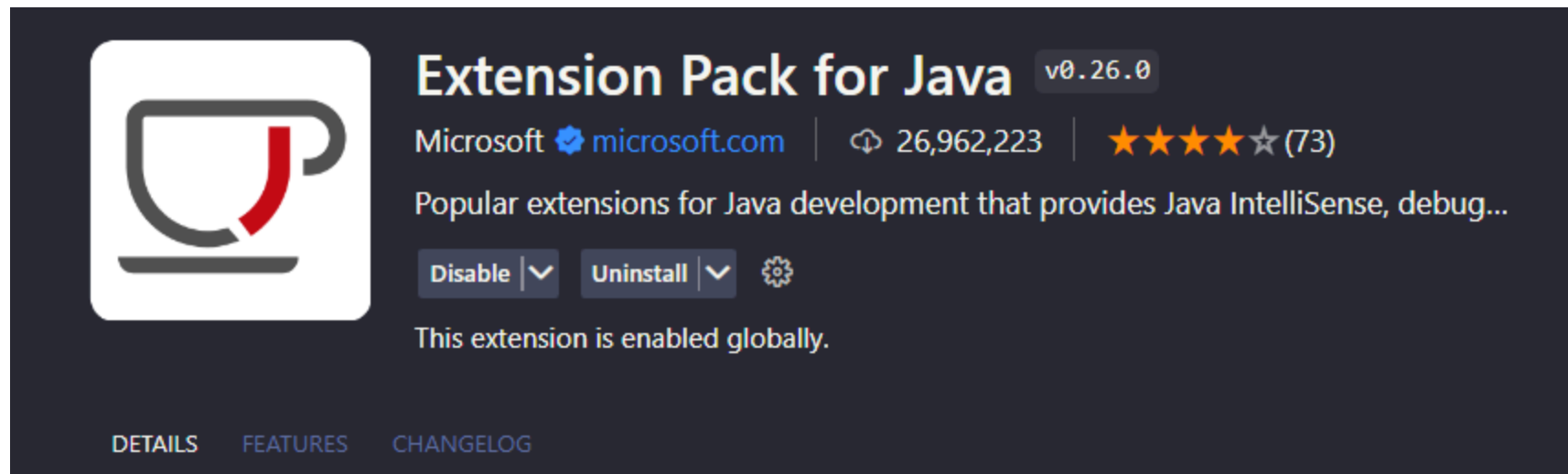




Material Básico das primeiras aula de java.

Como configurar o vsCode

1. Após instalar o JDK e configurar as variáveis de ambiente, instale a seguinte extensão para o vsCode.



2. Crie um projeto Java:

- i. Aperte a tecla "F1" (ou FN + F1) do seu teclado;
- ii. Digite: "create java project", pressione a tecla "enter";
- iii. Escolha a primeira opção, "no build tools";
- iv. Selecione a pasta;
- v. Crie um nome para o projeto e pressione a tecla "enter";

Tipos de dados mais comuns em Java

- Em java as variáveis são declaradas com a seguinte nomenclatura:
 - *Tipo nome*
- Existem vários *tipos* primitivos de dados na linguagem, os mais comumente utilizados são:

```
int idade = 20; //variável do tipo inteiro
float saldoBancario = 0.50 // variável do tipo racional (ponto flutuante)
double valorDoImovel = 2345.500 // mesma lógica do float, porém com mais casas decimais.
String meuNome = "Astolfo" //variável utilizada para guardar cadeias de caracteres
boolean isJavaFun = true; // variável do de tipo lógico, verdadeiro ou falso (true or false)
char sexo = 'M'// variável que pode conter apenas um caractere.
```

entrada de dados simples em java

1. Em java para que possamos inserir dados simples através do teclado, utilizamos um dado personalizado, chamado "Scanner".

```
Scanner entradaDeDados = new Scanner(System.in);
```

2. Existem várias entradas de dados, entre elas temos:

```
// Entrada de dados para tipos inteiros
System.out.print("Inteiro: ");
int idade = entradaDeDados.nextInt();
// Tipos racionais
System.out.print("Float: ");
float saldoBancario = entradaDeDados.nextFloat();
System.out.print("Double: ");
double valorDoImovel = entradaDeDados.nextDouble();
// Tipos cadeia de caracteres (string)
System.out.println("String: ");
String meuNome = entradaDeDados.next();
//tipos lógicos
System.out.print("Lógico: ");
boolean isJavaCool = entradaDeDados.nextBoolean();
```

Operações matemáticas básicas em java

1. Dentre as operações básicas, possuímos:

```
int numero1 = 10;
int numero2 = 20;
// Soma
System.out.println(numero1 + " + " + numero2 + " = " + (numero1 + numero2));
// Subtração
System.out.println(numero1 + " - " + numero2 + " = " + (numero1 - numero2));
// Multiplicação
System.out.println(numero1 + " * " + numero2 + " = " + (numero1 * numero2));
// Divisão
System.out.println(numero1 + " / " + numero2 + " = " + (numero1 / numero2));
// Mod ou resto da divisão
System.out.println(numero1 + " % " + numero2 + " = " + (numero1 % numero2));
```

Controle de fluxo

Em um determinado ponto do programa, precisamos realizar algumas decisões que irão impactar diretamente ou não, no resultado em que visamos alcançar, para que possamos controlar o fluxo de dados em nossas aplicações, normalmente, utilizamos estruturas de decisão, em java, elas são:

```
if(condicao) {  
    // Ação, caso a condição seja atendida, as próximas possibilidades não  
    // serão executadas  
} else if(outraCondicao) {  
    // caso a primeira não seja atendida, essa é a próxima tentativa  
} else {  
    // caso nenhuma condição seja suprida, essa será cumprida de qualquer jeito.  
}
```


Relações de comparação

Para que possamos realizar decisões nas estruturas "if", utilizamos a seguinte nomenclatura:

```
int idade1 = 20;
int idade2 = 10;

if(idade1 > idade2){ // se idade1 for MAIOR que idade2, faça tal coisa
}else if(idade1 >= idade2){// se idade1 for MAIOR OU IGUAL a idade2, faça tal coisa
}else if(idade1 < idade2){ // se idade01 for MENOR que idade2, faça tal coisa
}else if(idade1 <= idade2){// se idade01 for MENOR OU IGUAL a idade2, faça tal coisa
}else if(idade1 == idade2){// se idade01 EQUIVALE a idade2, faça tal coisa
}else{ //caso nenhuma condição acima seja cumprida, faça tal coisa
}
```

Operadores lógicos

Em Java, temos os seguintes operadores lógicos:

- "E" lógico (&&):
 - Retorna true se ambas as expressões forem verdadeiras.
 - Exemplo:

```
boolean expressao1 = true;  
boolean expressao2 = false;  
boolean resultado = expressao1 && expressao2;  
System.out.println("Resultado do E lógico: " + resultado); // Saída: false
```

- "OU" lógico (||):
 - Retorna true se pelo menos uma das expressões for verdadeira.
 - Exemplo:

```
boolean expressao3 = true;  
boolean expressao4 = false;  
boolean resultado2 = expressao3 || expressao4;  
System.out.println("Resultado do OU lógico: " + resultado2); // Saída: true
```

- NÃO lógico (!)
 - Inverte o valor da expressão (se era true, torna-se false, e vice-versa).
 - Exemplo:

```
boolean expressao5 = true;  
boolean resultado3 = !expressao5;  
System.out.println("Resultado do NÃO lógico: " + resultado3); // Saída: false
```

Estrutura de controle switch/case

- switch/case em Java é uma estrutura de controle que nos permite testar várias condições e executar diferentes blocos de código dependendo do valor da expressão.
- Sua sintaxe básica é a seguinte:

```
switch (expressao) {  
    case valor1:  
        // código a ser executado se expressao == valor1  
        break;  
    case valor2:  
        // código a ser executado se expressao == valor2  
        break;  
    // ...  
    default:  
        // código a ser executado se nenhum dos casos anteriores for verdadeiro  
}  

```

Características do switch/case

- A expressão é avaliada uma vez.
- O valor da expressão é comparado com os valores de cada caso.
- Se houver correspondência, o bloco de código associado ao caso é executado.
- A palavra-chave break é usada para sair do bloco switch.
- O bloco default é opcional e é executado se nenhum dos casos corresponder.

- Neste exemplo, se o dia for escolhido para 4, a saída será "Quinta-feira". Se dia escolhido for 10 (ou qualquer outro valor não listado), a saída será "Dia inválido".

```
int dia = 4;
switch (dia) {
    case 1:
        System.out.println("Segunda-feira");
        break;
    case 2:
        System.out.println("Terça-feira");
        break;
    case 3:
        System.out.println("Quarta-feira");
        break;
    case 4:
        System.out.println("Quinta-feira");
        break;
    case 5:
        System.out.println("Sexta-feira");
        break;
    case 6:
        System.out.println("Sábado");
        break;
    case 7:
        System.out.println("Domingo");
        break;
    default:
        System.out.println("Dia inválido");
}
```

FIM :)