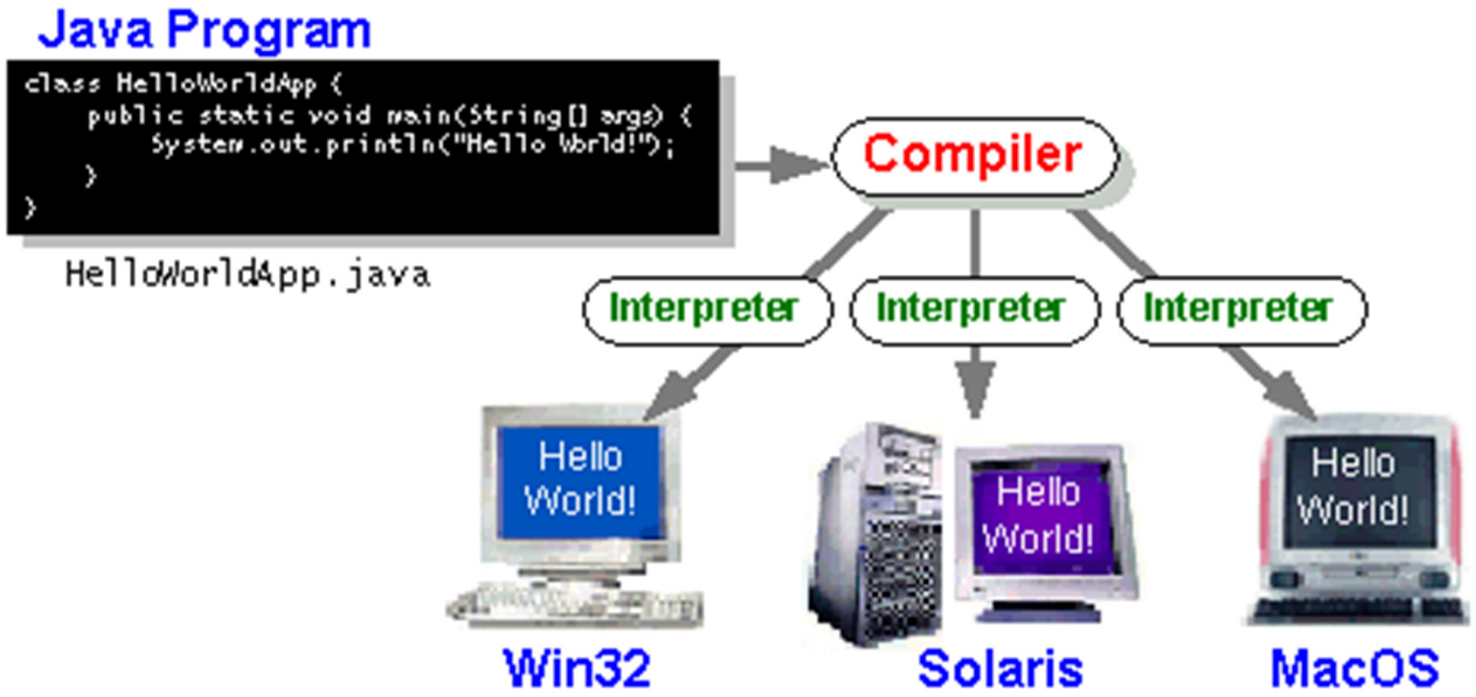
Programação Orientada a Objetos I

CÁSSIO CAPUCHO PEÇANHA – 02

O que buscamos?



Questões Léxicas

Questões Léxicas

- Durante a compilação, os caracteres no código Java são reduzidos a uma série de tokens.
- O compilador Java reconhece cinco tipos de tokens :
  - Identificadores
  - Palavras Reservadas
  - Literais
  - Separadores
  - Operadores
  - Comentários

Questões Léxicas - Identificadores

- Identificadores são usados para dar nomes a variáveis, métodos, classes e outras entidades no código.
- Eles são essencialmente rótulos que fornecem um nome único para que você possa se referir a uma determinada entidade no programa.
- Identificadores são usados para identificar e acessar diferentes elementos do código Java.

Regras para identificadores em Java:

1. Os identificadores devem começar com uma letra (a a z ou A a Z), um sublinhado (_) ou um cifrão ($).
2. Após o primeiro caractere, os identificadores podem conter letras, dígitos (0 a 9), sublinhados (_) ou cifrões ($).
3. Os identificadores são sensíveis a maiúsculas e minúsculas, ou seja, "nome" e "Nome" são identificadores diferentes em Java.
4. Não é permitido usar palavras-chave do Java como identificadores. Por exemplo, você não pode usar "class", "public", "int", etc., como identificadores.

Questões Léxicas - Identificadores

Exemplos de identificadores válidos em Java:

- idade
- nomeCompleto
- _saldo
- $totalVendas
- numeroDeTelefone

Exemplos de identificadores inválidos em Java:

- 123valor (começa com um dígito)
- valor total (espaço em branco não é permitido)
- public (palavra-chave do Java)
- while (palavra-chave do Java)

Questões Léxicas - Palavras Reservadas

- Palavras reservadas (ou palavras-chave) em Java são termos que têm um significado específico na linguagem e são reservados para uso exclusivo de fins específicos.
- Essas palavras têm funções específicas atribuídas pelo Java e não podem ser usadas como identificadores, ou seja, não podem ser usadas para dar nome a variáveis, métodos, classes ou outras entidades do código.
- As palavras reservadas em Java são usadas para definir a estrutura e a lógica do programa, e cada palavra tem um propósito específico dentro da linguagem.
- Essas palavras têm uma sintaxe especial e são reconhecidas pelo compilador do Java, o que significa que não podem ser redefinidas ou alteradas pelo programador.

Questões Léxicas - Palavras Reservadas

java

abstract	continue	for	new	switch
assert	default	if	package	synchronized
boolean	do	goto	private	this
break	double	implements	protected	throw
byte	else	import	public	throws
case	enum	instanceof	return	transient
catch	extends	int	short	try
char	final	interface	static	void
class	finally	long	strictfp	volatile
const	float	native	super	while

Declaração de Variáveis

- A declaração de variáveis é um conceito fundamental em programação e está diretamente relacionada aos conceitos de identificadores e palavras reservadas.
- Em programação, uma variável é uma área de memória que armazena um valor ou uma referência a um valor.
- Para utilizar uma variável em um programa, você precisa declará-la, ou seja, informar ao compilador que você deseja reservar um espaço de memória para armazenar um valor de um determinado tipo.

Declaração de Variáveis

- Em Java, a declaração de variáveis segue uma sintaxe específica.
- Você deve informar o tipo de dado da variável, seguido pelo nome do identificador (nome da variável), como mostrado abaixo:

java

```
tipoDeDado nomeDaVariavel;
```

Declaração de Variáveis

- Por exemplo, para declarar uma variável inteira chamada "idade", você faria o seguinte:

```java
int idade;
```

- Neste exemplo, "int" é uma palavra reservada que indica o tipo de dado inteiro em Java, e "idade" é o identificador (nome da variável) que foi escolhido para essa variável.

Declaração de Variáveis

- A declaração de variáveis é onde você informa ao compilador sobre a existência da variável e o tipo de dado que ela pode armazenar.
- Após a declaração, você pode atribuir um valor à variável usando o operador de atribuição (=) e, em seguida, usar a variável em outras partes do código.

java

```java
public class ExemploVariavel {
    public static void main(String[] args) {
        // Declaração da variável
        int idade;

        // Atribuição de valor à variável
        idade = 30;

        // Uso da variável em uma saída
        System.out.println("Idade: " + idade);
    }
}
```

Questões Léxicas - Literais

- Os literais são valores constantes que aparecem diretamente no código fonte de um programa, representando dados de tipos específicos.
- Eles são chamados de "literais" porque representam valores concretos que não mudam durante a execução do programa, ou seja, são valores fixos e conhecidos em tempo de compilação.
- Em outras palavras, os literais são valores que não precisam ser calculados ou avaliados, pois eles próprios representam o dado que será utilizado no programa.
- Eles são usados para representar informações fixas, como números, caracteres, strings, booleanos e até mesmo arrays.

Questões Léxicas - Literais

- Em linguagens de programação, como Java, os literais são escritos usando uma sintaxe específica para cada tipo de dado. Alguns exemplos de literais em Java são:
  - Literais numéricos: Exemplos de literais numéricos são: 10, 3.14, 1000L, 0b1010 (binário), 0xFF (hexadecimal).
  - Literais de caracteres: Literais de caracteres são escritos entre aspas simples, como 'A', 'b', '1'.
  - Literais de strings: Literais de strings são escritos entre aspas duplas, como "Olá", "Mundo".
  - Literais booleanos: Literais booleanos representam os valores true e false.
  - Literais de arrays: Literais de arrays representam uma sequência de valores, como {1, 2, 3, 4}.

Questões Léxicas - Literais

java

```java
public class ExemploLiterais {
    public static void main(String[] args) {
        int idade = 25;          // Literal numérico
        char letra = 'A';        // Literal de caractere
        String nome = "João";    // Literal de string
        boolean verdadeiro = true; // Literal booleano
        int[] numeros = {1, 2, 3, 4}; // Literal de array
    }
}
```

Tipos

- Em Java, os tipos se referem às categorias de dados que podem ser usadas para declarar variáveis, definir parâmetros de métodos e retornar valores de funções.
- Cada variável em Java possui um tipo que define o tipo de dados que ela pode armazenar.
- Java é uma linguagem de programação fortemente tipada, o que significa que todas as variáveis devem ser declaradas com um tipo específico e o tipo da variável não pode ser alterado após a declaração.

Tipos

Os tipos em Java podem ser divididos em duas categorias principais:

- Tipos primitivos: São os tipos básicos incorporados na linguagem e representam valores simples. Existem oito tipos primitivos em Java:
  - byte: Números inteiros de 8 bits.
  - short: Números inteiros de 16 bits.
  - int: Números inteiros de 32 bits.
  - long: Números inteiros de 64 bits.
  - float: Números de ponto flutuante de 32 bits.
  - double: Números de ponto flutuante de 64 bits.
  - char: Representa um único caractere Unicode de 16 bits.
  - boolean: Representa um valor lógico (verdadeiro ou falso).
- Tipos de referência: São os tipos que representam objetos em Java. Eles são criados a partir de classes ou interfaces definidas pelo programador. Alguns exemplos de tipos de referência em Java incluem:
  - String: Representa uma sequência de caracteres.
  - Arrays: Representa coleções de elementos do mesmo tipo.
  - Classes e interfaces definidas pelo usuário.

Questões Léxicas - Separadores

- Em Java, os separadores referem-se aos caracteres especiais que são usados para delimitar ou separar elementos diferentes no código fonte.
- Esses caracteres são importantes para definir a estrutura do código, facilitando sua leitura e compreensão.

Questões Léxicas - Separadores

Os principais separadores em Java incluem:

- Ponto e vírgula (;): O ponto e vírgula é um dos separadores mais utilizados em Java. Ele é usado para indicar o fim de uma instrução. Cada instrução em Java termina com um ponto e vírgula, permitindo que o compilador saiba onde termina uma instrução e começa a próxima.
- Chaves ({ e }): As chaves são usadas para delimitar blocos de código. Um bloco de código é um grupo de instruções relacionadas, como o corpo de uma classe, método, loop ou estrutura condicional. As chaves indicam o início e o término de um bloco de código.

Questões Léxicas - Separadores

- Parênteses (() e)): Os parênteses são usados para criar e chamar métodos e funções em Java. Eles são usados para definir os parâmetros de um método e também para indicar onde os argumentos são passados ao chamar um método.
- Vírgula (,): A vírgula é usada para separar múltiplos argumentos ou elementos em uma lista. Por exemplo, em uma chamada de método com vários parâmetros, os argumentos são separados por vírgulas.
- Dois-pontos (:): O dois-pontos é usado em Java para indicar a declaração de um tipo em uma variável, como em declarações de variáveis e nos casos de declaração de switch.

Questões Léxicas - Operadores

- Operadores são símbolos especiais que permitem realizar operações entre variáveis, constantes e valores.
- Eles são usados para manipular e processar dados, realizando cálculos matemáticos, comparações lógicas e operações bitwise, entre outras funcionalidades.
- Os operadores são uma parte fundamental da linguagem de programação, pois eles fornecem as ferramentas necessárias para criar lógica e funcionalidades nos programas.

Os operadores em Java podem ser classificados em diferentes categorias:

- Operadores aritméticos
- Operadores de atribuição
- Operadores de comparação
- Operadores lógicos
- Operadores bitwise

Questões Léxicas - Operadores

Operadores aritméticos: Usados para realizar operações matemáticas básicas, como adição, subtração, multiplicação, divisão, módulo (resto da divisão) e operações de incremento e decremento.

Exemplos:

- + (adição)
- - (subtração)
- * (multiplicação)
- / (divisão)
- % (módulo)
- ++ (incremento)
- -- (decremento)

Questões Léxicas - Operadores

Operadores de atribuição: Usados para atribuir valores a variáveis.

Simples: Os operadores de atribuição simples são representados pelo sinal de igual (=) e são usados para atribuir um valor diretamente a uma variável.

Eles são a forma mais básica de atribuição e são usados para atualizar o valor da variável com um novo valor.

A sintaxe é simples: variavel = valor.

```java
int x = 10; // A variável x recebe o valor 10.
```

Questões Léxicas - Operadores

Composto: Combinam um operador aritmético com o operador de atribuição simples. Esses operadores realizam uma operação aritmética entre o valor atual da variável e um valor específico, e atribuem o resultado de volta à variável.

Eles são uma forma abreviada de realizar uma operação e atualizar o valor da variável ao mesmo tempo. São representados por um operador específico seguido do sinal de igual (=).

Alguns exemplos comuns são: **+=**, **-=**, ***=**, **/=**, **%=**

java

```java
int y = 5;
y += 3; // Equivalente a y = y + 3, atualiza o valor de y para 8.
int z = 10;
z *= 2; // Equivalente a z = z * 2, atualiza o valor de z para 20.
z -= 5; // Equivalente a z = z - 5, atualiza o valor de z para 15.
```

Questões Léxicas - Operadores

Operadores de comparação:
Usados para comparar valores e produzir um resultado booleano (verdadeiro ou falso).

Exemplos:

- == (igual a)
- != (diferente de)
- < (menor que)
- > (maior que)
- <= (menor ou igual a)
- >= (maior ou igual a)

Questões Léxicas - Operadores

Operadores lógicos: Usados para combinar valores booleanos ou inverter o valor de uma expressão.

Exemplos:

- && (E lógico)
- || (OU lógico)
- ! (NÃO lógico)

Questões Léxicas - Operadores

Operadores bitwise: Usados para realizar operações bit a bit em valores inteiros.

Exemplos:

- & (AND bitwise)
- | (OR bitwise)
- ^ (XOR bitwise)
- ~ (Complemento bitwise)

Questões Léxicas - Comentários

Em Java, comentários são trechos de texto adicionados ao código fonte para explicar o funcionamento, fornecer informações ou fazer anotações que não serão interpretadas pelo compilador.

Os comentários são ignorados durante a compilação e a execução do programa, servindo apenas como notas para o programador ou outros membros da equipe.

Questões Léxicas - Comentários

Existem tipos de comentários em Java:

Comentários de linha única:

São utilizados para inserir informações em uma única linha. Eles começam com duas barras (//) e continuam até o final da linha.

```java
// Este é um comentário de linha única.
int idade = 30; // Declaração da variável idade.
```

Comentários de bloco:

São utilizados para inserir informações em múltiplas linhas. Eles começam com uma barra seguida de asterisco (/*) e terminam com um asterisco seguido de barra (*/).

```java
/*
Este é um comentário
de bloco que abrange
múltiplas linhas.
*/
```

Entrada/Saída

Entrada/Saída de Dados

- Toda linguagem de programação deve prover um meio de interação com o usuário.
- O meio mais básico é o uso do console, com entrada de dados pelo teclado e saída em texto;
- Em Java, a entrada e saída de dados básica é realizada por meio das classes do pacote java.util e java.io, que fornecem recursos para ler dados do teclado (entrada) e escrever dados na tela (saída).

Entrada/Saída de Dados

Entrada de Dados (Input):

- Para ler dados do teclado, podemos usar a classe **Scanner**, que faz parte do pacote java.util.
- Ela permite ler diferentes tipos de dados, como números inteiros, números de ponto flutuante e strings, entre outros.

```java
import java.util.Scanner;
public class ExemploEntrada {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Digite um número inteiro: ");
        int numeroInteiro = scanner.nextInt();

        System.out.println("O número inteiro digitado é: " + numeroInteiro);

        scanner.close(); // Fechar o scanner quando não for mais necessário
    }
}
```

Entrada/Saída de Dados

Saída de Dados (Output):

- Para imprimir dados na tela, podemos usar o método **System.out.println()**. Esse método imprime o valor entre parênteses na saída padrão (normalmente, a tela).

java

```java
public class ExemploSaida {
    public static void main(String[] args) {
        System.out.println("Olá, mundo!");
    }
}
```

Entrada/Saída de Dados

Formatação de Saída:

Para formatar a saída de dados, podemos usar **System.out.printf()**, que permite imprimir valores formatados com base em especificadores de formato.

java

```java
public class ExemploFormatado {
    public static void main(String[] args) {
        String nome = "João";
        double altura = 1.75;

        System.out.printf("Nome: %s, Altura: %.2f metros%n", nome, altura);
    }
}
```