**PARTE 1 – PROGRAMAÇÃO ESTRUTURADA.**

**Lista 14 de exercícios.**

**Instruções.**

As listas de exercícios desenvolvidas com ferramentas de inteligência artificial, e visa ajudá-lo na prática da programação. Você deve tentar fazer os exercícios sem olhar na solução, porém no início é muito natural olhar na solução para entender as propostas.

Sugestão para cada lista crie uma solução com o nome da Lista. Por exemplo, SolucaoLista1 e depois crie um projeto com o nome do exercício e da lista, por exemplo, L01EX01.

Seria interessante que após fazer o exercício, porta-se o mesmo para uma DLL, e chama-se a classe da dll no projeto principal.

Sumário

[**Lista de Exercícios: Usando a Instrução while em C#** 3](#_Toc160389632)

[1. Exercício 1: Contagem até 10 3](#_Toc160389633)

[2. Exercício 2: Contagem Regressiva 3](#_Toc160389634)

[3. Exercício 3: Soma dos Primeiros 100 Números 3](#_Toc160389635)

[4. Exercício 4: Exibir Números Pares até 20 3](#_Toc160389636)

[5. Exercício 5: Tabela de Multiplicação 3](#_Toc160389637)

[6. Exercício 6: Fatorial de um Número 3](#_Toc160389638)

[7. Exercício 7: Soma de Entradas do Usuário 3](#_Toc160389639)

[8. Exercício 8: Lista de Quadrados 3](#_Toc160389640)

[9. Exercício 9: Contagem de Letras 3](#_Toc160389641)

[10. Exercício 10: Inversão de String 3](#_Toc160389642)

[**Soluções dos Exercícios sobre Instrução while** 4](#_Toc160389643)

[1. Solução do Exercício 1 4](#_Toc160389644)

[2. Solução do Exercício 2 4](#_Toc160389645)

[3. Solução do Exercício 3 4](#_Toc160389646)

[4. Solução do Exercício 4 4](#_Toc160389647)

[5. Solução do Exercício 5 4](#_Toc160389648)

[6. Solução do Exercício 6 4](#_Toc160389649)

[7. Solução do Exercício 7 4](#_Toc160389650)

[8. Solução do Exercício 8 5](#_Toc160389651)

[9. Solução do Exercício 9 5](#_Toc160389652)

[10. Solução do Exercício 10 5](#_Toc160389653)

**Lista de Exercícios: Construtores Polimórficos em C#**

Exercício 1: Classe **Lampada**

* **Tarefa**: Crie uma classe **Lampada** com os atributos **potencia** (int) e **estado** (bool). Implemente dois construtores: um que inicialize ambos os atributos e outro que inicialize apenas a **potencia**, definindo **estado** como **false** por padrão.

Exercício 2: Classe **Conta**

* **Tarefa**: Desenvolva uma classe **Conta** com os atributos **titular** (string) e **saldo** (double). Forneça um construtor que inicialize o **titular** com um valor e defina o **saldo** inicial como zero, e outro construtor que inicialize ambos os atributos.

Exercício 3: Classe **Estudante**

* **Tarefa**: Implemente uma classe **Estudante** com **nome** (string), **matricula** (string) e **curso** (string). Crie um construtor para inicializar **nome** e **matricula**, e outro para inicializar todos os três atributos.

Exercício 4: Classe **Pedido**

* **Tarefa**: Elabore uma classe **Pedido** contendo os atributos **numero** (int), **descricao** (string) e **quantidade** (int). Defina um construtor para inicializar **numero** e **descricao**, e outro para inicializar todos os atributos.

Exercício 5: Classe **Circulo**

* **Tarefa**: Projete uma classe **Circulo** com os atributos **raio** (double) e **cor** (string). Forneça um construtor que inicialize apenas o **raio** e outro que inicialize tanto o **raio** quanto a **cor**.

**Soluções dos Exercícios de Construtores Polimórficos**

Solução do Exercício 1: Classe **Lampada**

* public class Lampada { public int Potencia { get; set; } public bool Estado { get; set; } public Lampada(int potencia, bool estado) { Potencia = potencia; Estado = estado; } public Lampada(int potencia) : this(potencia, false) { } }

Solução do Exercício 2: Classe **Conta**

* public class Conta { public string Titular { get; set; } public double Saldo { get; set; } public Conta(string titular) { Titular = titular; Saldo = 0; } public Conta(string titular, double saldo) { Titular = titular; Saldo = saldo; } }

Solução do Exercício 3: Classe **Estudante**

* public class Estudante { public string Nome { get; set; } public string Matricula { get; set; } public string Curso { get; set; } public Estudante(string nome, string matricula) { Nome = nome; Matricula = matricula; } public Estudante(string nome, string matricula, string curso) : this(nome, matricula) { Curso = curso; } }

Solução do Exercício 4: Classe **Pedido**

* public class Pedido { public int Numero { get; set; } public string Descricao { get; set; } public int Quantidade { get; set; } public Pedido(int numero, string descricao) { Numero = numero; Descricao = descricao; Quantidade = 0; // Default value } public Pedido(int numero, string descricao, int quantidade) : this(numero, descricao) { Quantidade = quantidade; } }

Solução do Exercício 5: Classe **Circulo**

* public class Circulo { public double Raio { get; set; } public string Cor { get; set; } public Circulo(double raio) { Raio = raio; } public Circulo(double raio, string cor) : this(raio) { Cor = cor; } }