b) Certo ☐ ()
7. Linguagem interpretada é uma linguagem de programação em que o código-fonte nessa linguagem é executado por um programa de computador

chamado interpretador, executado em seguida pelo sistema operacional ou processador. Exemplo de linguagens interpretadas (ASP, JavaScript, PHP, Python). Fonte: "Linguagem interpretada" (2022).

- a) Errado ☐
- b) Certo ☒

8. Linguagem compilada gera um arquivo executável, que pode ser executado diretamente pelo sistema operacional. Existem linguagens que não geram código executável, como as linguagens interpretadas. Exemplo de linguagens compiladas (C, C++, COBOL, Delphi, Visual Basic). Fonte: "Linguagem compilada" (2022).

- a) Errado ☐
- b) Certo ☒

9. Acerca das características das principais linguagens de programação, julgue o item seguinte.

Linguagem de programação de alto nível consiste em um conjunto de símbolos, palavras e regras próximo da linguagem humana, e que é fácil de escrever, depurar e manter. Fonte: "Questões de Concursos" (2022).

- a) Errado ☐
- b) Certo ☒

10. Em programação de computadores, uma palavra reservada é uma palavra que, em algumas linguagens de programação, não pode ser utilizada como um identificador por ser reservada para uso da gramática da linguagem. Por exemplo, na linguagem de programação Java, um programador não pode utilizar uma variável com o nome (**For**), pois esta palavra é reservada para construção de loops. Ela é uma "palavra-chave", e por ser de uso restrito, é também uma "palavra reservada". (Em algumas linguagens, como C ou Java, os termos têm o mesmo significado). Fonte: "Palavra reservada" (2022).

- a) Errado ☐
- b) Certo ☒

11. *Case-sensitive* é um anglicismo que se refere a uma categoria de análise tipográfica da informática. Em língua portuguesa, significa algo como "sensível à caixa das letras" ou "sensível a maiúsculas e minúsculas". Diz-se que um software é *case-sensitive* ou possui "*case-sensitivity*" quando ele consegue analisar uma cadeia de caracteres, avaliar a existência de caixa alta e caixa baixa e comportar-se de diferentes maneiras em função disso. Fonte: "Case-sensitive" (2022).

- a) Errado ☐
- b) Certo ☒

12. Diz-se que um software não é *case-sensitive* ou não possui "*case-sensitivity*" quando ele não consegue analisar uma cadeia de caracteres, e avaliar a existência de caixa alta e caixa baixa e comportar-se de diferentes maneiras em função disso. Fonte: "Case-sensitive" (2022).

- a) Errado ☒
- b) Certo ☐

Sobre Bibliotecas

13. Na ciência da computação, biblioteca é uma coleção de subprogramas utilizados no desenvolvimento de *software*. Bibliotecas contêm código e dados auxiliares, que provêm serviços a programas independentes, o que permite o compartilhamento e a alteração de código e dados de forma modular. Fonte: “Biblioteca (computação)” (2022).
- a) Errado ()
 - b) Certo (X)
14. Alguns executáveis são tanto programas independentes quanto bibliotecas, mas a maioria das bibliotecas não é executáveis. Fonte:
- a) Errado ()
 - b) Certo (X)
15. A maior parte dos sistemas operacionais modernos provê bibliotecas que implementam a maioria dos serviços do sistema, que transformaram em comodidades os serviços que uma aplicação moderna espera providos pelo sistema operacional. Assim sendo, a maior parte do código utilizado em aplicações modernas é fornecido por estas bibliotecas. Fonte: “Biblioteca (computação)” (2022).
- a) Errado ()
 - b) Certo (X)
16. A maior parte dos sistemas operacionais modernos não provê bibliotecas para implementam a maioria dos serviços do sistema. Assim sendo a maior parte do código utilizado em aplicações modernas não é fornecido por estas bibliotecas. Fonte: “Biblioteca (computação)” (2022).
- a) Errado (X)
 - b) Certo ()
17. Quando um programa é desenvolvido utilizando uma linguagem de programação pode usar um conjunto de funções pré-escritas por outros programadores que já resolvem determinados problemas para que os programadores não precisem em todo programa “reinventar a roda”, a esse conjunto de funções damos o nome de? Fonte: “Biblioteca (computação)” (2022).
- a) Variável ()
 - b) *String* ()
 - c) Biblioteca (X)
 - d) *Char* ()
 - e) *Integer* ()

Sobre variáveis

18. Sobre variáveis escolha a opção correta.
- Na programação, uma variável é um objeto (uma posição, frequentemente localizada na memória) capaz de reter e representar um valor ou expressão. Fonte: “Variável (programação)” (2022).

- a) Errado ()
- b) Certo (X)

19. Sobre variáveis escolha a opção correta.

Variável do tipo Inteiro: armazena números que não possuem casas decimais, apenas a parte inteira e podem ser positivos ou negativos (ex: 1, 2, 67, 999). Fonte: “Variável (programação)” (2022).

- a) Errado ()
- b) Certo (X)

20. Variável do tipo Inteiro: armazena caracteres (ex: A, B, C, a, b) e símbolos como (ex: @, \$, %). Fonte: “Variável (programação)” (2022).

- a) Errado (X)
- b) Certo ()

21. Sobre variáveis escolha a opção correta.

Variável do tipo Real: números que possuem casas decimais e também podem ser positivos, negativos e inteiros (ex: 1, 2.3, 5.7, 78) para usar casas decimais é necessário usar PONTO ao invés de VÍRGULA. Fonte: “Variável (programação)” (2022).

- a) Errado ()
- b) Certo (X)

22. Sobre variáveis escolha a opção correta.

Variável do tipo *Char* “Caractere”: nesse tipo, apenas um caractere pode ser armazenado, pode ser uma letra, número ou símbolo (dependendo da linguagem), conforme o padrão ASCII (ex: A, B, 7, J) os números serão interpretados como caractere então não terão compatibilidade com inteiros e reais. Fonte: “Variável (programação)” (2022).

- a) Errado ()
- b) Certo (X)

23. Sobre variáveis escolha a opção correta.

Variável do tipo *String*: são espaços para você armazenar um conjunto de caracteres “palavras”, frases e qualquer coisa que utilize vários caracteres (ex: “Olá mundo”). Fonte: “Variável (programação)” (2022).

- a) Errado ()
- b) Certo (X)

24. Uma variável é um espaço de memória reservado para armazenar um certo tipo de dado, que pode ser um número, caractere, símbolo. Fonte: “Variável (programação)” (2022).

- a) Errado ()
- b) Certo (X)

25. Variável tem um nome para referenciar o seu conteúdo. Fonte: “Variável (programação)” (2022).
- a) Errado ()
 - b) Certo (X)
26. Sobre variáveis escolha a opção correta. Fonte: “Variável (programação)” (2022).
- a) () O valor de uma variável é o conjunto da(s) lista(s) de controle acessadas por ela.
 - b) () O valor de uma variável é o conteúdo do(s) caractere(s) alfanuméricos de sua denominação.
 - c) () O identificador de uma variável é o conteúdo da(s) célula(s) de sequenciamento associada(s) a ela.
 - d) (X) O valor de uma variável é o conteúdo da(s) célula(s) de memória associada(s) a ela. (Correta)
 - e) () O valor de uma variável de controle é o identificador de uma estrutura de repetição.

Sobre Vetores e Matrizes

27. Vetor (*array* uni-dimensional) é uma variável que armazena várias variáveis do mesmo tipo. Fonte: De Oliveira Alves (2013a)
- a) Errado ()
 - b) Certo (X)
28. Os Vetores são matrizes de uma só dimensão. Fonte: De Oliveira Alves (2013a)
- a) Errado ()
 - b) Certo (X)
29. Os Vetores são *array* de múltiplas dimensões. Fonte: De Oliveira Alves (2013a)
- a) Errado (X)
 - b) Certo ()
30. Uma matriz é uma coleção de variáveis de mesmo tipo, acessíveis com um único nome e armazenados contiguamente na memória. A individualização de cada variável de um vetor é feita através do uso de índices. Fonte: De Oliveira Alves (2013a)
- a) Errado ()
 - b) Certo (X)

Sobre Operadores

Operadores Aritméticos

31. Operadores aritméticos são aqueles que estudamos na escola, aquelas funções básicas de somar (+), subtrair (-), multiplicar (*), dividir (/), operador de módulo ou resto da divisão (%). Fonte: Eduardo (2012 e De Oliveira Alves (2013a)

- a) Errado ()
- b) Certo (X)

32. Operadores aritméticos são representados por: (se, então, senão)? Fonte: Eduardo (2012 e De Oliveira Alves (2013a)

- a) Errado (X)
- b) Certo ()

Operadores Relacionais

33. Os operadores relacionais comparam dois valores e retornam um valor booleano (*true* ou *false*). Os operadores são Igual a (==), Diferente de (!=), Maior que (>), Menor que (<), Maior do que ou igual a (>=), Menor do que ou igual a (<=). Fonte: Corob-msft (2022).

- a) Errado (X)
- b) Certo ()

34. Os operadores relacionais comparam dois valores e retornam um valor booleano (*true* ou *false*). O exemplo a seguir ajuda a ilustrar o uso destes operadores? Fonte: Corob-msft (2022).

int a = 10, b = 25, c = 50, d = 100;

Esta operação (a == d) tem como resultado TRUE?

- a) Errado (X)
- b) Certo ()

35. Os operadores relacionais comparam dois valores e retornam um valor booleano (*true* ou *false*). O exemplo a seguir ajuda a ilustrar o uso destes operadores. Fonte: Corob-msft (2022).

int a = 10, b = 25, c = 50, d = 100;

Esta operação (a == d) tem como resultado FALSE?

- a) Errado ()
- b) Certo (X)

Operadores Lógicos

36. O operador lógico E (AND) somente retorna verdadeiro se todas as condições de entrada forem verdadeiras. Por exemplo, (3 > 2) é uma expressão lógica que possui valor verdadeiro. Por outro lado, (4 < 1) é uma expressão lógica que possui valor falso. Analise o "E" (&&) do exercício abaixo com código na "linguagem C". Fonte: Eduardo (2019)

((3 < 2) && (2 == 2)): resultado **FALSO**?

- a) Errado ()
- b) Certo (X)

37. O operador lógico E (AND) somente retorna verdadeiro se todas as condições de entrada forem verdadeiras. Por exemplo, (3 > 2) é uma expressão lógica que possui valor verdadeiro. Por outro lado, (4 < 1) é uma expressão lógica que possui valor falso. Analise o “E” (&&) do exercício abaixo com código na “linguagem C”. Fonte: Eduardo (2019)

((3 < 2) && (2 == 2)): resultado **VERDADEIRO**?

- a) Errado (X)
- b) Certo ()

38. O operador lógico OU (OR) somente retorna falso se todas as condições de entrada forem falsas. Por exemplo, (3 > 2) é uma expressão lógica que possui valor verdadeiro. Por outro lado, (4 < 1) é uma expressão lógica que possui valor falso. Analise o “OU” (||) do exercício abaixo com código na “linguagem C”. Fonte: Eduardo (2019)

((2 > 1) || (3 < 7)): resultado **VERDADEIRO**?

- a) Errado ()
- b) Certo (X)

39. O operador lógico OU (OR) somente retorna falso se todas as condições de entrada forem falsas. Por exemplo, (3 > 2) é uma expressão lógica que possui valor verdadeiro. Por outro lado, (4 < 1) é uma expressão lógica que possui valor falso. Analise o “OU” (||) do exercício abaixo com código na “linguagem C”. Fonte: Eduardo (2019)

((2 > 1) || (3 < 7)): resultado **FALSO**?

- a) Errado (X)
- b) Certo ()

40. O operador lógico NÃO (NOT) inverte a condição de entrada: verdadeiro se torna falso, e falso se torna verdadeiro. Por exemplo, (3 > 2) é uma expressão lógica que possui valor verdadeiro. Fonte: Eduardo (2019)

A operação: !(3 > 2), possui resultado **VERDADEIRO**?

- a) Errado (X)
- b) Certo ()

41. O operador lógico NÃO (NOT) inverte a condição de entrada: verdadeiro se torna falso, e falso se torna verdadeiro. Por exemplo, (3 > 2) é uma expressão lógica que possui valor verdadeiro. Fonte: Eduardo (2019)

!(3 > 2): resultado **FALSO**?

- a) Errado ()
- b) Certo (X)

Estrutura Condicional

42. Os comandos de decisão, como, por exemplo, os comandos *IF*, *THEN*, *ELSE*, não são considerados como comandos de controle de fluxo pelo fato de utilizarem operadores aritméticos. Fonte: Neiva (2022).
- a) Errado ☒ (X)
b) Certo ☐ ()
43. Em ciência da computação, indentação (reco, neologismo derivado da palavra em inglês *indentation*) é um termo aplicado ao código-fonte de um programa para ressaltar ou definir a estrutura do algoritmo. Na maioria das linguagens de programação, a indentação é empregada visando ressaltar a estrutura do algoritmo, aumentando assim a legibilidade do código. Em algumas linguagens, entretanto, a indentação é obrigatória. Python, Occam e Haskell, por exemplo, utilizam a mudança de indentação para definir a hierarquia dentre blocos de código. Fonte: Neiva (2022).
- a) Errado ☐ ()
b) Certo ☒ (X)
44. Sobre indentação, enquanto os blocos são delimitados explicitamente em C, Java e PHP por chaves e em Pascal e Fortran por palavras-chave como *then* e *endif*, em Python blocos são delimitados por espaços ou tabulações formando uma indentação visual; não existem símbolos de “abre” e “fecha”. Fonte: “Python/Conceitos básicos/Indentação” (2022).
- a) Errado ☐ ()
b) Certo ☒ (X)
45. Indentação não deixa o código mais legível, não serve para melhorar a manutenção do código e nem para facilitar a tarefa de encontrar erros na codificação. Fonte: Neitzke (2022).
- a) Errado ☒ (X)
b) Certo ☐ ()
46. Python requer uma indentação padronizada. Em outras linguagens, como C/C++ ou JavaScript, a indentação não é necessária devido aos delimitadores de blocos, utilizada somente para melhor visualização. Fonte: “Python/Conceitos básicos/Indentação” (2022).
- a) Errado ☐ ()
b) Certo ☒ (X)
47. Python não requer uma indentação padronizada. Em outras linguagens, como C/C++ ou JavaScript, a indentação é necessária devido aos delimitadores de blocos, utilizada somente para melhor visualização. Fonte: “Python/Conceitos básicos/Indentação” (2022).
- a) Errado ☒ (X)
b) Certo ☐ ()
48. Estrutura de seleção (expressão condicional ou ainda construção condicional) é, na ciência da computação, uma estrutura de desvio do fluxo de controle presente em linguagens de programação que realiza diferentes

computações ou ações dependendo se a seleção (ou condição) é verdadeira ou falsa, em que a expressão é processada e transformada em um valor booleano. Nas linguagens de programação, usamos as palavras em inglês para expressar uma estrutura de seleção, como *if*, *else if* e *else*. Fonte: “Estrutura de seleção” (2022).

- a) Errado ()
- b) Certo (X)

49. Quando o interpretador encontra o identificador "Se", ele espera em seguida uma condição booleana sob forma de expressão relacional (como, por exemplo, $(x > 0)$), que pode ser verdadeira ou falsa. Se a condição for verdadeira, o bloco de código seguido de "Então" é executado. Caso a condição seja falsa, o bloco de código seguido de "Senão" é executado. Em ambos os casos, após a execução do bloco de código, o fluxo do programa é retornado para o ponto indicado por "Fim Se". Note que a parte do "Senão" é opcional e pode ser omitida. Caso omitida, a estrutura é chamada de seleção simples; caso contrário é chamada seleção composta. Fonte: “Estrutura de seleção” (2022).

- a) Errado ()
- b) Certo (X)

50. Seleção simples e composta. A condição "se-então (senão)" é uma estrutura de seleção comum em diversas linguagens de programação. Sua estrutura básica é a seguinte: Fonte: “Estrutura de seleção” (2022).

Se (condição) Então
(bloco de código)

Senão
(bloco de código)

Fim Se

- a) Errado ()
- b) Certo (X)

51. A estrutura "Senão Se" é formada pela composição de estruturas de seleção, e um exemplo de sua estrutura é o seguinte: Fonte: “Estrutura de seleção” (2022).

Se (condição) Então
(bloco de código)

Senão
Se (condição) Então
(bloco de código)

Fim Se

Fim Se

- a) Errado ()
- b) Certo (X)

52. Seleção de múltipla escolha. A seleção de múltipla escolha compara um dado valor a **constantes**, desviando o fluxo de código para o ponto indicado pela primeira constante que é igual à **constante** escolhida. Sua estrutura básica é: Fonte: “Estrutura de seleção” (2022).

Escolha X

Caso V1:
(bloco de código)

Caso V2:
(bloco de código)

Caso contrário:
(bloco de código)

Fim Escolha

- a) Errado ()
- b) Certo (X)

Estrutura de repetição

53. Estrutura de repetição permite executar mais de uma vez o mesmo comando ou conjunto de comandos, de acordo com uma condição ou com um contador. Fonte: “Estrutura de seleção” (e Zanelato (2018)

- a) Errado ()
- b) Certo (X)

54. Estrutura de repetição permite executar somente uma vez o mesmo comando ou conjunto de comandos, de acordo com uma condição ou com um contador. Fonte: “Estrutura de seleção” (e Zanelato (2018)

- a) Errado (X)
- b) Certo ()

55. Estrutura de repetição são utilizadas para repetir ações semelhantes, executadas para todos os elementos de uma lista de dados, ou simplesmente para repetir um mesmo bloco de código até que a condição de parada seja satisfeita. Fonte: “Estrutura de seleção” (e Zanelato (2018)

- a) Errado ()
- b) Certo (X)

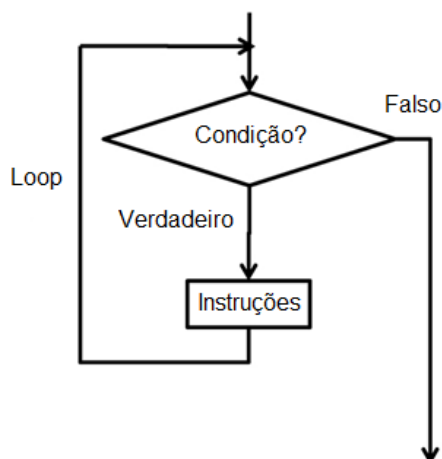
56. Estrutura de Repetição Enquanto (While)

Essa estrutura realiza um teste lógico no início do loop, e cada vez que o teste retorna o valor verdadeiro, os comandos associados ao laço são executados.

Quando o teste retornar falso, o laço é encerrado e o processamento volta para o fluxo principal do programa. Fonte: “Estrutura de seleção” (e Zanelato (2018)

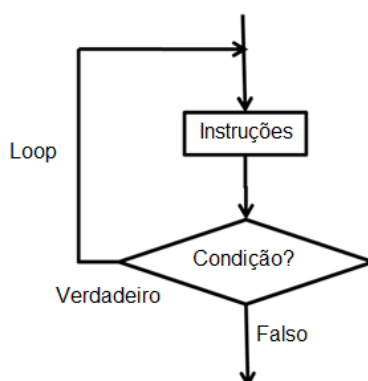
- a) Errado ()
- b) Certo (X)

57. A ilustração a seguir mostra o funcionamento de qual estrutura de repetição?
Fonte: Autoria própria.



- a) ☐ For (Para)
- b) ☐ Do While (Faça enquanto)
- c) ☒ While (Enquanto)

58. A ilustração a seguir mostra o funcionamento de qual estrutura de repetição?
 Fonte: Autoria própria.



- a) ☐ For (Para)
- b) ☐ While (Enquanto)
- c) ☒ Do While (Faça enquanto)

59. O Do while tem quase o mesmo funcionamento que o while, mas neste caso, o bloco de comando estará dentro das chaves do comando DO e a verificação será feita pelo WHILE logo após a execução, a diferença é que com o uso dele teremos os comandos executados ao menos uma única vez. Fonte: "Programação C/C++ - Comandos de Repetição" (2022).

- a) Errado ☐
- b) Certo ☒

Sobre Funções

60. As funções (functions), também conhecidas como sub-rotinas, são muito utilizadas em programação. Um dos grandes benefícios é não precisar copiar o código todas as vezes que precisar executar aquela operação, além de deixar a leitura do código mais intuitiva. No exemplo anterior, caso precisássemos descobrir a raiz quadrada de 10 números, bastaria chamar a função que calcula a raiz quadrada 10 vezes. Fonte: De Oliveira Alves (2013b)

- a) Errado ()
- b) Certo (X)

61. Exemplo de Programa em C.

```
int main( )
{
    return 0;
}
```

Essa função main, neste exemplo, é do tipo int

Significa que a função deverá obrigatoriamente retornar um **número inteiro**.

Fonte: Autoria própria.

- a) Errado ()
- b) Certo (X)

Sobre Procedimentos

62. Os procedimentos (procedures) diferem das funções apenas por não retornarem resultado. Enquanto que a função é um algoritmo que será executado e produzirá um resultado final concreto que poderá ser usado por outro algoritmo. Fonte: De Oliveira Alves (2013b)

- a) Errado ()
- b) Certo (X)

Sobre Registros

63. Um registro é uma estrutura de dados heterogênea, ou seja, formada por um conjunto de variáveis (campos) que podem assumir tipos diferentes de dados, inclusive os tipos compostos (vetores, matrizes e registros). Fonte: "Struct - Registros em Linguagem C com struct - Embarcados" (2017)

- a) Errado ()
- b) Certo (X)

64. Um registro é uma estrutura de dados não heterogênea, ou seja, formada por um conjunto de variáveis (campos) que podem assumir tipos iguais de dados. Fonte: "Struct - Registros em Linguagem C com struct - Embarcados" (2017)

- a) Errado (X)
- b) Certo ()

65. Em Linguagem C Registros são chamados de Estruturas e a palavra-chave é Struct. Fonte: “Struct - Registros em Linguagem C com struct - Embarcados” (2017)

Ex: Registros Linguagem C.

```
struct aluno {
    int codigo;
    char nome[200];
    float nota;
};
struct aluno aluno_especial, aluno_regular, aluno_ouvinte;
```

a) Errado ()
b) Certo (X)

Sobre: Programação Orientada a Objetos.

66. Uma classe é um tipo definido pelo usuário que contém a “receita”, a especificação para os objetos, algo mais ou menos como o tipo inteiro contém o molde para as variáveis declaradas como inteiros. A classe envolve, associa, funções e dados, controlando o acesso a estes, defini-la implica em especificar os seus atributos (dados) e seus métodos (funções). Fonte: “Caelum Escola de Tecnologia Cursos Online” (2022).

a) Errado ()
b) Certo (X)

67. Objetos (computacionais) são caracterizados por atributos e métodos. Atributos são as propriedades de um objeto. Métodos são as ações que um objeto pode realizar. Os objetos são características definidas pelas classes. Neles é permitido instanciar objetos da classe para inicializar os atributos e invocar os métodos. Fonte: “Caelum Escola de Tecnologia Cursos Online” (2022).

a) Errado ()
b) Certo (X)

68. Atributos são as características de um objeto, essas características também são conhecidas como variáveis, utilizando o exemplo dos cães, temos alguns atributos, tais como: cor, peso, altura e nome. Fonte: “Caelum Escola de Tecnologia Cursos Online” (2022).

a) Errado ()
b) Certo (X)

69. Métodos são as ações que os objetos podem exercer quando solicitados, onde podem interagir e se comunicarem com outros objetos, utilizando o exemplo dos cães, temos alguns exemplos: latir, correr, pular. Fonte: “Caelum Escola de Tecnologia Cursos Online” (2022).

a) Errado ()
b) Certo (X)

70. Abstração é a habilidade de concentrar nos aspectos essenciais de um contexto qualquer, ignorando características menos importantes ou

acidentais. Em modelagem orientada a objetos, uma classe é uma abstração de entidades existentes no domínio do sistema de software. Fonte: “Caelum Escola de Tecnologia Cursos Online” (2022).

- a) Errado ()
- b) Certo (X)

71. Generalizar uma classe, sob o conceito de herança, indica dizer que estamos centralizando as características comuns às classes que precisam herdá-la. As classes que herdam essa classe base, têm como objetivo especializá-la; dando assim um formato específico e característico de seu conteúdo, contudo ao utilizarmos uma classe especializada teremos acesso, tanto a sua estrutura em particular, quanto a estrutura da classe mais genérica que está sendo herdada com a finalidade de aplicação dos conceitos de generalização e especialização. Fonte: “Caelum Escola de Tecnologia Cursos Online” (2022).

- a) Errado ()
- b) Certo (X)

72. Polimorfismo é a capacidade de dar a um método ou a um objeto, instância de uma classe, a possibilidade de transformação em outro tipo. Permitir que um objeto assuma "várias formas" ou permitir que seja reescrito através de herança e uso de regras polimórficas, por ser em um momento inicialmente abstrato, faz com que o polimorfismo se torne uma boa prática de programação. Fonte: “Caelum Escola de Tecnologia Cursos Online” (2022).

- a) Errado ()
- b) Certo (X)

73. O encapsulamento permite ocultar a complexidade do código. Não é necessário entender como a operação funciona para poder utilizá-la. Um exemplo clássico é o de um motorista conduzindo um veículo. O motorista não precisa entender de mecânica e saber exatamente como o motor funciona internamente para poder dirigir o automóvel. Os detalhes da “implementação” do motor ficam ocultos do motorista. O encapsulamento protege os dados, permitindo o acesso a eles apenas a partir de métodos específicos e autorizados. Fonte: “Caelum Escola de Tecnologia Cursos Online” (2022).

- a) Errado ()
- b) Certo (X)

REFERÊNCIAS

Biblioteca (computação). Disponível em: <[https://pt.wikipedia.org/wiki/Biblioteca_\(computa%C3%A7%C3%A3o\)](https://pt.wikipedia.org/wiki/Biblioteca_(computa%C3%A7%C3%A3o))>. Acesso em: 20/1/2022.

Caelum Escola de Tecnologia Cursos Online. Disponível em: <<https://www.caelum.com.br/apostila-java-orientacao-objetos/orientacao-a-objetos-basica>>. Acesso em: 20/1/2022.

Case-sensitive. Disponível em: <<https://pt.wikipedia.org/wiki/Case-sensitive>>. Acesso em: 20/1/2022a.

Case-sensitive. Disponível em: <<https://pt.wikipedia.org/wiki/Case-sensitive>>. Acesso em: 20/1/2022b.

Compilador. Disponível em: <<https://pt.wikipedia.org/wiki/Compilador>>. Acesso em: 20/1/2022a.

Compilador. Disponível em: <<https://pt.wikipedia.org/wiki/Compilador>>. Acesso em: 20/1/2022b.

Compilador. Disponível em: <<https://pt.wikipedia.org/wiki/Compilador>>. Acesso em: 20/1/2022c.

CONCURSOS, T. Questão 617938 CEBRASPE (CESPE) - Oficial Técnico de Inteligência. Disponível em: <<https://www.teconconcursos.com.br/questoes/617938>>. Acesso em: 20/1/2022.

COROB-MSFT. Operadores relacionais e de igualdade C. Disponível em: <<https://docs.microsoft.com/pt-br/cpp/c-language/c-relational-and-equality-operators>>. Acesso em: 20/1/2022.

EDUARDO. Operadores Aritméticos em Linguagem C. Disponível em: <<http://linguagemc.com.br/operadores-aritmeticos-em-linguagem-c/>>. Acesso em: 20/1/2022.

EDUARDO. Operadores Lógicos em C. Disponível em: <<http://linguagemc.com.br/operadores-logicos-em-c/>>. Acesso em: 20/1/2022.

Estrutura de seleção. Disponível em: <https://pt.wikipedia.org/wiki/Estrutura_de_sele%C3%A7%C3%A3o>. Acesso em: 20/1/2022.

Linguagem compilada. Disponível em: <https://pt.wikipedia.org/wiki/Linguagem_compilada>. Acesso em: 20/1/2022.

Linguagem interpretada. Disponível em: <https://pt.wikipedia.org/wiki/Linguagem_interpretada>. Acesso em: 20/1/2022a.

Linguagem interpretada. Disponível em: <https://pt.wikipedia.org/wiki/Linguagem_interpretada>. Acesso em: 20/1/2022b.

NEITZKE, N. A. Boas práticas de Programação: Organizando o Código e seus Pacotes. Disponível em: <<https://www.devmedia.com.br/boas-praticas-de-programacao/21137>>. Acesso em: 20/1/2022.

NEIVA. Linguagem C: estruturas condicionais. Disponível em: <<https://br.ccm.net/faq/10200-linguagem-c-estruturas-condicionais>>. Acesso em: 20/1/2022.

DE OLIVEIRA ALVES, G. F. O que são Vetores e Matrizes (arrays). Disponível em: <<https://dicasdeprogramacao.com.br/o-que-sao-vetores-e-matrizes-arrays>>. Acesso em: 20/1/2022a.

DE OLIVEIRA ALVES, G. F. O que são Funções e Procedimentos? Disponível em: <<https://dicasdeprogramacao.com.br/o-que-sao-funcoes-e-procedimentos>>. Acesso em: 20/1/2022b.

Palavra reservada. Disponível em: <https://pt.wikipedia.org/wiki/Palavra_reservada>. Acesso em: 20/1/2022.

Programação C/C++ - Comandos de Repetição. Disponível em: <<https://www.inf.pucrs.br/~pinho/Laprol/ComandosDeRepeticao/Repeticao.html>>. Acesso em: 20/1/2022.

Python/Conceitos básicos/Indentação. Disponível em: <https://pt.wikibooks.org/wiki/Python/Conceitos_b%C3%A1sicos/Indenta%C3%A7%C3%A3o>. Acesso em: 20/1/2022a.

Python/Conceitos básicos/Indentação. Disponível em: <https://pt.wikibooks.org/wiki/Python/Conceitos_b%C3%A1sicos/Indenta%C3%A7%C3%A3o>. Acesso em: 20/1/2022b.

Questões de Concursos. Disponível em: <<https://www.qconcursos.com/questoes-de-concursos/questoes/026552a2-59>>. Acesso em: 20/1/2022.

Struct - Registros em Linguagem C com struct - Embarcados. Disponível em: <<https://www.embarcados.com.br/struct-registros-em-linguagem-c/>>. Acesso em: 20/1/2022.

Variável (programação). Disponível em: <[https://pt.wikipedia.org/wiki/Vari%C3%A1vel_\(programa%C3%A7%C3%A3o\)](https://pt.wikipedia.org/wiki/Vari%C3%A1vel_(programa%C3%A7%C3%A3o))>. Acesso em: 20/1/2022.

ZANELATO, J. Lógica de Programação - estruturas de repetição. Disponível em: <<https://podprogramar.com.br/logica-de-programacao-estruturas-de-repeticao/>>. Acesso em: 20/1/2022.