

# Lista Avaliativa 1

## Instruções:

Resolva as questões abaixo e mande sua resolução no teams anexada na tarefa no formato de sua preferência. Pode ser em pdf, txt, zip, como um notebook jupyter, ou qualquer outra coisa que você preferir.

Entregas até o dia da prova (17/04). Não serão aceitas entregas após o prazo, fiquem atentos.

Podem usar qualquer uma das linguagens que vimos em sala nas últimas duas questões, desde que se encaixem no paradigma.

Linguagens que vimos: BASIC, Lua, Rust, Go, Python, C#, Scala, Javascript. OBS: Não precisar usar no paradigma que vimos em sala, pode usar Python como imperativa por exemplo.

## Questão 1

```
1 ~ fn is_something(num: i32) -> bool {
2 ~     if num < 2 {
3 ~         return false;
4 ~     }
5 ~     for i in 2..num {
6 ~         if num % i == 0 {
7 ~             return false;
8 ~         }
9 ~     }
10 ~     return true;
11 ~ }
12
13 ~ fn main() {
14 ~     let mut array: [i32; 9] = [1, 3, 5, 7, 9, 10, 11, 13, 15];
15 ~     let mut sum = 0;
16 ~     for i in 0..array.len() {
17 ~         if is_something(array[i]) {
18 ~             sum += array[i];
19 ~         }
20 ~     }
21 ~     println!("Sum of something: {}", sum);
22 ~ }
```

Marque V ou F para as seguintes afirmativas sobre esse código. Justifique ou corrija as falsas.

( ) Esse código está escrito em Go.

( ) O código usa um laço de repetição para varrer (iterar) os elementos do `array`.

( ) A variável `sum` no `main` não precisa ser declarada como mutável, pois não é modificada fora do loop.

( ) Um nome melhor para a função `is_something()` seria `is_prime()` [é primo]

( ) A variável `array` não precisa ser declarada com `mut`, pois o valor dela não se altera.

Respectiva sequência: F – V – F – V – V – V

1º alternativa – justificativa: O código acima está escrito com a linguagem Rust.

3º alternativa – justificativa: A variável “sum” precisa ser declarada como mutável, pois toda vez que entra em loop, o seu valor se altera (vai somando os números primos).

## Questão 2

```
1  using System;
2
3  public class Person {
4      private string Name;
5      private int Age;
6
7      public Person(string name, int age) {
8          Name = name;
9          Age = age;
10     }
11
12     public virtual void PrintInfo() {
13         Console.Write("Name: " + Name);
14         Console.WriteLine(", Age: " + Age);
15     }
16 }
17
18
19 public class Student : Person {
20     private int StudentId;
21
22     public Student(string name, int age, int studentId) : base(name, age) {
23         StudentId = studentId;
24     }
25
26
27     public override void PrintInfo() {
28         base.PrintInfo();
29         Console.WriteLine("Student ID: " + StudentId);
30     }
31 }
32
33
34 class Program {
35     static void Main(string[] args) {
36         Person person = new Person("John", 25);
37         Student student = new Student("Jane", 20, 12345);
38
39         person.PrintInfo();
40         student.PrintInfo();
41     }
42 }
```

A respeito da linguagem C#, marque V ou F para as seguintes afirmações: (Justifique ou corrija as falsas)

- ( ) A classe `Student` herda da classe `Person` e adiciona um campo adicional para o ID do aluno.
- ( ) A classe `Person` tem um construtor que recebe um nome e uma idade e inicializa as propriedades `Name` e `Age` da classe.
- ( ) A classe `Student` substitui completamente o método `PrintInfo()` da classe `Person` em vez de estendê-lo.
- ( ) A palavra-chave `virtual`, na declaração do método `PrintInfo()` na classe `Person`, indica que ele não pode ser chamado.
- ( ) A saída desse código será:  
Name: John, Age: 25  
Name: Jane, Age: 20  
Student ID: 12345

Respectiva sequência: V – V – V – F – V – V

3º alternativa – justificativa: O método “PrintInfo” da classe `Student` que substitui o método “PrintInfo” da classe `Person`, mas sim dava para fazer em um método só.

4º alternativa – justificativa: O modificador “virtual” significa que o método pode ser sobrescrito/substituído por outro método na subclasse (no caso da classe `Student`).

## Questão 3

Crie um programa para calcular áreas de diferentes formas geométricas. O programa deve permitir ao usuário escolher a forma geométrica desejada (retângulo, triângulo ou círculo), e solicitar os parâmetros necessários para o cálculo da área.

- O programa deve verificar se os valores de entrada para o cálculo da área são números válidos (ou seja, não nulos e positivos).
- O programa deve calcular a área e imprimir o resultado no console.

O programa deve continuar solicitando a entrada do usuário até que ele digite uma opção válida para a forma geométrica, ou de alguma forma encerre a execução.

Considere que o valor de  $\pi$  é 3,14 para o cálculo da área do círculo.

## Questão 4

- Crie uma classe `Televisão` que receba como parâmetro seu modelo.
- Dentro dessa classe, crie atributos encapsulados de volume e canal. (O volume começará em 0)
- Crie um método que aumenta o volume ( `aumentar_volume` ) e outro que diminua ( `diminuir_volume` ) ele. Ambos receberam como parâmetro a quantidade que deverá ser aumentada ou diminuída da quantidade total.
- Crie um método para trocar de canal ( `trocar_canal` ). Esse método receberá como parâmetro um novo canal.
- Crie um método que imprima o modelo, junto com o volume atual e o canal atual.
- Também crie uma classe `SmartTV`, que herda de `Televisão` e que tem o atributo `internet` e um método `conectar_internet()`, que possa alterar a indicação de conexão a internet.
- Por fim, chame os métodos de aumentar e diminuir com os valores 70 e 27, respectivamente, e troque o canal para 'Canal #1', e então chame o método para imprimir tudo. Também teste a implementação de um objeto da classe `SmartTV`