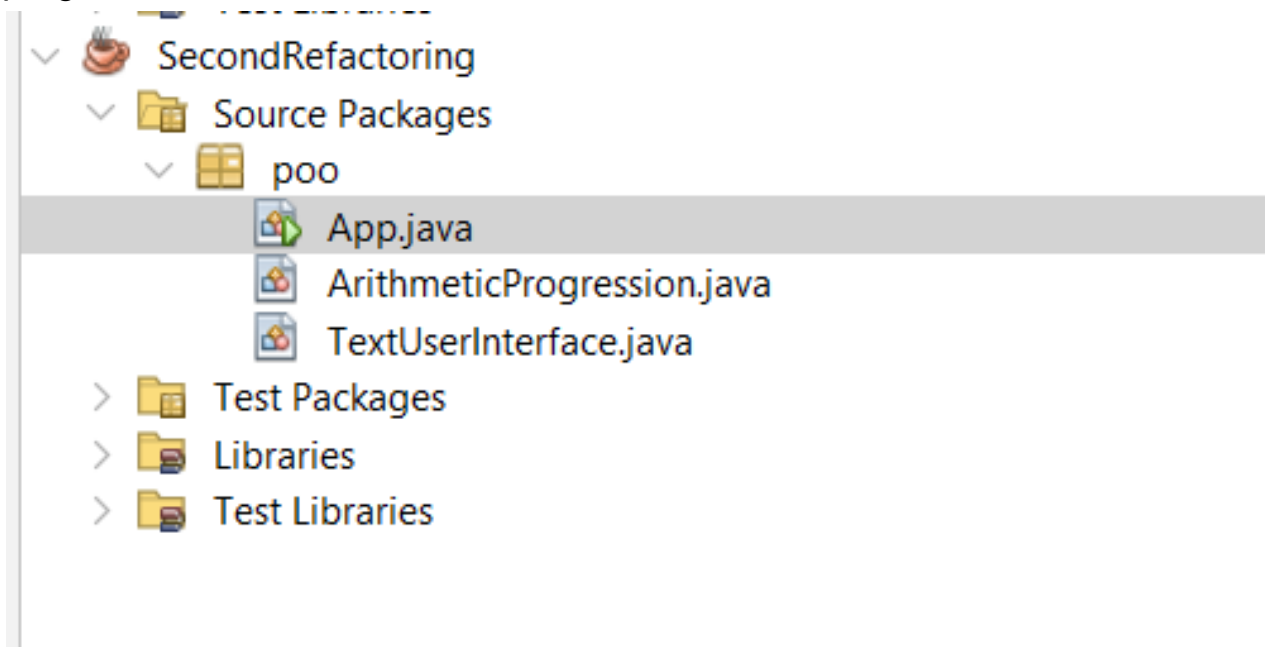


Nome: Lucas Ertel Soares Ra: 2096960

Refatoração código progressão aritmética para o paradigma de orientação a objetos

1- Aqui está a separação por pacote onde tem 3 classes sendo a App responsável pela execução da aplicação , a classe TextUserInterface responsável pela entrada e leituras do primeiro elemento a razão da progressão aritmética e número de termos para realizar a soma e a listagem desses termos com sua somas.e a classe ArithmeticProgression responsável por realizar os cálculos da soma progressão .



[Clique Aqui para ver a sequência dos pacotes no github](#)

1-Classe App (**Use as teclas ctrl + para aproximar para ler o código**) [Click aqui para ver a classe no github](#)

```
public class App {  
  
    public void go() {  
        TextUserInterface txtUserInterface = new TextUserInterface();  
  
        int numberOfterms = txtUserInterface.readNumberOfTerms();  
  
        int firstterm = txtUserInterface.readFirstTerm();  
  
        int commonDifference = txtUserInterface.readReason();  
  
        ArithmeticProgression pol = new ArithmeticProgression(firstTerm: firstterm, commonDifference);  
  
        txtUserInterface.printAllTerms(pa: pol, numberOfTerms: numberOfterms);  
  
        int sum = pol.sumOfAllTerms(numberOfTerms: numberOfterms);  
  
        System.out.println("A soma da pa1 é: " + sum);  
  
        numberOfterms = txtUserInterface.readNumberOfTerms();  
  
        firstterm = txtUserInterface.readFirstTerm();  
  
        commonDifference = txtUserInterface.readReason();  
  
        ArithmeticProgression po2 = new ArithmeticProgression(firstTerm: firstterm, commonDifference);  
  
        txtUserInterface.printAllTerms(pa: pol, numberOfTerms: numberOfterms);  
  
        sum = po2.sumOfAllTerms(numberOfTerms: numberOfterms);  
  
        System.out.println("A soma da pa2 é: " + sum);  
    }  
  
    public static void main(String[] args) {  
        App app = new App();  
        app.go();  
    }  
}
```

2- TextUserInterface (**Use as teclas ctrl + para aproximar para ler o código**) [Click aqui para ver a classe no github](#)

```

package poo;

import java.util.Scanner;

public class TextUserInterface {

    Scanner console = new Scanner(System.in);

    public TextUserInterface() {
    }

    //-----
    public int readAllTerms(String msg , int numbermin, int numbermax){
        do
        {
            System.out.println(x: msg);
            numbermax = console.nextInt();
        } while(numbermax < numbermin);

        return numbermax;
    }

    public int readNumberOfTerms ()
    {
        return readAllTerms(msg:"Digite o número de termos da PA:", numbermin: 2, numbermax: Integer.MAX_VALUE);
    }
}

```

```

//-----
public int readFirstTerm()
{
    return readAllTerms(msg:"Digite o primeiro termo da PA:", numbermin: 1, numbermax: Integer.MAX_VALUE);
}

//-----
public int readReason()
{
    return readAllTerms(msg:"Digite a razao da PA:", numbermin: 1, numbermax: Integer.MAX_VALUE);
}

public void printAllTerms (ArithmeticProgression pa, int numberOfTerms)
{
    for(int i=1; i<=numberOfTerms; i++)
    {
        int ithTerm = pa.termOfIndex(index: i);
        System.out.printf(format: "a%d=%d\n", args: i, args: ithTerm);
    }
}
}

```

3 Classe ArithmeticProgression (**Use as teclas ctrl + para aproximar para ler o código**)

[Click aqui para ver a classe no github](#)

```
package poo;

public class ArithmeticProgression {

    private int firstTerm;
    private int commonDifference;

    //-----

    public ArithmeticProgression(int firstTerm, int commonDifference) {
        this.firstTerm = firstTerm;
        this.commonDifference = commonDifference;
    }

    public int termOfIndex(int index)
    {
        return firstTerm + (index-1) * commonDifference;
    }

    //-----

    public int sumOfAllTerms(int numberOfTerms)
    {
        int lastTerm = termOfIndex(index: numberOfTerms);

        return ( (firstTerm + lastTerm) * numberOfTerms) / 2);
    }

}
```