**POO – Programação Orientada a Objetos**

- Método de programação (paradigma) que usa tipos de dados personalizados.

- Em vez de operar apenas com tipos de dados primitivos, podemos construir novos tipos de dados.

- Baseia-se fundamentalmente no conceito de **objetos**.

**Vantagens da OO**

- Fornece uma estrutura modular para a construção de programas

- O software se torna mais fácil de manter

- Reuso de Código -> desenvolvimento mais rápido

- Objetos podem ser reutilizados em aplicações diferentes

- Encapsulamento: não é necessário conhecer e implementação interna de um objeto para poder usa-lo

**Abstração**

- **Abstrair** é selecionar aspectos específicos de um problema a ser analisado, deixando de lado outros aspectos. Representar uma entidade do mundo real na forma de ideias.

- Entidades abstraídas podem se comunicar entre si, por meio de troca de mensagens.

*“Pelo princípio da abstração, isolamos os objetos que queremos representar do ambiente complexo em que se situam, e nesses objetos representamos somente as características que são relevantes pro problema em questão.”*

**Classes**

- Uma classe representa uma ideia ou conceito e classifica objetos que tenham propriedades similares.

- Blocos de construção mais importantes em sistemas OO, com responsabilidades bem definidas dentro da aplicação.

- Coleção de objetos descritos com os mesmos atributos e operações.

- Tipo personalizado de dados, ‘Molde’ para criação de objetos .

**Objeto**

- Ocorrência especifica de uma classe – “Instancia de classe”

- Representa entidades do mundo real, como carros, pessoas, contas correntes, etc,. e outros conceitos, como gráficos (círculos, quadrados, cones, esferas, etc.)

- Tem características próprias (atributos) e executa ações (métodos), provenientes da classe que originou o objeto.

Herança. Encapsulamento e Polimorfismo.

**Herança**

- Homens, baleias e gatos são mamíferos que compartilham características, mas com atributos distintos.

- como modelar baleias, gatos e humanos em software?

- Podemos criar classes distintas para cada animal, mas essas classes teriam muitos comportamentos em comum, como respirar, mamar e reproduzir-se, o que ocasionaria repetição desnecessária de código.

**Encapsulamento**

- Combinação de atributos e métodos em uma classe, deixando visível apenas o necessário para comunicação entre objetos, como alguns detalhes da implementação ou logica de um método.

- Permite ocultar a complexidade do código.

- Não é necessário entender o funcionamento interno da classe para poder utilizar seus métodos.

**Polimorfismo**

- Poli = Muitos; Morphos = formas. Portanto, polimorfismo = muitas formas.

- Uma operação de um objeto pode assumir mais de um comportamento dependendo da chamada recebida, tratando e devolvendo respostas distintas.

- Ocorre quando um objeto tem um comportamento diferente para uma mesma ação.

**POO**

- Metodo de programação (paradigma) que usa tipos de dados personalizados.

- Em vez de operar apenas com tipos de dados primitivos, podemos construir novos tipos de dados.

- Baseia-se fundamentalmente no conceito de **objetos**.

**Criando Classes**

**Método construtor**

Método especial usado para inicializar um objeto instanciado.

**Sintaxe:**

def \_\_init\_\_(self, [parametros]):

Código do método construtor

Quando a classe inclui o método \_\_init\_\_(), ele é **automaticamente invocado** quando ela é instanciada.

**Métodos são: funções internas de uma classe.**

**O identificador *self***

- Sintaticamente, o parâmetro self, identifica a instancia sobre a qual um método é invocado. Ou seja, é uma referencia à instância atual da classe (objeto).

- é usado para acessar os membros que pertencem à classe em si.

- Deve ser sempre primeiro parâmetro usado na definição de um método.