**Paradigmas de orientação a objetos**

**Abstração**

A classe é um modelo do objeto, os atributos são características da classe e devem ser definidos dentro do método constructor(), a palavra reservada **this** faz referência com a classe. Para definir um atributo é utilizada a seguinte sintaxe: **this.nomeAtributo**

class ContaBancaria {

  constructor() {

*this*.agencia = 1075

*this*.numeroConta = 8351125

*this*.saldo = 50

  }

  depositar(*valor*) {

*this*.saldo = *this*.saldo + valor

  }

  sacar(*valor*){

*this*.saldo = *this*.saldo - valor

  }

  consultarSaldo(){

    document.write("O saldo atual é: " + *this*.saldo)

  }

}

let objeto = **new** *ContaBancaria*()

objeto.consultarSaldo()

**Herança**

É usada para evitar a redundância de código, a herança entre classes é realizada através da palavra reservada **extends**. Os termos mais utilizados para definição de herança são: superclasse(classe herdada) e subclasse(classe que herda), classe pai e classe filha, classe genérica e classe especializada.

class Animal {

  constructor() {

*this*.cor = ''

*this*.tamanho = ''

  }

  dormir() {

    document.write('Dormir')

  }

}

class Cachorro extends *Animal*{

  constructor() {

*this*.orelhas = 'Grande e caídas'

  }

  correr() {

    document.write('Correr')

  }

  rosnar() {

    document.write('Rosnar')

  }

}

class Passaro extends *Animal*{

  constructor() {

*this*.bico = 'Curto'

  }

  voar() {

    document.write('Voar')

  }

}

**Polimorfismo**

Se trata da sobrescrita de métodos, acontece quando se trabalha com o paradigma de herança, exemplo: quando uma subclasse herda os atributos e métodos de uma superclasse, não necessariamente o mesmo comportamento precisa ser imposto aos métodos herdados, então nessa situação os métodos herdados pela classe filha são corrigidos pela sobreposição.

class Passaro{

  voar(){

    document.write('Voou!')

  }

}

class Papagaio extends *Passaro*{

}

class Avestruz extends *Passaro*{

   voar(){

    document.write('Não sabe voar!')

  }

}

let papagaio = **new** Papagaio()

let avestruz = **new** *Avestruz*()

papagaio.voar()

document.write("<br />")

avestruz.voar()