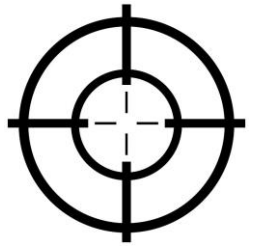
Atividades de fixação

Fixando os conceitos principais da POO

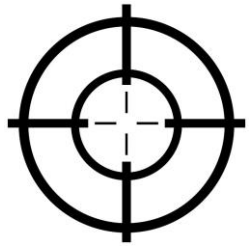


Para aprender, adequadamente, a programação orientada à objetos é preciso treinar!

O exercício proposto ajudará ao desenvolvedor iniciante praticar a criação de objetos, o conceito de herança, o conceito de encapsulamento e o conceito de polimorfismo.

O exercício está separado em duas partes de objetivos.

Programando uma calculadora especial

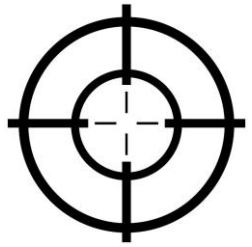


Considere que você foi contratado para desenvolver uma calculadora, que funcione como um aplicativo do tipo cli (command-line interface), feito em TypeScript.

A calculadora deve realizar as operações básicas, como soma, subtração, divisão, potência, etc...

Cada operação deve ser realizada por uma classe específica. Veja no exemplo!

Exemplos...

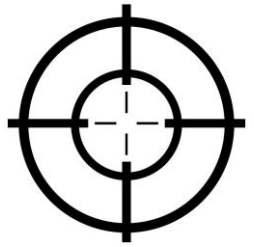


```
export default abstract class Calculo {
    public abstract calcular(numero1: number, numero2: number): number;
}
```

```
export default class Soma extends Calculo {
    public calcular(numero1: number, numero2: number): number {
        return numero1 + numero2
    }
}
```

```
export default class Multiplicacao extends Calculo {
    public calcular(numero1: number, numero2: number): number {
        return numero1 * numero2
    }
}
```

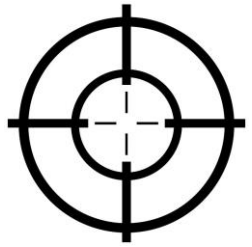
A calculadora...



```
let mensagens = new Mensagens()

let iniciar = () => {
  let leitor = readline.createInterface({
    input: process.stdin,
    output: process.stdout
  });

  leitor.question(`Quais são seus números e a operação desejada?\n`, (valor) => {
    let instrucoes = valor.split(' ')
    let numero1 = Number(instrucoes[0])
    let numero2 = Number(instrucoes[1])
    let operacao = instrucoes[2]
    if(instrucoes.length == 1){
      operacao = instrucoes[0]
    }
    console.log(`Estas foram suas instruções: ${instrucoes}\n`)

    switch (operacao) {
      case 'Somar':
        let calculo = new Soma()
        console.log(`O resultado da operação é: ${calculo.calcular(numero1, numero2)}\n`)
        break;
      case 'Subtrair':
        calculo = new Subtracao()
        console.log(`O resultado da operação é: ${calculo.calcular(numero1, numero2)}\n`)
        break;
      case 'Multiplicar':
        calculo = new Multiplicacao()
        console.log(`O resultado da operação é: ${calculo.calcular(numero1, numero2)}\n`)
        break;
      case 'Sair':
```

```
import * as readline from 'readline';
import Mensagens from './mensagens';
import Multiplicacao from './multiplicacao';
import Soma from './soma';
import Subtracao from './subtracao';
```

Operações...

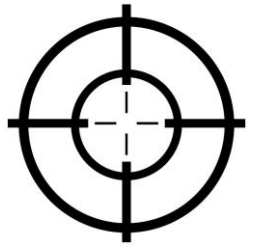


Cada operação ocorrerá sempre sobre dois números, que serão fornecidos pelo usuário.

```
switch (operacao) {
  case 'Somar':
    let calculo = new Soma()
    console.log(`O resultado da operação é: ${calculo.calcular(numero1, numero2)}\n`)
    break;
  case 'Subtrair':
    calculo = new Subtracao()
    console.log(`O resultado da operação é: ${calculo.calcular(numero1, numero2)}\n`)
    break;
  case 'Multiplicar':
    calculo = new Multiplicacao()
    console.log(`O resultado da operação é: ${calculo.calcular(numero1, numero2)}\n`)
    break;
```

O usuário poderá escolher quais operações ele quer fazer, sobre os dois números.

Um processo para cada operação



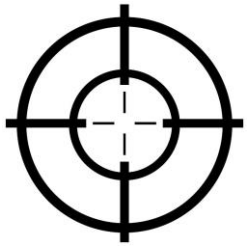
Cada operação deverá ser feita por uma classe, utilizando o polimorfismo.

```
export default class Soma extends Calculo {
    public calcular(numero1: number, numero2: number): number {
        return numero1 + numero2
    }
}
```

```
export default class Multiplicacao extends Calculo {
    public calcular(numero1: number, numero2: number): number {
        return numero1 * numero2
    }
}
```

```
switch (operacao) {
  case 'Somar':
    let calculo = new Soma()
    console.log(`O resultado da operação é: ${calculo.calcular(numero1, numero2)}\n`)
    break;
  case 'Subtrair':
    calculo = new Subtracao()
    console.log(`O resultado da operação é: ${calculo.calcular(numero1, numero2)}\n`)
    break;
  case 'Multiplicar':
    calculo = new Multiplicacao()
    console.log(`O resultado da operação é: ${calculo.calcular(numero1, numero2)}\n`)
    break;
```

Objetivos, parte 1

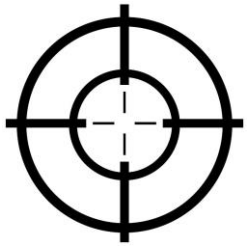


A primeira parte de objetivos é o desenvolvimento das operações básicas.

A calculadora deve permitir os cálculos de soma, subtração, divisão, potenciação e radiciação.

Os cálculos devem sempre utilizar dois números!

Objetivos, parte 2



Construa também, a opção de calcular as raízes de uma função do segundo grau, utilizando a fórmula de bhaskara. Neste caso, o usuário poderá informar mais que dois número.

Biblioteca para receber dados do usuário...

```
import * as readline from 'readline';
import Mensagens from './mensagens';
import Multiplicacao from './multiplicacao';
import Soma from './soma';
import Subtracao from './subtracao';
```

Módulo readline, do Node.js, permite o recebimento de um fluxo de dados, inseridos pela linha de comando. Por isso, o módulo readline facilita a entrada ou leitura de dados fornecidos pelo usuário.

https://nodejs.org/api/readline.html

Projeto de auxílio