

Questões

1.4 Preencha as lacunas em cada uma das seguintes afirmações:

- a) A unidade lógica que recebe informações de fora do computador para uso por ele é a un. de entrada.
- b) O processo de instrução do computador para resolver um problema específico é chamado programação.
- c) assembly é um tipo de linguagem de computador que utiliza abreviações em inglês para instruções de linguagem de máquina.
- d) un. de saída é uma unidade lógica que envia informações que já foram processadas pelo computador para vários dispositivos, de modo que possam ser utilizadas fora da máquina.
- e) Mem. RAM e Mem. secun. são unidades lógicas do computador que retêm informações.
- f) ULA é uma unidade lógica do computador que realiza cálculos.
- g) ULA é uma unidade lógica do computador que toma decisões lógicas.
- h) As linguagens mais convenientes para que o programador escreva programas rápida e facilmente são as Linguagens de alto nível.
- i) A única linguagem que um computador pode entender diretamente é a Linguagem de máquina dele.
- j) CPU é uma unidade lógica do computador que coordena as atividades de todas as outras unidades lógicas.

1.5 Preencha as lacunas em cada uma das seguintes afirmações:

- a) A linguagem de programação java é agora utilizada para desenvolver aplicativos corporativos de grande porte, aprimorar a funcionalidade de servidores da web, fornecer aplicativos a dispositivos de consumo popular e para muitos outros propósitos.
- b) Inicialmente, o C tornou-se muito conhecido como a linguagem de desenvolvimento do sistema operacional UNIX.
- c) O TCP garante que as mensagens, que consistem em partes sequencialmente numeradas chamadas bytes, sejam adequadamente encaminhadas do emissor para o receptor, cheguem intactas e sejam montadas na ordem correta.
- d) A linguagem de programação C++ foi desenvolvida por Bjarne Stroustrup no início dos anos 1980 na Bell Laboratories.

1.6 Preencha as lacunas em cada uma das seguintes afirmações:

- a) Os programas Java normalmente passam por cinco fases: Escrita, Compilar bytecodes, Carregar o Programa na memória, Verificação Dos bytecodes e execução.
- b) Um(a) IDE fornece muitas ferramentas que suportam o processo de desenvolvimento de software, como editores para escrever e editar programas, depuradores a fim de localizar erros de lógica em programas e muitos outros recursos.
- c) O comando java invoca JVM, que executa programas Java.
- d) Um(a) JVM é um aplicativo de software que simula um computador, mas oculta o sistema operacional e o hardware subjacentes dos programas que interagem com ela(e).
- e) O Carregador De classe Verificador transfere os arquivos `.class` contendo os bytecodes do programa para a memória principal.
- f) O De classe examina bytecodes para assegurar que eles são válidos.

1.7 Explique as duas fases de compilação de programas Java.

1.8 Um dos objetos mais comuns do mundo é um relógio de pulso. Discuta como cada um dos seguintes termos e conceitos se aplicam à noção de um relógio: objeto, atributos, comportamentos, classe, herança (considere, por exemplo, o alarme dele), modelagem, mensagens, encapsulamento, interface e ocultamento de informações.

1.7) A primeira fase de compilação começa quando o programador usa o comando `javac` no arquivo `.java` para converter o código fonte em bytecodes que são alocados no arquivo `.class` que é gerado após a primeira compilação. Já a segunda fase de compilação, ocorre quando os bytecodes são carregados na memória e o verificador se certifica de que está tudo certo para os bytecodes serem traduzidos em linguagem de máquina e por fim a aplicação ser executada.

1.8) Poderíamos ter uma superclasse `Number` com um atributo `valor`, subclasses `Hora`, `Minuto` e `Segundo` herdariam as particularidades da superclasse `Number` e teriam às suas próprias como seus intervalos específicos. Poderíamos ter também uma interface `ponteiro` com um atributo `number` e comportamentos como `avancar` e `voltar number`, classes como `PonteiroHora`, `PonteiroMinuto` e `PonteiroSegundo` implementariam essa interface e especificariam seus comportamentos de acordo com as características de cada uma. E por fim uma classe `relógio` teria atributos das classes criadas anteriormente e teria o comportamento de alarme, recebendo o valor da hora como parâmetro.