

Programação Orientada a Objetos - POO

É um paradigma* de programação!

O que é Programação Orientada a Objetos?

É um paradigma* de programação!

Paradigma de programação: é uma forma de desenvolver softwares. É ele que define como o programador deve pensar e organizar o código.

Principais Paradigmas:

Programação Estruturada: C, Pascal

Programação Funcional: Haskell, Lisp, Scala

Programação Orientada a Objetos:
Java

Programação Orientada a Objetos: POO

OOP: Object Oriented Programming

Objetivo: Aproximar o mundo digital do mundo real

História:

Baixo nível >

Programação linear >

Programação Estruturada (linear, mas organizada - permite a criação de sistemas mais complexos) >

Programação Modular >

Alto nível (POO)

Programação Orientada a Objetos: POO

OOP: Object Oriented Programing

Postulado*: o que se considera como fato reconhecido e ponto de partida, implícito ou explícito, de uma argumentação; premissa

Criação: Alan Kay (matemático e biólogo)

Motivação: Desenvolveu o POO pensando na educação e para crianças - 1970

Postulado* de Alan Kay:

"O computador ideal deve funcionar como um organismo vivo, isso é, cada célula se relaciona com outras a fim de alcançar um objetivo, mas cada uma funciona de forma autônoma. As células poderiam também reagrupar-se para resolver um outro problema ou desempenhar outras funções"

Programação Orientada a Objetos: POO

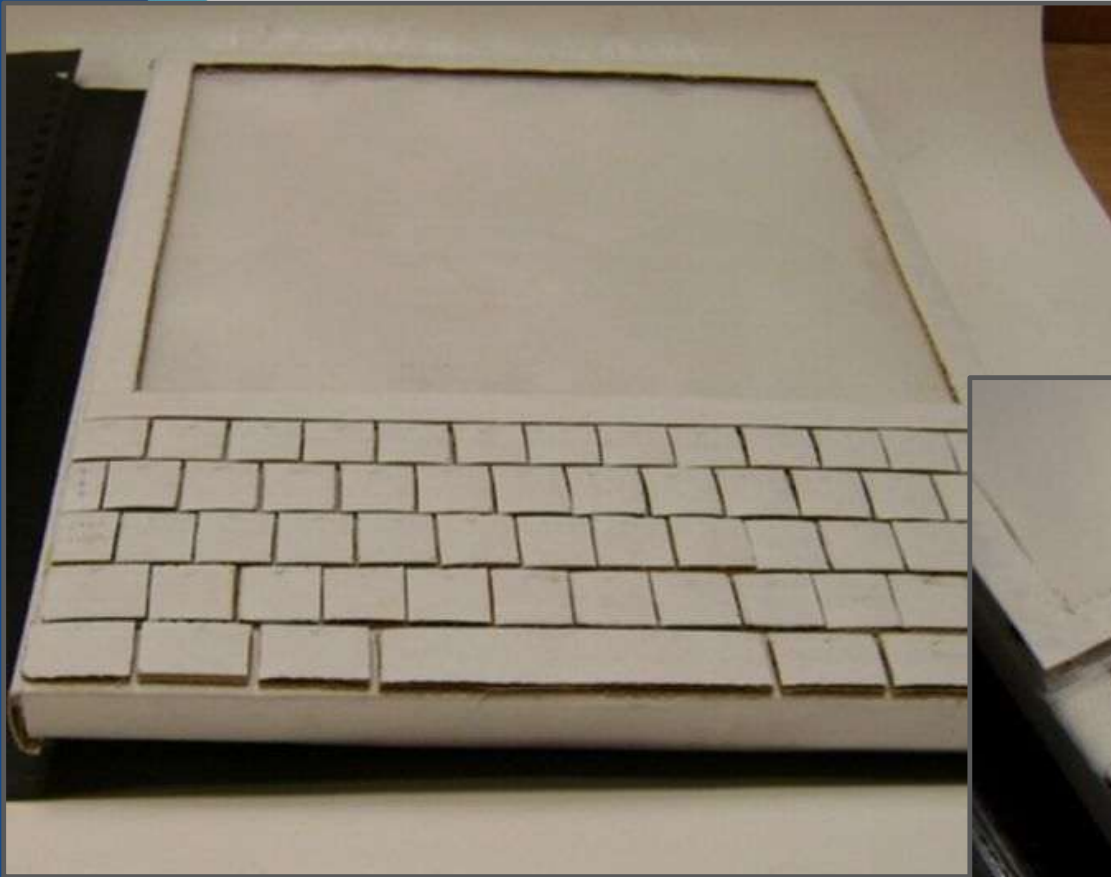
OOP: Object Oriented Programming

Curiosidades: Prototipou o Dynabook, "avô" do notebook - um aparelho com uma tela e um teclado, que serviria para ensinar crianças

Na época, não havia tecnologia para desenvolver tal produto, mas ele já sabia que as linguagens de programação existentes não seriam suficientes para que a sua ideia saísse do papel

Supostamente, qualquer objeto pode ser descrito/explicado utilizando POO





Programação Orientada a Objetos: POO

OOP: Object Oriented Programing

Linguagens POO:

C++

Java - POO Obrigatória

PHP

Python

Javascript / Typescript

Ruby

Visual Basic

Programação Orientada a Objetos: POO

OOP: Object Oriented Programming

Vantagens: COMERNada



Confiável: Isolamento entre as partes

Oportuno: Partes diferentes podem ser desenvolvidas em paralelo

Manutenível: Atualização é mais fácil, pois permite alterar somente uma parte sem afetar as demais

Extensível: Pode crescer em módulos/funcionalidades, somente "encaixando" isso ao restante do código existente

Reutilizável: Possibilidade de reutilizar módulos, no mesmo ou em outros sistemas

Natural: Fácil de entender, pois tenta simular a própria natureza (objetos reais)

Programação Orientada a Objetos: POO

Classes

Objetos

Atributos
Métodos

Abstração

Classes

É um **Molde**

Para ser uma classe:

Atributos: O que tem

Métodos: O que faz

Estado: Como está

Classes

O que tem:

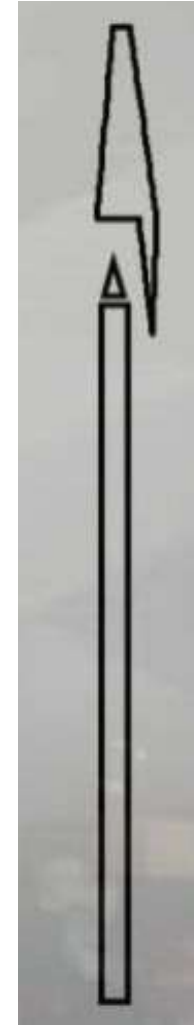
- Modelo
- Cor
- Carga
- Tampa

O que faz :

- Escrevo
- Rabisco
- Tampar
- Destampar

Como está:

- Com 90% de carga
- Ponta fina
- Destampada
- Escrevendo



Objetos - Atributos e Métodos

Objeto: Coisa material ou abstrata que pode ser percebida pelos sentidos e descrita por meio das suas **características**, **comportamentos** e **estado** atual.

Controle:

- +material
- +formato
- +cor
- função01
- função02
- função03
- *led ligado/desligado
- *buscando rede/ conectado

Propaganda:

- +nome
- +mensagem
- +imagem
- exibir
- ocultar
- *maximizada / oculta
- *piscando / ofuscada

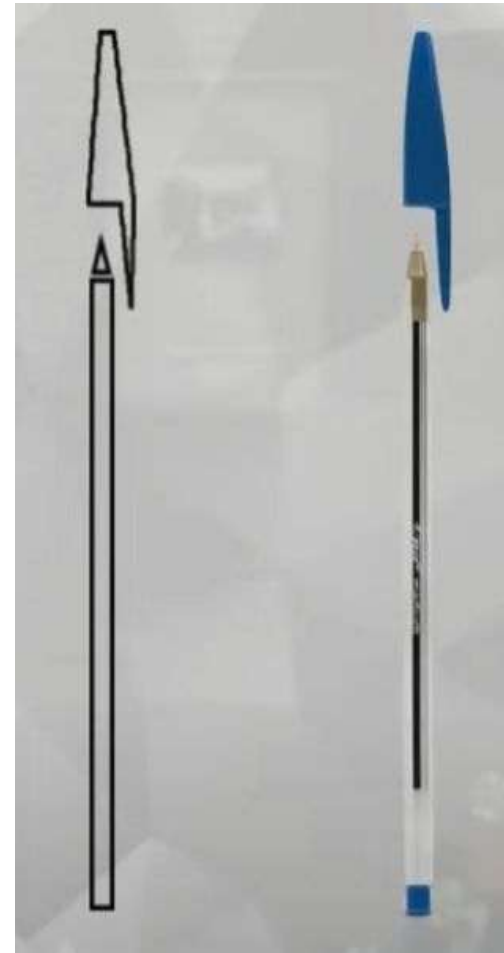


Classe vs. Objeto

O Objeto é a materialização de uma Classe.

A Classe é o molde e o Objeto é o produto que partiu desse molde.

Chamamos o processo de Criar um Objeto através de uma classe, de Instanciar um Objeto



Classe > Objeto

Atributos:

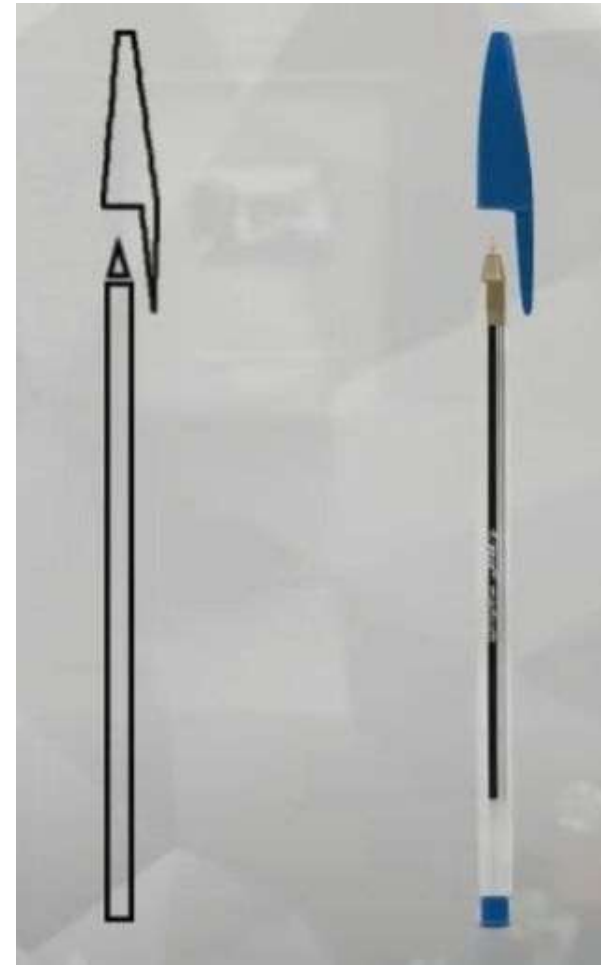
- Modelo
- Cor
- Carga
- Tampa

Métodos():

- Escrevo
- Rabisco
- Tampar
- Destampar

Estado:

- Com 90% de carga
- Ponta 0.5
- Destampada
- Escrevendo



Abstração

Fazer com que a Classe, ou seja, o molde de Objeto, seja o mais genérico possível e, ao mesmo tempo, útil e focado ao caso específico.

Exemplo:

Ao criar um molde/classe para um objeto caneta, mesmo sabendo que toda caneta possui uma marca, somente vou precisar declarar isso como um atributo caso essa marca for utilizada de fato na minha caneta.

Pode ser que seja necessário apenas na classe/objeto da embalagem.

Atividade avaliativa n1.1

- criar uma Classe
- criar Atributos, Métodos e Estado
- Comentar o que é o que no código (Atributos, Métodos e Estado)
- Instanciar alguns Objetos a partir da Classe
- Atribuir diferentes valores aos Objetos
- Realizar algumas mudanças e verificações de Estado

```
// Pseudo código exemplo
Classe Caneta
    modelo: String
    cor: String
    ponta: double
    Carga: double
    tampada: boolean

    Metodo escrever()
        Se (tampada) entao
            Escreva(erro)
        senao
            Escreva("qq coisa")
    FimDoMetodo

    Metodo tampar()
        tampada = verdadeiro
    FimDoMetodo

    Metodo destampar()
        tampada = falso
    FimDoMetodo

FimClasse

// -----
//nome do objeto | criar | Classe
caneta01 = nova Caneta
caneta01.cor = "vermelha"
caneta01.ponta = 0.5
caneta01.tampada = false
caneta01.escrever()

outraCaneta = nova Caneta
outraCaneta.cor = "azul"
outraCaneta.ponta = 1.0
outraCaneta.tampada = false
outraCaneta.tampar()
```