PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS

Programação Orientada a Objetos (POO)

Lista 02

Aluno: Athos Martinez Andrade

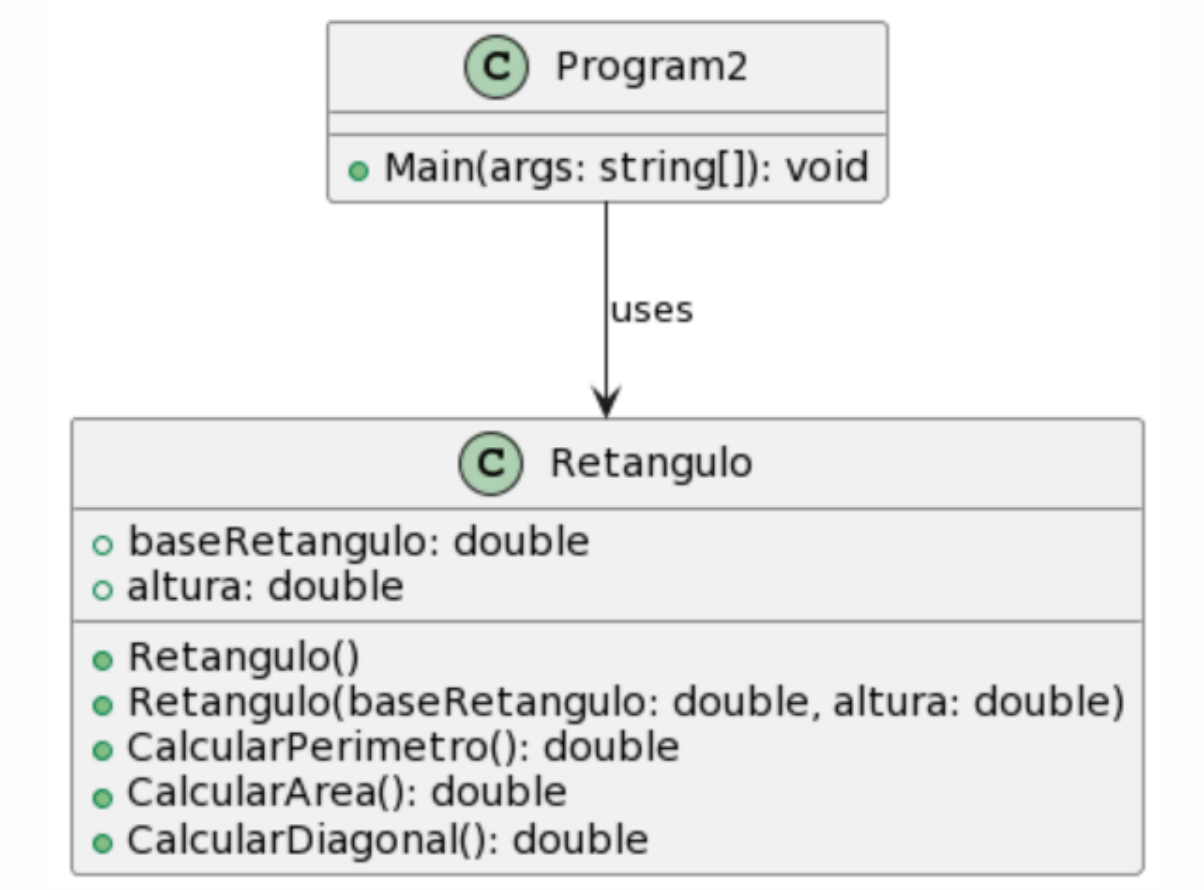
Belo Horizonte- MG
2023

QUESTÃO 1 -

```
using Questão1;  // Esta linha importa o namespace Questão1, permitindo o uso das classes desse namespace no código.

using System;
using System.Security.Cryptography;

namespace MyApp // Define o namespace da aplicação.
{
    internal class Program2
    {
        static void Main(string[] args)
        {
            Retangulo retangulo1 = new Retangulo(); // Cria uma instância da classe Retangulo chamada retangulo1.
            Retangulo retangulo2 = new Retangulo(); // Cria uma instância da classe Retangulo chamada retangulo2.
            Retangulo retangulo3 = new Retangulo(); // Cria uma instância da classe Retangulo chamada retangulo3.

            // Solicita ao usuário a entrada da altura e da base do primeiro retângulo.
            Console.WriteLine("Digite a altura do primeiro retangulo: ");
            float a1 = float.Parse(Console.ReadLine());
            Console.WriteLine("Digite a base do premeiro retanugulo: ");
            float b1 = float.Parse(Console.ReadLine());
            retangulo1.altura = a1; // Define a altura do retângulo1 com o valor inserido pelo usuário.
            retangulo1.baseRetangulo = b1; // Define a base do retângulo1 com o valor inserido pelo usuário.

            // Solicita ao usuário a entrada da altura e da base do segundo retângulo.
            Console.WriteLine("Digite a altura do segundo retangulo: ");
            float a2 = float.Parse(Console.ReadLine());
            Console.WriteLine("Digite a base do segundo retanugulo: ");
            float b2 = float.Parse(Console.ReadLine());
            retangulo2.altura = a2; // Define a altura do retângulo2 com o valor inserido pelo usuário.
            retangulo2.baseRetangulo = b2; // Define a base do retângulo2 com o valor inserido pelo usuário.

            // Solicita ao usuário a entrada da altura e da base do terceiro retângulo.
            Console.WriteLine("Digite a altura do terceiro retangulo: ");
            float a3 = float.Parse(Console.ReadLine());
            Console.WriteLine("Digite a base do terceiro retanugulo: ");
            float b3 = float.Parse(Console.ReadLine());
            retangulo3.altura = a3; // Define a altura do retângulo3 com o valor inserido pelo usuário.
            retangulo3.baseRetangulo = b3; // Define a base do retângulo3 com o valor inserido pelo usuário.

            // Imprime informações sobre o Retângulo 1
            Console.WriteLine("Retangulo 1:");
            Console.WriteLine($"Perímetro: {retangulo1.CalcularPerimetro()}"); // Calcula e imprime o perímetro do retângulo1.
            Console.WriteLine($"Área: {retangulo1.CalcularArea()}"); // Calcula e imprime a área do retângulo1.
            Console.WriteLine($"Diagonal: {retangulo1.CalcularDiagonal()}"); // Calcula e imprime a diagonal do retângulo1.
            Console.WriteLine();
```

```
            // Imprime informações sobre o Retângulo 2
            Console.WriteLine("Retangulo 2:");
            Console.WriteLine($"Perímetro: {retangulo2.CalcularPerimetro()}"); //
Calcula e imprime o perímetro do retângulo2.
            Console.WriteLine($"Área: {retangulo2.CalcularArea()}"); // Calcula e
imprime a área do retângulo2.
            Console.WriteLine($"Diagonal: {retangulo2.CalcularDiagonal()}"); //
Calcula e imprime a diagonal do retângulo2.
            Console.WriteLine();

            // Imprime informações sobre o Retângulo 3
            Console.WriteLine("Retangulo 3:");
            Console.WriteLine($"Perímetro: {retangulo3.CalcularPerimetro()}"); //
Calcula e imprime o perímetro do retângulo3.
            Console.WriteLine($"Área: {retangulo3.CalcularArea()}"); // Calcula e
imprime a área do retângulo3.
            Console.WriteLine($"Diagonal: {retangulo3.CalcularDiagonal()}"); //
Calcula e imprime a diagonal do retângulo3.
        }
    }
}


using System;
using System.Collections.Generic;
using System.Linq;
using System.Runtime.CompilerServices;
using System.Text;
using System.Threading.Tasks;

namespace Questão1
{
    internal class Retangulo
    {
        public double baseRetangulo { get; set; }
        public double altura { get; set; }

        // Construtor padrão da classe Retangulo.
        public Retangulo()
        {
            // Este construtor não realiza nenhuma ação específica ao ser
chamado.
        }

        // Construtor sobrecarregado da classe Retangulo que permite definir base
e altura ao criar uma instância.
        public Retangulo(double baseRetangulo, double altura)
        {
            this.baseRetangulo = baseRetangulo;
            this.altura = altura;
        }

        // Método para calcular o perímetro do retângulo.
        public double CalcularPerimetro()
        {
            return 2 * (baseRetangulo + altura);
        }

        // Método para calcular a área do retângulo.
        public double CalcularArea()
        {
            return baseRetangulo * altura;
        }
```

```
        // Método para calcular a diagonal do retângulo usando o teorema de
Pitágoras.
        public double CalcularDiagonal()
        {
            return Math.Sqrt(baseRetangulo * baseRetangulo + altura * altura);
        }
    }
}
```

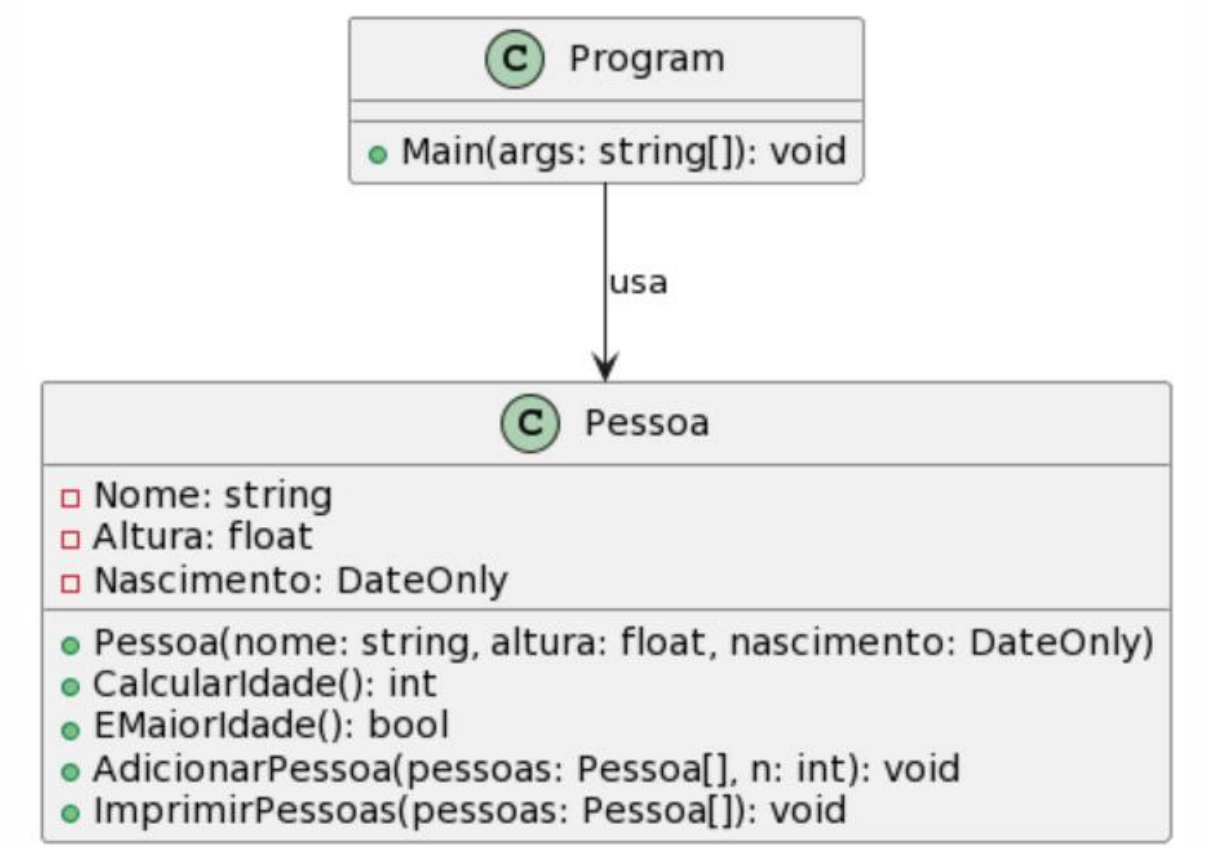


QUESTÃO 2 -

```
using System;

namespace MyApp // Observe que o namespace real pode depender do nome do projeto.
{
    internal class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Digite o numero de pessoas que você deseja
adicionar: ");
            int numeroPessoas = int.Parse(Console.ReadLine());
            Pessoa[] pessoas = new Pessoa[numeroPessoas];

            // Loop para adicionar pessoas ao array
            for (int i = 0; i < numeroPessoas; i++)
            {
                Pessoa.AdicionarPessoa(pessoas, i);
            }

            // Imprime informações sobre as pessoas no array
```

```
            Pessoa.ImprimirPessoas(pessoas);
        }
    }
}


using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

internal class Pessoa
{
    // Propriedades da classe Pessoa
    string Nome { get; set; }
    float Altura { get; set; }
    DateOnly Nascimento { get; set; }

    // Construtor da classe Pessoa
    public Pessoa(string nome, float altura, DateOnly nascimento)
    {
        Nome = nome;
        Altura = altura;
        Nascimento = nascimento;
    }

    // Método estático para adicionar uma pessoa a um array de pessoas
    public static void AdicionarPessoa(Pessoa[] pessoas, int n)
    {
        Console.WriteLine("Digite o nome da pessoa: ");
        string nome = Console.ReadLine();
        Console.WriteLine("Digite a altura da pessoa: ");
        float altura = float.Parse(Console.ReadLine());
        Console.WriteLine("Digite sua data de nascimento (DD/MM/YYYY): ");
        DateOnly nascimento = DateOnly.Parse(Console.ReadLine());
        pessoas[n] = new Pessoa(nome, altura, nascimento);
    }

    // Método estático para imprimir informações sobre as pessoas no array
    public static void ImprimirPessoas(Pessoa[] pessoas)
    {
        foreach (var pessoa in pessoas)
        {
            Console.WriteLine($"Pessoas - Nome: {pessoa.Nome}, idade:
{pessoa.CalcularIdade()}, altura: {pessoa.Altura}. É de maior?
({pessoa.EMaiorIdade()})");
        }
    }

    // Método para calcular a idade da pessoa
    public int CalcularIdade()
    {
        DateTime hoje = DateTime.Now;
        int idade = hoje.Year - Nascimento.Year;
        return idade;
    }

    // Método para verificar se a pessoa é maior de idade
    public bool EMaiorIdade()
    {
        int idade = CalcularIdade();
        if (idade >= 18)
```

```
        {
            Console.WriteLine("É maior de idade");
            return true;
        }
        else
        {
            Console.WriteLine("É menor de idade");
            return false;
        }
    }
}
```



Questão 3 -

```
using Curso; // Importando o namespace onde estão as classes

namespace MyApp
{
    internal class Program
    {
        static void Main(string[] args)
        {
            // Solicitando ao usuário o nome do curso
            Console.WriteLine("Informe o nome do curso:");
            string nomeDoCurso = Console.ReadLine();

            // Criando uma instância da classe MeuCurso (curso)
            MeuCurso curso = new MeuCurso
            {
                Identificador = 1, // Definindo o identificador do curso
                Nome = nomeDoCurso // Definindo o nome do curso com base na
entrada do usuário
```

```
            };

            for (int i = 0; i < 3; i++)
            {
                Console.WriteLine($"\nInforme os detalhes da disciplina {i +
1}:");
                Disciplina disciplina = new Disciplina(); // Criando uma
instância da classe Disciplina

                // Preenchendo os detalhes da disciplina com a entrada do usuário
                Console.Write("Código da Disciplina: ");
                disciplina.Codigo = int.Parse(Console.ReadLine());

                Console.Write("Nome da Disciplina: ");
                disciplina.Nome = Console.ReadLine();

                Console.Write("Número da Sala: ");
                disciplina.Sala = int.Parse(Console.ReadLine());

                Console.Write("Número do Prédio: ");
                disciplina.Predio = int.Parse(Console.ReadLine());

                Console.Write("Número de Alunos Matriculados: ");
                disciplina.NumeroAlunosMatriculados =
int.Parse(Console.ReadLine());

                // Loop para adicionar alunos à disciplina
                for (int j = 0; j < disciplina.NumeroAlunosMatriculados; j++)
                {
                    Console.WriteLine($"\nInforme os detalhes do aluno {j + 1} da
disciplina {i + 1}:");
                    Aluno aluno = new Aluno(); // Criando uma instância da classe
Aluno

                    // Preenchendo os detalhes do aluno com a entrada do usuário
                    Console.Write("Matrícula do Aluno: ");
                    aluno.Matricula = int.Parse(Console.ReadLine());

                    Console.Write("Nome do Aluno: ");
                    aluno.Nome = Console.ReadLine();

                    Console.Write("Data de Nascimento do Aluno (yyyy-MM-dd): ");
                    string dataNascimentoAlunoS = Console.ReadLine();
                    DateOnly.TryParseExact(dataNascimentoAlunoS, "dd/MM/yyyy",
out DateOnly dataNascimentoAluno);
                    aluno.DataNascimento = dataNascimentoAluno;

                    // Adicionando o aluno à lista de alunos da disciplina
                    disciplina.Alunos.Add(aluno);
                }

                // Adicionando a disciplina à lista de disciplinas do curso
                curso.Disciplinas.Add(disciplina);
            }

            // Informando que o curso foi criado com sucesso
            Console.WriteLine("\nCurso criado com sucesso!\n");

            // Chamando o método para imprimir os detalhes do curso e suas
disciplinas
            curso.ImprimirDetalhes();
        }
    }
```

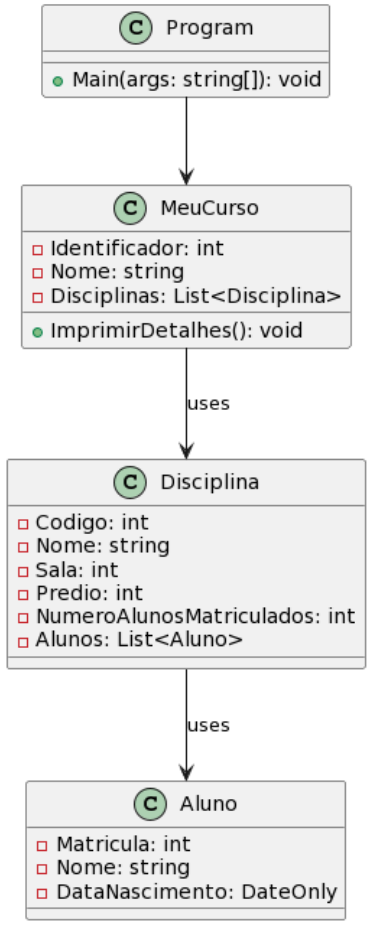
```
}
```

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Curso
{
    internal class MeuCurso
    {

        public int Identificador { get; set; }
        public string Nome { get; set; }
        public List<Disciplina> Disciplinas { get; set; } = new
List<Disciplina>();

        public MeuCurso()
        {
        }

        public void ImprimirDetalhes()
        {
            Console.WriteLine($"Curso: {Nome}");
            Console.WriteLine("Disciplinas:");

            foreach (var disciplina in Disciplinas)
            {
                Console.WriteLine($"  Código: {disciplina.Codigo}");
                Console.WriteLine($"  Nome: {disciplina.Nome}");
                Console.WriteLine($"  Sala: {disciplina.Sala}");
                Console.WriteLine($"  Prédio: {disciplina.Predio}");
                Console.WriteLine($"  Número de Alunos Matriculados:
{disciplina.NumeroAlunosMatriculados}");
                Console.WriteLine("  Alunos Matriculados:");

                foreach (var aluno in disciplina.Alunos)
                {
                    Console.WriteLine($"    Matrícula: {aluno.Matricula}");
                    Console.WriteLine($"    Nome: {aluno.Nome}");
                    Console.WriteLine($"    Data de Nascimento:
{aluno.DataNascimento.ToShortDateString()}");
                }
            }
        }
    }
}
```

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Curso
{
    internal class Disciplina
    {
        public int Codigo { get; set; }
        public string Nome { get; set; }
```

```
        public int Sala { get; set; }
        public int Predio { get; set; }
        public int NumeroAlunosMatriculados { get; set; }
        public List<Aluno> Alunos { get; set; } = new List<Aluno>();
    }

}
```

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Curso
{
    internal class Aluno
    {
        public int Matricula { get; set; }
        public string Nome { get; set; }
        public DateOnly DataNascimento { get; set; }
    }
}
```



QUESTÃO 4 –

```
using Banco; // Importa o namespace "Banco".
using System; // Importa o namespace "System", que contém classes e métodos
fundamentais.
```

```
namespace MyApp // Define o namespace "MyApp" para o programa.
{
    internal class Program // Declara uma classe chamada "Program".
    {
        static void Main(string[] args) // O método principal do programa.
        {

            Console.WriteLine("Digite o número de clientes que irá ser
adicionados: "); // Exibe uma mensagem no console.

            int nClientes = int.Parse(Console.ReadLine()); // Lê e converte a
entrada do usuário em um número inteiro.
            Clientes[] clientes = new Clientes[nClientes]; // Cria um array de
objetos da classe "Clientes".
            for (int i = 0; i < nClientes; i++) // Inicia um loop para adicionar
clientes ao array.
            {
                Clientes.AdicionarCliente(clientes, i); // Chama o método
"AdicionarCliente" da classe "Clientes".
            }

            Clientes.ImprimirClientes(clientes); // Chama o método
"ImprimirClientes" da classe "Clientes" para exibir os clientes.

        }
    }
}



using System; // Importa o namespace "System".

namespace Banco // Define o namespace "Banco" para a classe "Clientes".
{
    internal class Clientes // Declara uma classe chamada "Clientes".
    {
        public string Nome { get; set; } // Propriedade que representa o nome do
cliente.
        public int codigoCliente { get; set; } // Propriedade que representa o
código do cliente.

        public Clientes() { } // Construtor padrão da classe "Clientes".

        public Clientes(string nome, int codigoCliente) // Construtor que aceita
nome e código do cliente como parâmetros.
        {
            Nome = nome; // Inicializa a propriedade "Nome" com o valor passado
como parâmetro.
            this.codigoCliente = codigoCliente; // Inicializa a propriedade
"codigoCliente" com o valor passado como parâmetro.
        }

        public static void AdicionarCliente(Clientes[] clientes, int n) // Método
estático para adicionar um cliente ao array.
        {
            Console.WriteLine("Digite o nome do cliente: "); // Exibe uma
mensagem no console.
            string nome = Console.ReadLine(); // Lê o nome do cliente a partir da
entrada do usuário.
            Console.WriteLine("Digte o código do cliente: "); // Exibe uma
mensagem no console.
```

```
            int codigoCliente = int.Parse(Console.ReadLine()); // Lê e converte o
código do cliente a partir da entrada do usuário.

            clientes[n] = new Clientes(nome, codigoCliente); // Cria um novo
objeto "Clientes" e o adiciona ao array.
        }

        public static void ImprimirClientes(Clientes[] clientes) // Método
estático para imprimir informações sobre os clientes.
        {
            foreach (var cliente in clientes)
            { // Inicia um loop para cada cliente no array.
                Console.WriteLine($"Cliente - Nome: {cliente.Nome} - Código
cliente: {cliente.codigoCliente}"); // Exibe informações sobre o cliente no
console.
            }
        }
    }
}
```