

# **PROGRAMAÇÃO ORIENTADA A OBJETOS (POO/OOP)**

**Prof. Matheus da Fonseca Dummer**

**2022**

# **POSSIVELMENTE O PARADIGMA MAIS POPULAR ATUALMENTE**

**Classes e objetos são o núcleo da maior parte dos sistemas  
modernos**



## POO é:

- Utilização de classes abstratas para definir um "modelo" a ser seguido, onde diversos atributos em comum são alocados.
- Dentro das classes, temos uma série de especificações onde todos os objetos criados a partir dela podem herdar e acessar.
- Também podemos ter métodos/funções específicas para cada classe, criando assim sub-rotinas.
- Uma das facilidades de se usar POO é poder ter várias partes do código específicas para cada operação, podendo ser reutilizadas e também modificadas com rapidez.

## **POO é estruturada sobre quatro alicerces:**

- Encapsulamento
- Abstração
- Herança
- Poliformismo

## Encapsulamento

- É o ato de manter os atributos da classe contidos, para evitar acesso fora da classe ou dos métodos necessários e desenhados para o usuário/programa acessar.
- Evitar acessar os dados da classe de fora dela é uma das medidas de segurança para o programa.

# Abstração

- Através da abstração podemos definir toda uma lógica específica de operação e funções dentro da classe, mas expor apenas os métodos e interfaces necessárias para se conversar com a classe, simplificando as chamadas em outras partes do código.

# Herança

- Podemos criar sub-classes (as vezes chamada de classe filho/child) para receber atributos da classe principal (as vezes chamada de classe pai/parent). Estes atributos são heranças, que podem ser alterados e modificados conforme as necessidades da classe filho.



# Polimorfismo

- Semelhante a herança, aqui lidamos com métodos e atributos que podem ser passados de uma classe para a outra, caso seja necessário.

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

namespace Introducao_a_OOP
{
    public class Pessoa
    {
        public string nome;
        public string cidade;
        public string hobby;
        public int idade;

        public void Descricao()
        {
            Console.WriteLine("Olá, eu me chamo " + nome + " sou de "
                               + cidade + ". Tenho " + idade + " e gosto de " + hobby + " nas horas vagas.");
        }
    }

    internal class Program
    {
        static void Main(string[] args)
        {
            Pessoa joao = new Pessoa();

            joao.nome = "João";
            joao.cidade = "SCS";
            joao.idade = 20;
            joao.hobby = "Tocar piano";

            joao.Descricao();
        }
    }
}
```

## Construindo uma classe básica

```
public class Classe1
{
    public string atributo1;
    public int atributo2;

    public void Metodo()
    {
        Console.WriteLine("Olá, eu sou um método!");
    }
}
```

- Para chamarmos e utilizarmos esta classe, precisamos depois criar um objeto da mesma.
- Para fazer isso usamos:

```
Classe1 exemplo = new Classe1();  
  
exemplo.atributo1 = "exemplo de atributo";  
Exemplo.atributo2 = 12;  
Exemplo.Metodo();
```

# Anatomia de uma classe

```
<visibilidade> class <nome da classe>
{
    <visibilidade><tipo> <nome atributo> { get; set; }; //para atribuir as propriedades automaticamente

    <visibilidade> <tipo de retorno> <nome do método> <parametros>
    {
        // o que se deseja fazer no método
    }
}
```

## Prática

- Vamos fazer uma atividade prática que consiste em criar uma classe e dentro desta classe, vamos criar métodos para simular um pequeno banco, onde criamos um objeto Pessoa e atribuímos um nome, um salário, uma profissão a este objeto. Depois vamos criar um método para simular uma análise de crédito e retornar se é possível fornecer um cartão de crédito para a pessoa ou não.

## Referências

Microsoft. C# Reference Guide, 2022. Disponível em:  
<https://docs.microsoft.com/en-us/dotnet/csharp/language-reference/>

MANZANO, José Augusto N. G. Microsoft C# Community 2015. 1. Ed.  
São Paulo: Erica, 2016

SANTANCHÈ, André. Institute of Computing – UNICAMP, 2011.  
Disponível em:  
<https://www.ic.unicamp.br/~santanch/teaching/oop/slides/poo0201-objeto-classe-v02.pdf>