

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

CURSO REFORÇO POO01

LUCAS FERNANDO
KELLY CRISTINA



INTRODUÇÃO

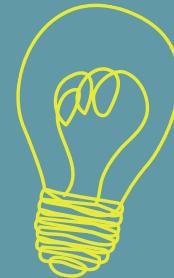
01

COMPILEADOR



02

INTERPRETADOR



PARADIGMAS



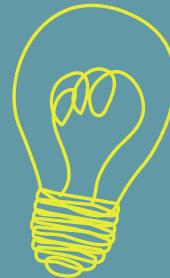
01

COMPILADOR

Em linguagem C, um compilador analisa todo o código a fim de traduzi-lo de uma vez;

02

INTERPRETADOR



PARADIGMAS



01

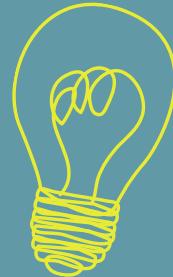
COMPILADOR

Em linguagem C, um compilador analisa todo o código a fim de traduzi-lo de uma vez;

02

INTERPRETADOR

Em Java o interpretador faz o trabalho de conversão aos poucos, sempre que uma declaração ou função é executada



PARADIGMAS



TIPOS DE DADOS



Classificação	Tipo	Descrição
Lógico	boolean	Pode possuir os valores true (verdadeiro) ou false (falso)
	byte	Abrange de -128 a 127 (8 bits)
	short	Abrange de -32768 a 32767 (16 bits)
	int	Abrange de -2147483648 a 2147483647 (32 bits)
Inteiro	long	Abrange de -2^{63} a $(2^{63})-1$ (64 bits)
Ponto Flutuante	float	Abrange de 1.40239846^{-45} a 3.40282347^{+38} com precisão simples (32 bits)
	double	Abrange de $4.94065645841246544^{-324}$ a $1.7976931348623157^{+308}$ com precisão dupla (64 bits)
Caracter	char	Pode armazenar um caracteres unicode (16 bits) ou um inteiro entre 0 e 65535



PILARES DA PROGRAMAÇÃO ORIENTADA A OBJETOS





PILARES



01 → ?

02 → ?

03 → ?

04 → ?





PILARES



01 ABSTRAÇÃO



PILARES



01 ABSTRAÇÃO

02 ENCAPSULAMENTO

PILARES



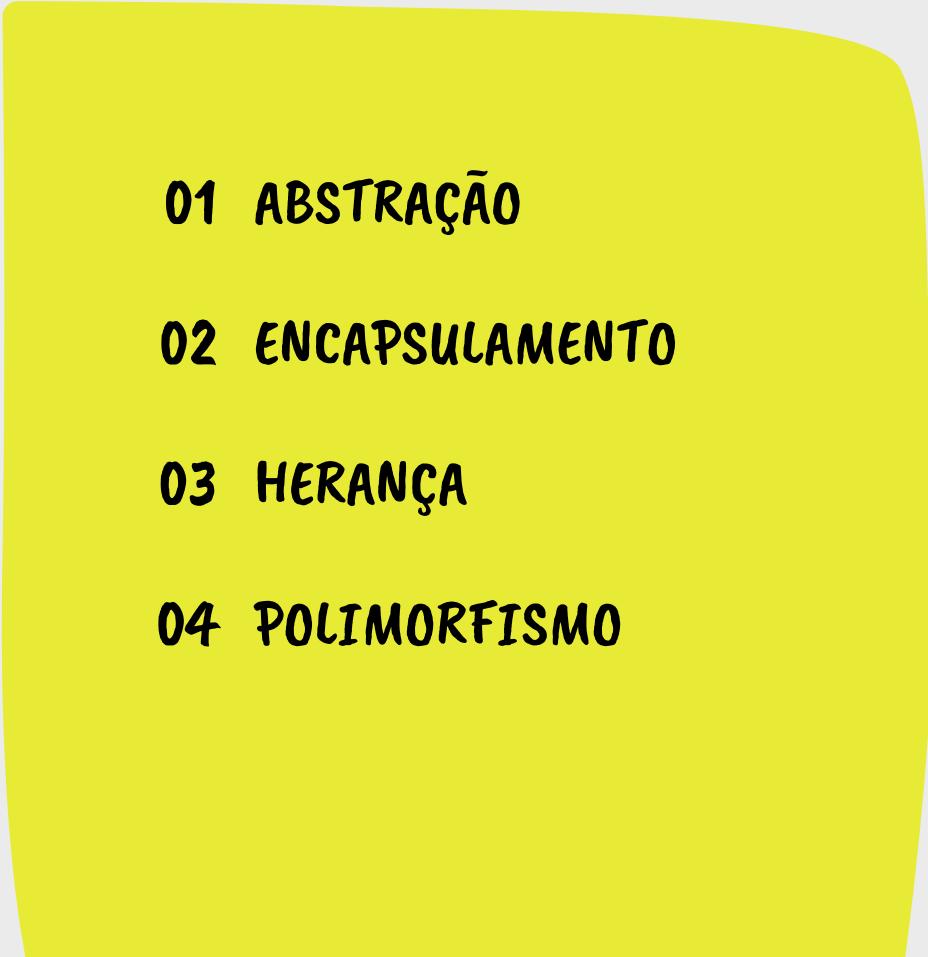
01 ABSTRAÇÃO

02 ENCAPSULAMENTO

03 HERANÇA



PILARES



01 ABSTRAÇÃO

02 ENCAPSULAMENTO

03 HERANÇA

04 POLIMORFISMO





ABSTRAÇÃO

CAPACIDADE QUE TEMOS DE ABSTRAIR O MUNDO REAL EM OBJETOS. OU SEJA CRIAR REPRESENTAÇÕES (POR MEIO DE OBJETOS DO MUNDO REAL);

NA POO , AS ABSTRAÇÕES SÃO FEITAS POR MEIO DA CRIAÇÃO DE CLASSES;



ENCAPSULAMENTO

CONSISTE NA SEPARAÇÃO DOS ASPECTOS INTERNOS E EXTERNOS DE UM OBJETO.

É UTILIZADO PARA IMPEDIR O ACESSO DIRETO AO ESTADO DE UM OBJETO. (SEGURANÇA).



HERANÇA

É O MECANISMO PELO QUAL UMA CLASSE PODE ESTENDER OUTRA, APROVEITANDO SEUS COMPORTAMENTOS (MÉTODOS) E ESTADOS POSSÍVEIS (ATRIBUTOS);

RÉUSO E REAPROVEITAMENTO DE CÓDIGO;





POLIMORFISMO

SIGNIFICA VÁRIAS FORMAS;

PERMITE QUE UMA MESMA OPERAÇÃO POSSA SER DEFINIDA PARA DIFERENTES TIPOS DE CLASSE, E CADA UMA DELAS A IMPLEMENTA COMO QUISER;



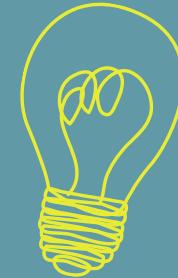
OBJETO



COISA MATERIAL OU
ABSTRATA QUE PODE SER
PERCEBIDA PELOS
SENTIDOS E DESCrita POR
MEIO DE SUAS
CARACTERÍSTICAS,
COMPORTAMENTOS E
ESTADO ATUAL



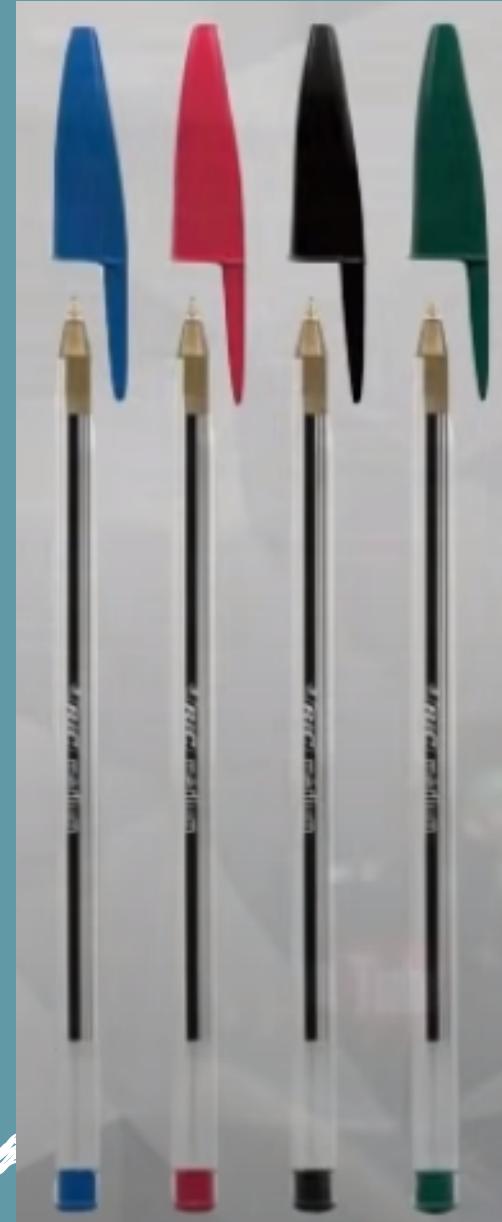
OBJETO



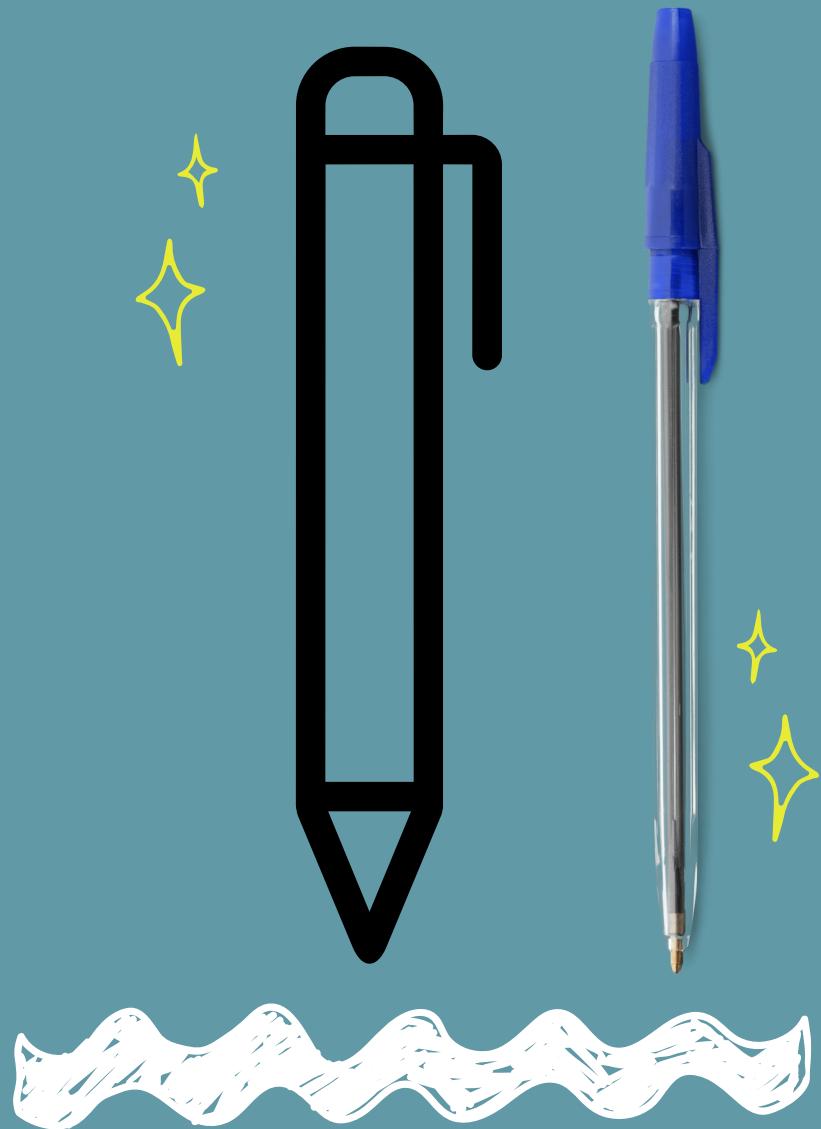
Objeto caneta, sendo
representado no mundo real
com características próprias.



- **POSSO TER VÁRIOS OBJETOS CANETAS;**
- **SE ANALISAR TODAS SEGUE A MESMA FORMA OU CLASSIFICAÇÃO;**
- **QUANDO SE CRIA CANETAS TEM UM MODELO (MODO A SER SEGUIDO);**



**ASSIM TENHO O
MODELO E A CANETA**



CLASSE



Uma classe é uma forma de definir um tipo de dado. Ela é formada por dados e comportamentos. Para definir os dados são utilizados os atributos, e para definir o comportamento são utilizados métodos;

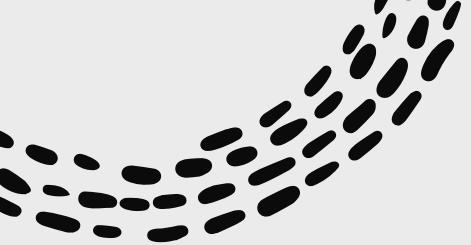




CLASSE



OBJETO

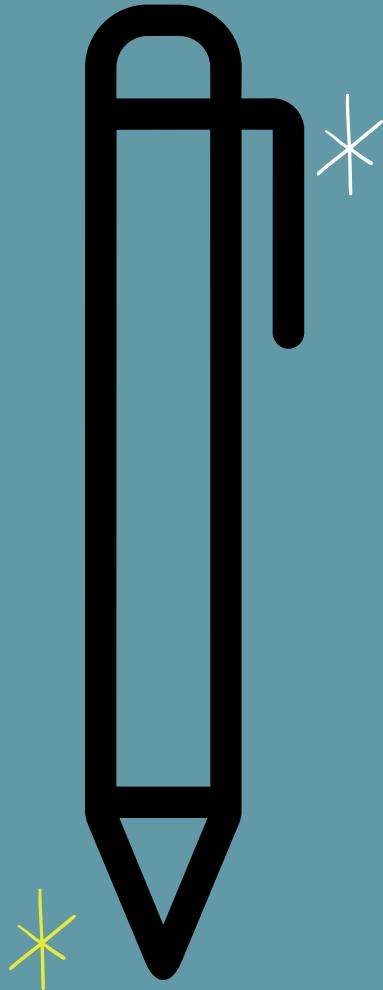


UMA CLASSE RESPONDE A TRES PERGUNTAS:



- ★ **QUE COISAS EU TENHO?**
- ★ **QUE COISAS EU FAÇO?**
- ★ **COMO EU ESTOU AGORA?**

QUE COISAS EU TENHO ?



- **MODELO**
- **COR**
- **PONTA**
- **CARGA**
- **TAMPA**



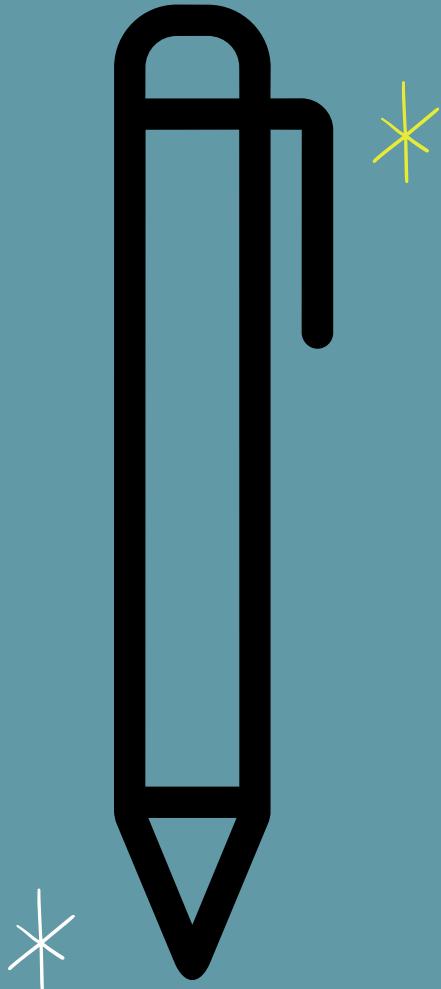
QUE COISAS EU FAÇO?



- **ESCREVER**
- **RABISCAR**
- **PINTAR**
- **TAMPAR**
- **DESTAMPAR**



COMO ESTOU AGORA ? (ESTADO)



- 50% DE TINTA;
- TEM PONTA FINA;
- COR AZUL;
- DESTAMPADA;
- ESCREVENDO;



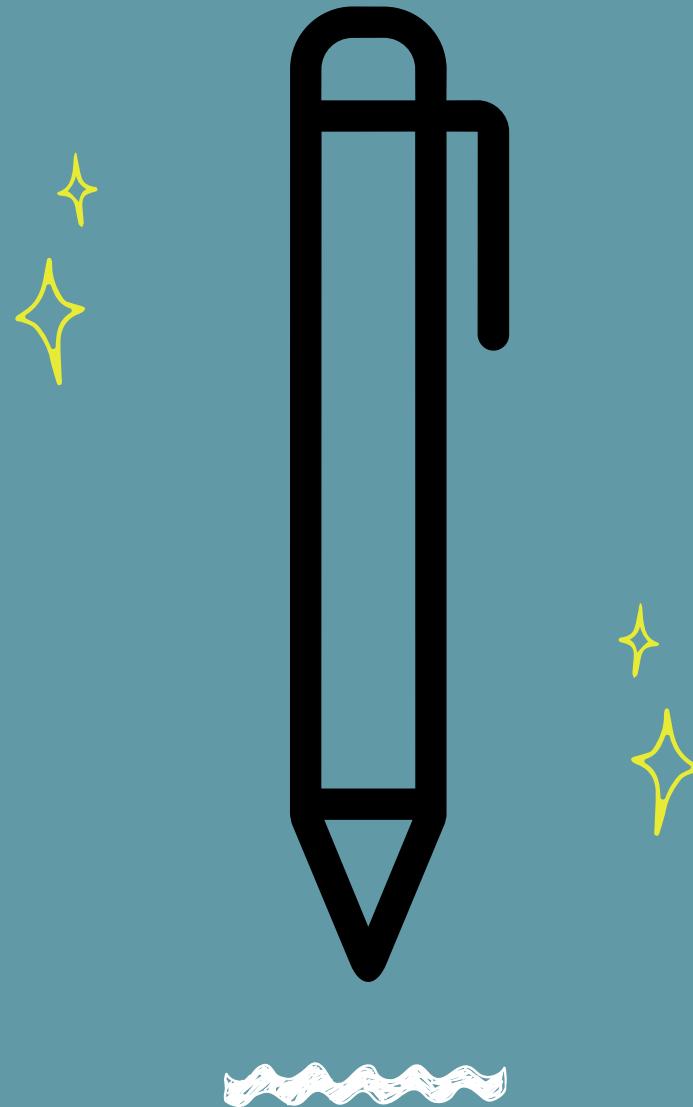
ATRIBUTOS

MODELO
COR
PONTA
CARGA
TAMPA

MÉTODOS

ESCREVER
RABISCAR
PINTAR
TAMPAR
DESTAMPAR

ESTADO



**AS CARACTERÍSTICAS QUÊ
DESCREVEM UM OBJETO É
DENOMINADO **ATRIBUTO;****

**AS AÇÕES QUE UM OBJETO PODE
EXECUTAR SÃO DENOMINDADOS
MÉTODOS;**

EXEMPLO



EXEMPLO

A CANETA POSSUI COMO CARACTERÍSTICAS **MODELO**, **COR**, **CARGA DE TINTA**, **TAMPA** ONDE QUANDO UTILIZAMOS **PODEMOS RABISCAR E TAMPAR**;

CANETA

+modelo: String
+cor: String
+carga: Int
+tampa: Boolean

+ rabiscar(): void
+tampar(): void

EXEMPLO

Palavra reservada

```
public class Caneta {  
    String modelo;  
    String cor;  
    Integer carga;  
    Boolean tampa;
```

Nome da classe

```
    public void rabiscar() {  
        System.out.println("Rabiscando na folha de caderno");  
    }  
  
    public void tampar() {  
        System.out.println("Tampando caneta");  
    }  
}
```

EXEMPLO

```
public class Caneta {  
    String modelo;  
    String cor;  
    Integer carga;  
    Boolean tampa;
```

Atributos

```
public void rabiscar() {  
    System.out.println("Rabiscando na folha de caderno");  
}  
public void tampar() {  
    System.out.println("Tampando caneta");  
}
```

Métodos

EXEMPLO

```
public class Carro {  
    String modelo;  
    String cor;  
    Integer carga;  
    Boolean tampa;
```

Tipo de dado

Nome

- não pode começar com número;
- não pode ser palavra reservada;

EXEMPLO

Modificador de acesso



Tipo de retorno



Nome do método



```
public void rabiscar() {  
    System.out.println("Rabiscando na folha de caderno");  
}  
  
public void tampar() {  
    System.out.println("Tampando caneta");  
}
```

PRÁTICA



PRÁTICA

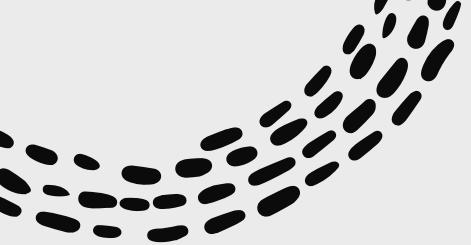
1- Cria uma classe FabricaCaneta e implemente.

- **Atributos**

- **marca (String)**
- **modelo (String)**
- **cor (String)**
- **tamanho (Integer)**

- **Métodos**

- **verificarTampa(se tiver tampada mostrar mensagem: A caneta esta tampada; Senão mostrar mensagem: Caneta destampada;**
- **verificarTinta(se tinta<30, mostrar mensagem: Caneta esta com pouca tinta, senão: Caneta esta com muita tinta ainda, não é necessário comprar outra;**



REFERÊNCIAS



GUANABARA (CURSO EM VÍDEO)



DEV MEDIA



FIM

DÚVIDAS OU PERGUNTAS?

