SENAI CIMATEC

SISTEMA FIEB

TECNOLOGIA E INOVAÇÃO EM PROL DA INDÚSTRIA

SENAI CIMATEC

SISTEMA FIEB

Desenvolvimento de sistemas I - 72 h

SENAI CIMATEC SISTEMA FIEB
Federação das Indústrias do Estado da Bahia

Classes

Prof: Diego Corrêa

Entidade

- Entidade = Objeto
- Alguma coisa que tenha sua própria existência, características e que apresente alguma função dentro do mundo real
- São abstrações dos objetos existentes no mundo real

Entidade

- Possuem os mesmos comportamentos
- Possuem os mesmos estados
- Podem representar os objetos no mundo real em vários níveis de abstração
- Depende de decisões do observador do mundo.

Objetos

- Objetos possuem estados (atributos) e comportamento (métodos)

- Uma porta tem:
  - Estados: aberta, fechada
  - Comportamento: abrir, fechar

Atributos

- Características que compõem um objeto
  - Podem ser um simples valor ou outro Objeto

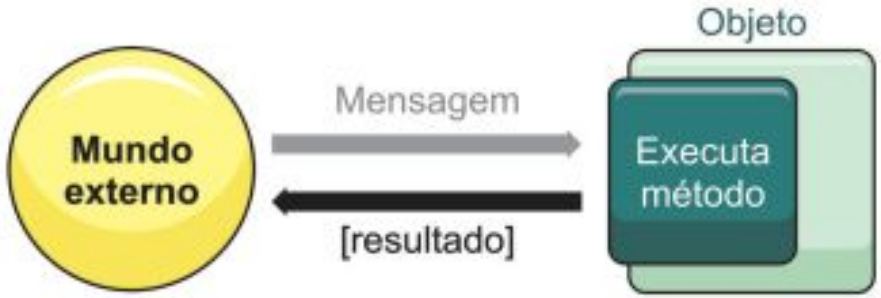
- Objeto:Funcionário
  - Atributos:nome, cpf

- Objeto:Departamento
  - Atributos: funcionário, localização

Métodos

- É uma operação que realiza ações e *__modifica__ os valores dos atributos do objeto responsável pela sua execução

- *Nem sempre um método modifica o valor dos atributos ou altera o comportamento, pois pode ser apenas um método de consulta



Métodos x Atributos

- Os atributos vêm do conceito de abstração:
  - Propriedades essenciais para representar um objeto real;

- Os métodos vêm da descrição das funções do objeto:
  - Objeto: Funcionário
    - Atributos: nome, cpf
    - Método: baterPonto()

Classes

- Exemplo prático
- Considere um programa para um banco
- O que toda conta tem e é importante para nós?

Classes

- Exemplo prático
- Considere um programa para um banco
- O que toda conta tem e é importante para nós?
  - número da conta
  - nome do dono da conta
  - saldo
  - limite

Classes

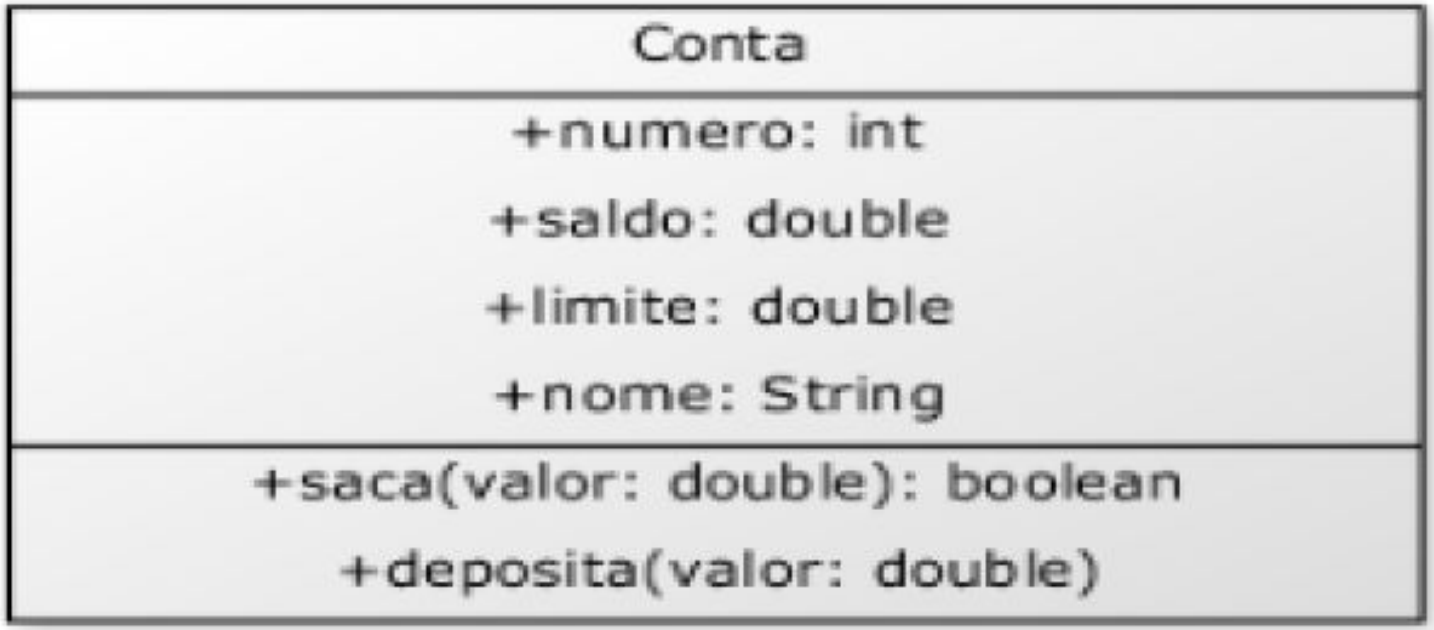
- O que toda conta faz e é importante para nós?

Classes

- O que toda conta faz e é importante para nós?
  - Sacar uma quantidade x
  - Depositar uma quantidade x
  - Imprimir o nome do dono da conta
  - Exibir o saldo atual
  - Transferir uma quantidade x para uma outra conta y
  - Exibir o tipo de conta

Classe Conta

Implementar a classe Conta abaixo



Atributos da Conta

```
class Conta {
    int numero;
    String dono;
    double saldo;
    double limite;
}
```

Instância de Objetos

Construir um objeto com todas as características de uma mesma classe significa que está fazendo uma Operação de Instanciação

Criando uma instância

```
class Programa {
    public static void main(String[] args) {
    Conta minhaConta;
    minhaConta = new Conta();
    }
}
```

- Através da variável minhaConta, podemos acessar o objeto recém criado para alterar seu dono, seu saldo, etc

SENAI CIMATEC SISTEMA FIEB
Federação das Indústrias do Estado da Bahia

Atribuindo valores

```
class Programa {
    public static void main(String[] args) {
        Conta minhaConta; minhaConta = new Conta();
        minhaConta.dono = “Fran";
        minhaConta.saldo = 1000.0;
        minhaConta.limite = 2000.0;
        minhaConta.numero = 123;

        }
}
```

Métodos

- Como vimos, operações realizadas em objetos
  - No nosso exemplo: Queremos criar um método que **saca** uma determinada **quantidade** e não devolve **nenhuma informação** para quem acionar esse método

Ex: Conta

```
class Conta {
    void saca(double quantidade) {
    double novoSaldo = this.saldo - quantidade;
    this.saldo = novoSaldo;
    }
}
```

- A palavra chave **void** diz que, quando você pedir para a conta sacar uma quantia, nenhuma informação será

Ex: Conta

```
class Conta {
    double salario;
    void saca(double quantidade) {
    double novoSaldo = this.saldo - quantidade;
    this.saldo = novoSaldo;
    }
}
```

Argumento: Usado para quando o método for chamado – uma quantidade será informada

Especifica que é um atributo da classe

A palavra chave **void** diz que, quando você pedir para a conta sacar uma quantia, nenhuma informação será enviada de volta a quem pediu

Ex: Conta

```
class Conta {
    void deposita(double quantidade) {
        this.saldo += quantidade;
    }
}
```

Invocação do método

Invocação do método

```
class Programa{
 public static void main(String[] args) {
    Conta minhaConta;
    minhaConta = new Conta();
    minhaConta.dono = “Fran";
    minhaConta.saldo = 1000;
    minhaConta.saca(200);
    minhaConta.deposita(500);
    System.out.println(minhaConta.saldo);
    }
}
```

Método com retorno

- Um método pode retornar um valor para o código que o chamou
- Nos exemplos anteriores onde estávamos usando o void
- A palavra chave return indica que o método vai terminar ali, retornando tal informação

SENAI CIMATEC SISTEMA FIEB
Federação das Indústrias do Estado da Bahia

Ex: Conta

```
class Conta {
    boolean sacaValor(double valor) {
        if (this.saldo + this.limite < valor ) {
                return false;
        } else {
                double novoSaldo = this.saldo - quantidade;
                this.saldo = novoSaldo;
                return true;
                }
         }
}
```

SENAI CIMATEC SISTEMA FIEB
Federação das Indústrias do Estado da Bahia

Ex: Conta

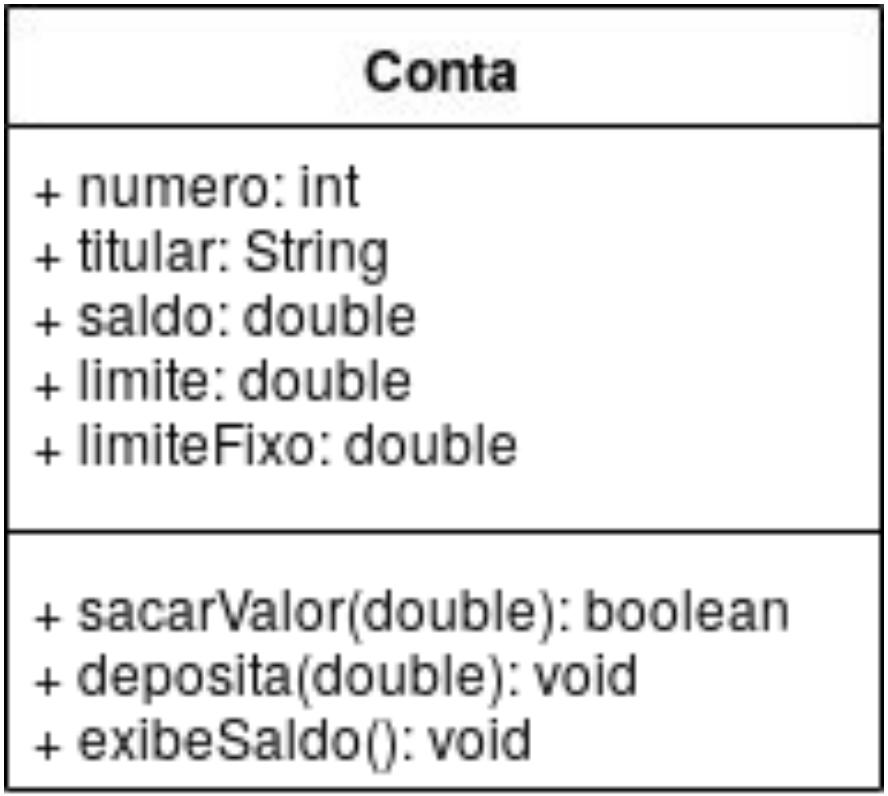
```
minhaConta.saldo = 1000;
boolean consegui = minhaConta.saca(2000);
 if (consegui == true) {
    System.out.println("Consegui sacar");
    minhaConta.exibeSaldo();
 } else {
    System.out.println("Não consegui sacar");
    minhaConta.exibeSaldo();
}
```

Exercício

Implementar o diagrama a seguir, alterando os métodos de saque e depósito. O Saque deve considerar retirar dinheiro só do saldo, ou do saldo e do limite. O depósito deve considerar sempre o limite, colocando o dinheiro de volta assim que depositado e o restante direto no saldo.

SENAI CIMATEC SISTEMA FIEB
Federação das Indústrias do Estado da Bahia

Exercício



Instanciando 2 Objetos

```
class TestaDuasContas {
    public static void main(String[] args) {
        Conta minhaConta;
        minhaConta = new Conta();
        minhaConta.saldo = 1000;
        Conta outraConta;
        meuSonho = new Conta();
        outraConta.saldo = 15000;
    }
}
```

Objeto como parâmetro

E se quisermos ter um método que transfere dinheiro entre duas contas?

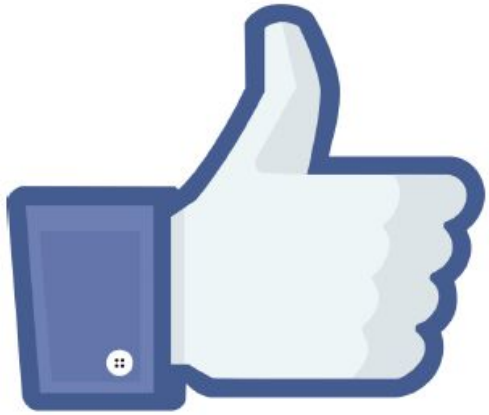
– Como fazer?

Objeto como parâmetro

- O método deve receber dois parâmetros
- Conta de destino (Tipo Conta)
- Valor (que vai ser transferido)

```
class Conta {
    void transfere(Conta destino, double valor) {
        this.saldo = this.saldo - valor;
        destino.saldo = destino.saldo + valor;
    }
}
```

Que tal se a gente incrementar o código e verificar também se a conta possui a quantidade a ser transferida disponível?



Agora crie dois objetos: conta1 e conta2, atribua saldos para ambos e transfira 50 reais da conta1 para a conta2, verificando se há dinheiro suficiente na conta1 e informando se a operação foi realizada ou não.

Atividade

- Criar uma classe com o nome Carro com os atributos:
  - String cor;
  - String modelo;
  - double velocidadeAtual;
  - double velocidadeMaxima;
- E os métodos:
  - void liga(): que exibirá uma mensagem “Carro ligado”;
  - void acelera(): passando por parâmetro a quantidade que será somada à velocidadeAtual.
  - int pegaMarcha(): condições para retornar em qual marcha o carro está:
    - Entre velocidade 0 e 20: retorna marcha 1
    - Entre velocidade 20 e 40: retorna marcha 2
    - Entre velocidade 40 e 60: retorna marcha 3
    - Entre velocidade 70 e 100: retorna marcha 4
    - Acima de 100: retornar marcha 5