Formação Java Developer

BLOCO 1 | CURSO 1 | SEÇÃO 1 | AULA 1 – Introdução e objetivos

Principais aplicações Java de forma básica e inicial

Java fundamentos – Base para fundamentos para a linguagem

Apresentar a história e configurar o ambiente para uso Java

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

BLOCO 1 | CURSO 1 | SEÇÃO 1 | AULA 2 – O que é Java

A Linguagem Java é composta por um conjunto bem definido de funcionalidades (features).

Funcionalidades essenciais vão ser o foco inicial.

É uma linguagem simples e fácil de entender. É orientada a objetos. Significa que em Java tudo é escrito em termos de classes e Objetos.

POO – Programação Orientada a Objetos.

1 – Classe e Objeto

2 – Encapsulamento

3 – Abstração

4 – Herança

5 – Polimorfismo

Os programas Java são compilados para gerar arquivos de bytecode e a JVM (Java Virtual Machine) interpreta o arquivo de bytecode durante a execução.

BLOCO 1 | CURSO 1 | SEÇÃO 2 | AULA 1 – Java e sua História

Cafeteria em uma cidade chamada Java. Crescente e inovadora em 1993 aproveitando o crescimento da WWW.

Ambientes Java:

Páginas web | Aplicações corporativas | Softwares destinados ao consumidor final | Aplicativos para smartphones.

BLOCO 1 | CURSO 2 | SEÇÃO 1 | AULA 1 – Introdução a IDEs

Apresentação de Interfaces de Desenvolvimento Integrados.

- Eclipse

- IntelliJ

- Visual Studio Code

- NetBeans

IDE usada por mim VSCode.

BLOCO 1 | CURSO 2 | SEÇÃO 1 | AULA 2 – Configuração do Ambiente

Para começar a desenvolver em Java é necessário que tenhamos o JDK instalado.

O **JDK** (Java Development Kit) é um conjunto de ferramentas indispensáveis para o desenvolvimento de aplicações em **Java**. Ele é um ambiente de desenvolvimento que fornece todos os recursos necessários para escrever, compilar e executar programas Java.

**Principais componentes do JDK:**

1. **Compilador (javac)**: Transforma o código-fonte Java (arquivos .java) em bytecode (arquivos .class), que é interpretado pela JVM.
2. **Java Runtime Environment (JRE)**: Inclui a JVM (Java Virtual Machine) e bibliotecas essenciais para executar programas Java. O JRE também é parte do JDK.
3. **Ferramentas de Desenvolvimento**:
   * **Debugger (jdb)**: Ajuda a depurar código.
   * **JavaDoc**: Gera documentação em HTML a partir de comentários no código.
   * **Jar**: Permite criar arquivos compactados .jar para distribuir aplicações Java.
   * Outras ferramentas, como javap (para descompilar bytecode) e java (para executar aplicações Java).

**Uso do JDK**

* **Desenvolvimento**: É usado para criar aplicativos Java, tanto desktop, quanto para web, dispositivos móveis e outras plataformas.
* **Aprendizado**: Ferramenta essencial para quem está aprendendo a programar em Java, pois contém tudo que você precisa para compilar e testar código.

**Download e versões**

A Oracle disponibiliza o JDK gratuitamente no site oficial, mas também há implementações de terceiros, como o OpenJDK (uma versão open-source do JDK).

**Importância**

Sem o JDK, você não conseguiria desenvolver ou compilar programas em Java, pois ele contém o compilador e as ferramentas necessárias para transformar o código em algo executável pela JVM.

**Instalando o JDK**

JDK 23 x64 Installer instalada no meu PC.

Local instalado: C:\Program Files\jdk-23\

Texto

Descrição gerada automaticamente

BLOCO 1 | CURSO 2 | SEÇÃO 2 | AULA 1 – Instalando VSCode

Instalado o Java in Visual Studio Code

Instalado Extensões Java por meio do arquivo baixado (Coding Pack for Java)

Arquivo PrimeiroProjeto criado (Java)

Tela de vídeo game

Descrição gerada automaticamente

Fiz a configuração de tamanho de fonte em Open settings (configurações parte inferior esquerda) User e indo em font size.

Texto

Descrição gerada automaticamente

Texto

Descrição gerada automaticamente

Final de aula testei o arquivo App.java e escrevi nele o texto apresentado acima.

BLOCO 1 | CURSO 2 | SEÇÃO 2 | AULA 2 – Instalando outras IDEs

Foi mostrado como instalar outras IDEs para desenvolvimento Java

BLOCO 1 | CURSO 3 | SEÇÃO 1 | AULA 1 – Introdução

Apresentação do curso para informar a aplicação da sintaxe na linguagem Java

Vamos ver nesse módulo:

- Estrutura inicial;

- Padrão de nomenclatura;

- Declarando variáveis e métodos;

- Identação;

- Organizando os arquivos;

- Java Beans.

BLOCO 1 | CURSO 3 | SEÇÃO 2 | AULA 1 – Anatomia das classes 1

Public class NomeDaMinhaClasse {

// SEU CÓDIGO AQUI

}

Tela de computador

Descrição gerada automaticamente

Criada uma classe (MinhaClasse) para entender cada parte da sintaxe do Java

System.out.print – é o comando para mostrar algo na tela.

BLOCO 1 | CURSO 3 | SEÇÃO 2 | AULA 2 – Anatomia das classes - 2

Continuando a estrutura do Java

Quando se trata de escrever códigos na linguagem Java, é recomendado seguir algumas convenções de escrita.

- Arquivo.java: Todo arquivo.java deve começar com letra MAIÚCULA. Se a palavra for composta, a segunda palavra deve também ser maiúscula.

Ex.: Calculadora.java | CalculadoraCientifica.java

O nome da classe deve ser o mesmo nome do arquivo.

- Nome de variável: Toda variável deve ser escrita com letra minúscula, porém se a palavra for composta, a primeira letra da segunda palavra deverá ser maiúscula.

- Para tornar o valor de uma variável imutável, existe uma escrita especial que deixa característica:

Ex.: string br = “Brasil”

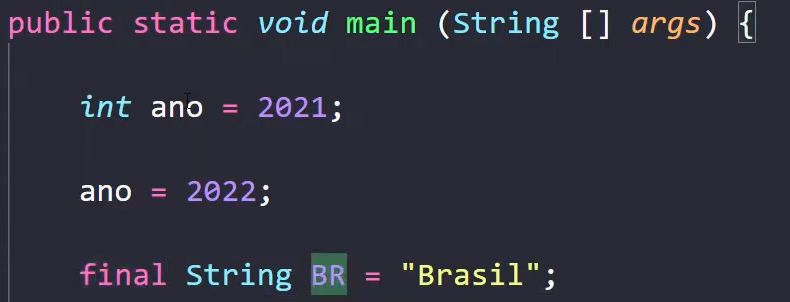
// para essa variável do tipo string a variável br recebeu o valor “Brail”

E pode sofrer alterações no valor com o decorrer do tempo de aplicação.

Mas,

Se escrevermos assim:

String BR = “Brasil” – a variável em maiúscula deixa ela sem poder sofrer alterações.



A palavra *final* faz essa função. Deixa a variável sem poder sofrer alterações.

Ou seja, usamos final e colocamos a variável em maiúscula

BLOCO 1 | CURSO 3 | SEÇÃO 2 | AULA 3 – Anatomia das classes - 3

Declarando variáveis e métodos

Como identificar entre declaração de variável e métodos em nosso programa? Existe uma estrutura comum para ambas as finalidades.

Declarar uma variável em Java segue sempre a seguinte estrutura:

Tipo NomeBemDefinido = Atribuição

Int idade = 23

Double altura = 1.62

Declarando métodos em Java segue uma estrutura bem simples:

TipoRetorno NomeObjetivoNoInfinitivo Parametros(s)

Int somar (int numeroUm, int numero2)

String formatarCep (long cep)

Public static String nomeCompleto (String primeiroNome, String segundoNome)

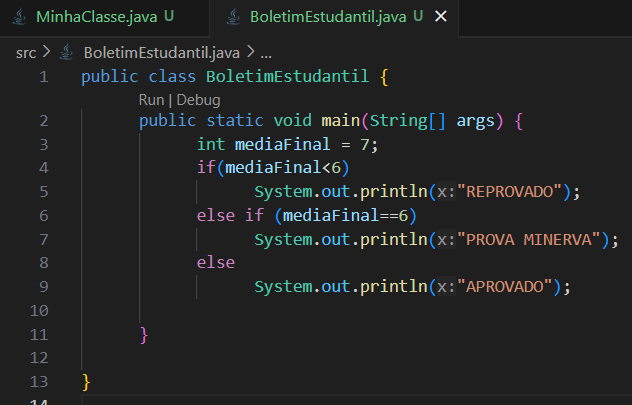
Texto

Descrição gerada automaticamente

BLOCO 1 | CURSO 3 | SEÇÃO 2 | AULA 4 – Anatomia das classes - 4

Identação

Basicamente indentar é um termo utilizado para escrever o código do programa de forma hierárquica, facilitando assim a visualização e o entendimento do programa.



BLOCO 1 | CURSO 3 | SEÇÃO 2 | AULA 5 – Anatomia das classes - 5

Organizando arquivos

À medida que nosso sistema vai evoluindo, surgem novos arquivos (código fonte) em nossa estrutura de arquivos do projeto. Isso exige que seja realizado uma organização destes arquivos através de pacotes (packages).

Tela de computador com letras e números em fundo preto

Descrição gerada automaticamente

Os pacotes são criados de forma de diretórios e subdiretórios de forma sequencial.

BLOCO 1 | CURSO 3 | SEÇÃO 2 | AULA 6 – Anatomia das classes - 6

Java Beans

Uma das maiores dificuldades na programação é escrever algoritmos legíveis a níveos que sejam compreendidos por todo seu time ou por você mesmo no futuro. Para isso a linguagem Java sugere, através de convenções, formas de escrita universal para nossas classes, atributos, métodos e pacotes.

BLOCO 1 | CURSO 3 | SEÇÃO 3 | AULA 1 – Tipos de variáveis - 1

Java Sintaxe

Tipos e variáveis

Explorar os tipos de dados.

- Numéricos inteiros e decimais

- Tipos lógicos

- Caracteres

- Objetos

- Diferença entre variável e constante.

BLOCO 1 | CURSO 3 | SEÇÃO 3 | AULA 2 – Tipos de variáveis - 2

Criando mais um repositório no GitHub para atualização da pasta.

Fiz atualização na pasta local e gerenciei com a pasta do GitHub

BLOCO 1 | CURSO 3 | SEÇÃO 3 | AULA 3 – Tipos de variáveis - 3

Tipos e variáveis

Tipos de dados

No Java, existem algumas palavras reservadas para a representação dos tipos de dados básicos que precisam ser manipulados para a construção de programas. Estes tipos de dados são conhecidos como tipos primitivos (Primitive Types).

Os oitos tipos primitivos em Java são:

Int | byte | short | long | float | double | boolean | char

Esses tipos não são considerados objetos, e, portanto, representam valores brutos. Eles são armazenados diretamente na pilha de memória.

Byte | int | short | long são variáveis do tipo INTEIRO.

Declaração de variáveis

Variável é uma identificação de um espaço em memória utilizado pelo nosso programa. Seguindo as convenções em linguagem de programação, toda variável é composta por: tipos de dados + identificação + valor atribuído.

Obs.: Existem algumas peculiaridades a trabalhar com alguns tipos específicos.

Quando usamos float ou long colocamos F e L no final das variáveis respectivamente. Ou seja, o tipo long precisa terminar com L, o tipo float precisa terminar com F e alguns cenários do dia-a-dia podem estimular uma alteração de tipos de dados convencional.

BLOCO 1 | CURSO 3 | SEÇÃO 3 | AULA 4 – Tipos de variáveis - 4

Constantes

São valores armazenados em memória que não podem ser modificados depois de declarados. Em Java, esses valores são representados pela palavra reservada final, seguida do tipo.

Por convenção, constantes são sempre escritas em CAIXA ALTA.

Para transformar uma variável em constante temos que seguir a estrutura:

*Final* tipo nome = valor

BLOCO 1 | CURSO 3 | SEÇÃO 4 | AULA 1 – Operadores – 1

Tipos de operadores

Objetivo dessa seção: Apresentar o significado dos símbolos especiais de acordo com a linguagem de programação Java.

- Operadores aritméticos

- Operadores unários

- Operadores relacionais

- Operadores lógicos

- Operador ternário

BLOCO 1 | CURSO 3 | SEÇÃO 4 | AULA 2 – Operadores – 2

Operadores

Símbolos especiais que tem um significado próprio para a linguagem e estão associados a determinadas operações.

Atribuição ( = )

É utilizado para definir o valor inicial ou sobrescrever o valor de uma variável. Em java, definimos um tipo, nome e opcionalmente atribuímos um valor à variável através do operador de atribuição.

Ex.:

String nome = “Leonardo”;

Int idade = 22;

Double peso = 68.5;

Char sexo = ‘M’;

Boolean doadorOrgao = false;

Date dataNascimento = new Date(). (aqui é uma classe que é usada como uma variável e usando a palavra new)

Aritméticos

Utilizado para realizar operações matemáticas entre valores numéricos, podendo se tornar ou não uma expressão mais complexa.

Adição | Subtração | Multiplicação | Divisão

Obs.: O operador de adição (+), quando utilizado em variáveis do tipo texto, realizará a concatenação de textos.

Na imagem abaixo foi feita uma simulação de debug utilizando ponto de parada (breakpoint)

Texto

Descrição gerada automaticamente

BLOCO 1 | CURSO 3 | SEÇÃO 4 | AULA 3 – Operadores – 3

Unários

Esses operadores são aplicados juntamente com um outro operador aritmético. Eles realizam alguns trabalhos básicos como incrementar, decrementar, inverter valores numéricos e booleanos.

( + ) Operador unário de valor positivo

( - ) Operador unário de valor negativo

(++) Operador unário de incremento de valor

(--) Operador unário de decremento de valor

( ! ) Operador unário de negação