Introdução

Nesse módulo, entendemos quais os tipos de variáveis temos disponíveis no Java. As variáveis armazenam dados e esses dados podem ser primitivos ou encapsulados em uma classe (*variáveis wrapper*). Vamos entender mais sobre o assunto?

Tipos de dados

O Java é historicamente uma linguagem de programação estaticamente tipada, ou seja, precisamos informar o tipo de dado que uma variável conterá. No entanto, desde a versão 10 da linguagem ela ganhou suporte à inferência de tipos, permitindo que o programador não declare o tipo na inicialização da variável. Isso pode ser feito usando o tipo especial "var":

```
String nome = "Lucas";
var sobrenome = "Santos";
```

Repare que o tipo da variável sobrenome será inferido para "String" pelo compilador.

Tipos primitivos

Tipos primitivos são tipos de dados cujos valores são armazenados diretamente em memória.
Uma característica deles é que as variáveis que são de um tipo primitivo não podem ser inicializadas como nulas.
A linguagem Java possui 8 tipos primitivos:

- byte
- short
- int
- long
- float
- double
- boolean
- char

É possível aprender mais sobre os tipos primitivos do Java nessa documentação:

https://docs.oracle.com/javase/tutorial/java/nutsandbolts/datatypes.html

Tipos wrapper

Cada um dos tipos primitivos possui uma classe dedicada para eles e essas classes são chamadas de "wrapper" (ou invólucros).

Essas classes wrapper possuem dois principais objetivos:

- Encapsular tipos primitivos em objetos para permitir realizar operações que são frequentes no Java, como criar Listas (ArrayList), HashMaps, serializar, etc.:
- Prover métodos utilitários para tipos primitivos como conversão entre tipos, comparação de valores, ordenação, etc. Como exemplo, o tipo wrapper "Integer" permite converter valores inteiros para "float" de forma muito fácil:

```
Integer temperatura = 25;
temperatura.floatValue();
```

Expressões Regulares

Expressões regulares, também conhecido como **Regex**, são uma forma concisa e rápida de encontrar padrões em um texto.
São úteis para **validar** a entrada de dados dos usuários de um programa, como **emails**, **CPFs**, **RGs**, etc...

Existem ferramentas online onde podemos testar e compreender qualquer expressão regular, como o **RegExr** (https://regexr.com/) e o Regex101 (https://regex101.com/) .

Essas ferramentas também possuem bibliotecas de expressões regulares criadas e compartilhadas por outros usuários.

O exemplo ao lado é um programa que solicita o endereço de email do usuário e o valida:

```
import java.util.Scanner;
import java.util.regex.Matcher;
import java.util.regex.Pattern;

public class Main {
    public static void main(String[] args) {
        String emailRegex =
"[a-z0-9!#$%&'*+/=?^_`{|}~-]+(?:\\.[a-z0-9!#$%&'*+/=?^_`{|}~-]
+)*@(?:[a-z0-9](?:[a-z0-9-]*[a-z0-9])?\\.)+[a-z0-9](?:[a-z0-9-
]*[a-z0-9])?";
        Scanner scanner =  new Scanner(System.in);

        System.out.print("Informe seu email: ");
        String email = scanner.nextLine();

        Pattern pattern = Pattern.compile(emailRegex);
        Matcher matcher = pattern.matcher(email);

        if (matcher.find()) {
            System.out.println("Conta criada com sucesso!");
        } else {
            System.out.println("Endereço de email não é
válido");
        }
    }
}
```