

Departamento de Engenharias e Computação  
Colegiado de Ciência da Computação  
Disciplina: Linguagem de Programação III



Cap 2 – Conceitos Básicos de Orientação a Objetos

## **Parte 1 – Classes, Objetos, Atributos e Métodos**

Professor: Otacílio José Pereira

# Plano de Aula

- ***Objetivos***

- Iniciar o aprendizado de Orientação a Objetos
- Compreender os conceitos básicos de Classes, Objetos, Atributos e Métodos

- ***Tópicos***

- Contexto da disciplina
- Foco agora e próximos passos
- Prog. estruturada versus Orientação a Objetos
- Classes e Objetos
- Atributos e Métodos
- Paralelo com conceitos anteriores  
Estruturas e funções
- Exercícios





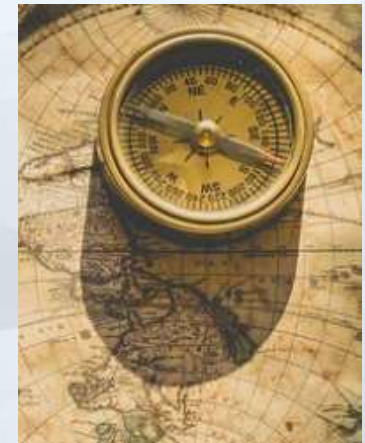
## Contexto

- Onde estamos?
- Foco agora!
- Cenário de exemplo

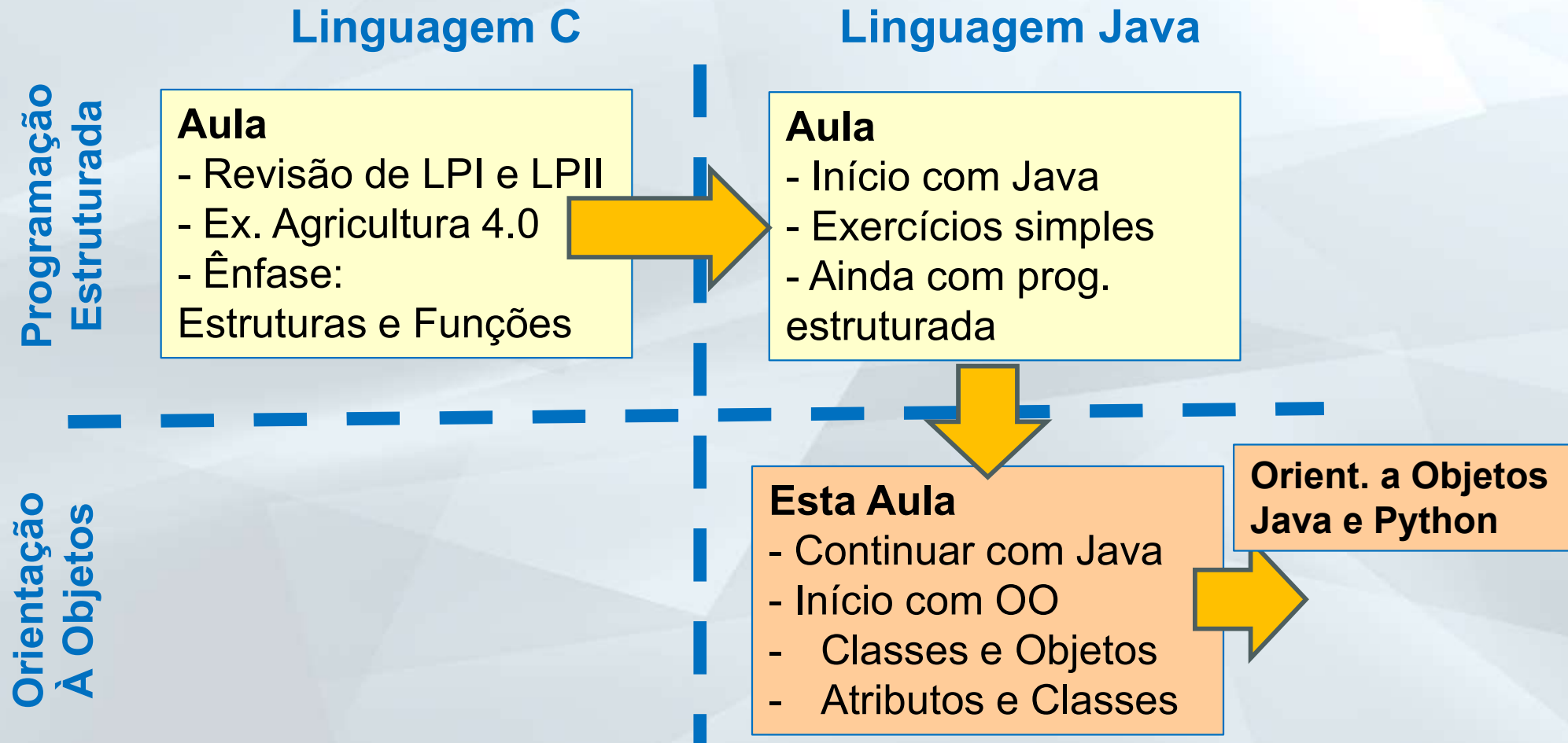
# Onde estamos?

- *Considerando nosso planejamento inicial*
- Capítulo 1 – Introdução
- Capítulo 2 - Conceitos básicos de Orientação a Objetos
  - Parte 1 : Classes, objetos, atributos e métodos
  - Parte 2 : Encapsulamento, modif. de acesso, getters e setters
  - Parte 3 : Construtores
  - Parte 4 : Relação entre classes
- Capítulo 3 – Herança e Polimorfismo
- Capítulo 4 – Classes abstratas e interfaces
- Capítulo 5 – Generics, collections e outros tópicos
- Capítulo 6 – Desenvolvimento de um projeto em OO

Foco



# Onde estamos?



# O que nos espera?

- Todo paradigma ou linguagem acaba tendo um roteiro comum de se aprender.
- Organizaremos nossa trajetória neste início pelas seguintes partes deste capítulo
  - **Parte 1:** Classes, objetos, atributos e métodos e discussões iniciais sobre orientação a objetos
  - **Parte 2:** Encapsulamento, modificadores de acesso, getters e setters e sobrecarga de métodos
  - **Parte 3:** Construtores e instanciação de objetos
  - **Parte 4:** Atributos e métodos de classe
  - **Parte 5:** Outros elementos básicos



# **Introdução à Orientação a Objetos**

- Como programamos?
- Relação com fluxograma
- Orientação a objetos
- Comentários



# Programação Estruturada

- Qual o foco principal quando pensamos um sistema da forma como fizemos até agora, através dos tópicos de programação estruturada
  - O foco principal é o organizar o encadeamento das instruções
  - Qual a sequência de comandos, em qual fluxo ocorrem?
  - Repense um dos problemas anteriores (por exemplo)
  - Como as instruções ocorrem no programa
- Enfim, o ponto chave é fluxo de controle
  - Sequencial: Uma instrução após a outra
  - Condicional: com uso dos comandos IF e Switch
  - Repetição: Com o uso dos comandos For e While
  - Modularização: que continua com uma visão de fluxo, mas agrupando instruções em seus módulos e





# Exemplos

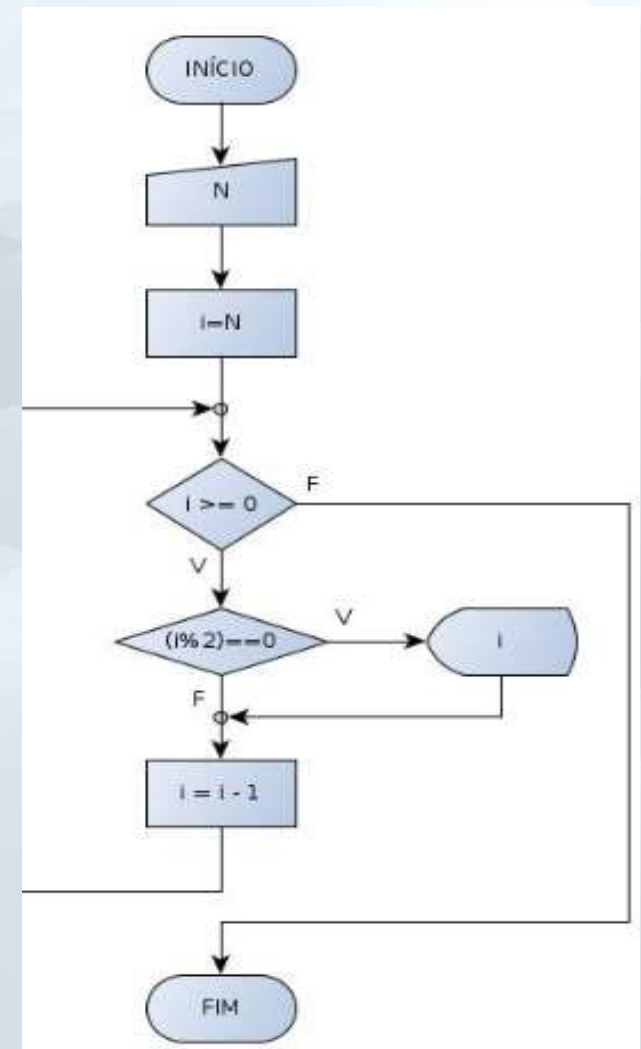
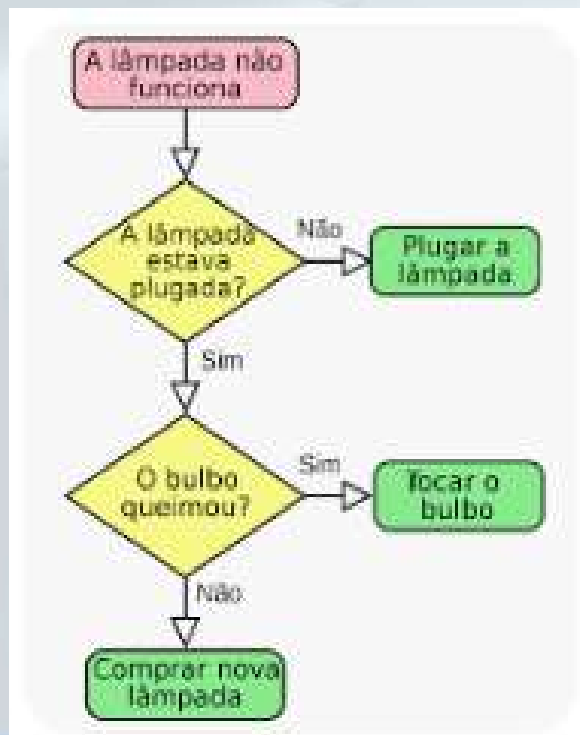
- Como programamos até agora?

Fluxo de  
Controle

```
63 char op;  
64 do{  
65     printf("\n Digite o valor do sensor:\n");  
66     scanf("%f", &s_umidade.valor);  
67  
68     if (s_umidade.valor < 30)  
69     { printf(" Necessário ligar bomba.\n");  
70       e_bomba.status = 'L';}  
71  
72     if (s_umidade.valor > 80)  
73     { printf(" Necessário desligar bomba.\n");  
74       e_bomba.status = 'D';}  
75  
76     printf("\n Digite a opção: (C)Continuar e (S).  
77     printf(" Opção: ");  
78     fflush(stdin);  
79     scanf("%c", &op);  
80 }while(op != 'S');
```

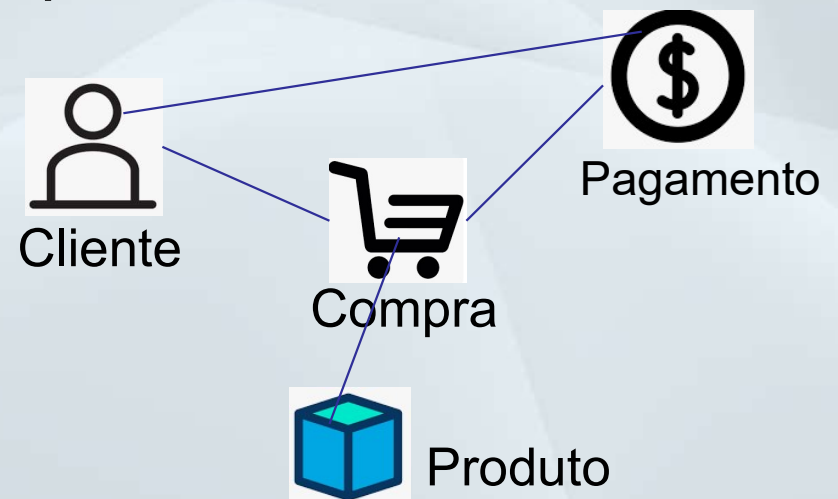
# Relação com fluxograma

- Se organizássemos nossos programas em um esquema, como ele seria?
- Em geral é comum usarmos um fluxograma quando estamos aprendendo.
- Ele representa visualmente bem esta noção de pensarmos em fluxo



# Orientação a objetos

- Seguindo um paradigma de programação orientada a objetos, como compreendemos um sistema que deve ser implementado?
  - O foco é concentrar-se nos objetos (classes) que compõem o domínio do problema.
  - Por exemplo, quais objetos compõem o domínio dos problema abaixo
    - Banco ou instituição financeira
    - Escola
    - Biblioteca
    - Site de Vendas pela Internet



# Exemplos

- Em um sistema de entrega de comidas (iFood, Uber eats e outros), quais as entidades ou objetos envolvidos?

- Restaurantes
- Itens do cardápio ou pratos
- Entregadores
- Pedidos

Cardápio

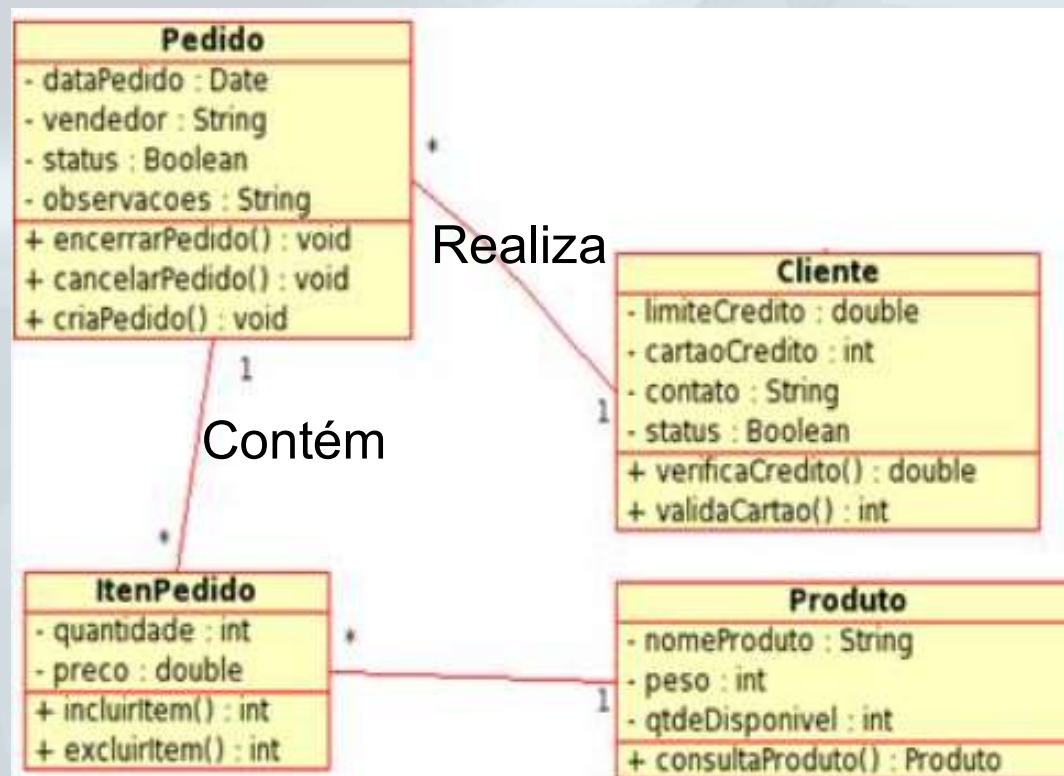
Clientes

- Como estes objetos se relacionam?
  - Clientes pesquisam restaurantes
  - Entregadores entregam pedidos, E assim por diante



# Relação com Diagrama de Classes/Objetos

- Na orientação a objetos, um diagrama expressivo é o de classes, que indicam que tipos de objetos se relacionam entre si



# Atenção: Ainda podem existir fluxos

- O fato de tentarmos compreender o sistema em questão por meio de seus objetos, não quer dizer que não haja fluxos nos cenários
- A abordagem de iniciar a compreensão e avaliar como será o sistema é que é diferente.
- Do mesmo modo, perceber que ao pensarmos um sistema via programação estruturada, não quer dizer que não existam as informações, entidades e objetos
- A abordagem de compreensão (inicial) é que se concentra no fluxo







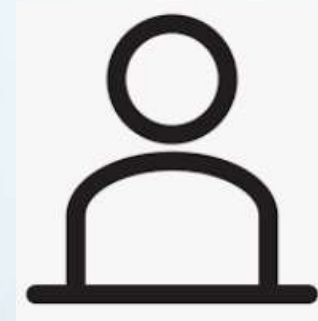
# Classes e Objetos

- Classes
- Objetos
- Relação com estruturas
- Atributos
- Métodos





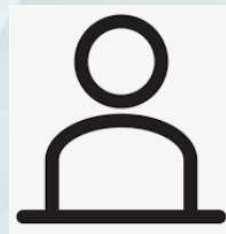
# Classes e Objetos



- O que é uma classe?
  - Tais objetos podem ser organizados em grupos de características e comportamentos comuns.
  - Uma classe define as características comuns a um grupo de objetos
  - Porém a classe não serve para organizar no sentido de guardar os objetos, ela serve de modelo de construção.
  - Para os nossos exemplos, todas as pessoas (João, Maria, Carlos) são da classe Pessoa, pois possuem as mesmas características (CPF, Nome e outros) por exemplo.

# Classes e Objetos

- O que é um objeto?
  - São entidades concretas, por exemplo, (João, Maria, Carlos) que possuem características e comportamentos bem definidos
  - É comum vermos também os termos instâncias ou exemplares, isto é Objetos são instâncias de uma classe
  - João, Maria, Carlos são exemplares ou instâncias da classe Pessoa



# Exemplo 1 – Sistema Bancário

## – Classes

- Conta                      Cliente                      Transação



## – Objetos

- Funcionam como exemplares ou instâncias da classe
- **Conta**                      Número: 23                      da Agência: 327
- **Conta**                      Número: 45                      da Agência: 456
- **Cliente:**                      Juvenilda
- **Cliente:**                      Teobaldo
- **Transação:**                      Saque                      Dia: 23/08                      Valor: 200,00
- **Transação:**                      Depósito                      Dia: 21/08                      Valor: 250,00

# Exemplo 1 – Agricultura 4.0

## – Classes

- Sensor

Equipamento



## – Objetos

- **Sensor** de Umidade: ID 10
- **Sensor** de Temperatura: ID 15
- **Sensor** de Velocidade: ID 21

Valor lido: 30%

Valor lido: 25°

Valor lido: 56km/h

- **Equipamento** Bomba ID 121
- **Equipamento** Drone ID 131
- **Equipamento** Trator ID 115

Status: Ligada

Status: Desligado

Status: Ligado

# Exercícios



- Para fixar a ideia de Classes e Objetos, vale discutir outros cenários.
- Cenário sugerido 1: Aplicativo de Mobilidade
- Cenário sugerido 2: Loja de Material de Construção
- Pense(m) agora você(s) em pelo menos três cenários
  - Quais classes?      Quais objetos?
- Em uma analogia “Molde” e “Exemplares”, ou no caso do “Formulário em branco” e “Formulário preenchido”, o que representa a classe e o que representa o objeto?

# Como programar?

## – Classe Conta

- Uma classe é definida, em geral, dentro de seu próprio arquivo
- Na versão mais simples, nela estabelecemos os seus campos ou atributos
- Como podem existir vários arquivos, o termo

## – Objetos

- Com base na classe, as variáveis e objetos podem ser criados
- Perceba que cada um com seus valores
- E cada um ocupa seu espaço em memória

```
11 package vl_Basico;
12
13 class Conta {
14     int Numero;
15     int Agencia;
16     String Titular;
17     double Saldo;
```

```
1 package vl_Basico;
2
3 public class Principal {
4
5     public static void main(String[] args)
6     {
7         Conta c1, c2;
8
9         c1 = new Conta();
10        c1.Numero = 10;    c1.Agencia = 25;
11        c1.Titular = "Carla";
12        c1.Saldo = 100.0;
13
14        c2 = new Conta();
15        c2.Numero = 12;    c2.Agencia = 25;
16        c2.Titular = "Francisco";
17        c2.Saldo = 120.0;
18
19        System.out.println("Numero : " + c1.Numero);
20        System.out.println("Numero : " + c2.Numero);
21    }
22 }
```



# Discussões: Classes e Tipos

- Qual relação destes conceitos com outros conhecimentos ligados a programação?
  - Tipos de dados e variáveis
    - Em C o que é um tipo de dados?
    - Quando declaramos :        `int x;`        `char c;`
    - O que é o tipo?
    - O que é a variável com o seu valor?
  - Estruturas (structs)
    - O que é uma estrutura em uma programação em C?
    - O que é uma variável com os campos da estrutura preenchidos.
    - `struct Ponto p1;`
- Tipos e estruturas definem um molde (características e operações) a qual variáveis do respectivo tipo devem adotar.
- Analogamente as classes são equivalentes aos tipos/estruturas e os objetos são equivalentes aos valores (memória alocada) nas variáveis.

```
struct sensor
{
    int    id;
    float  valor;
    char   grandeza[21];
};
typedef struct sensor TSENSOR;
```

```
public class Sensor
{
    int    id;
    String grandeza;
    float  valor;
}
```



# Discussões: Alocação da Memória

- Para aproveitarmos que estamos com este racioncínio “quente” em nossa mente
- Ainda sobre a comparação entre Classe em Orientação a Objetos e Estruturas (structs) em Programação Estruturadas (em C)
- **Quais as diferenças?**
  - **Alocação de memória**
    - Perceba que precisamos explicitamente usar o “comando” **new**
    - Neste momento que é criado um espaço em memória baseado nos campos da classe
  - **Nas estruturas**, só definir a variável do tipo estrutura e a **memória já é alocada**.
- **Métodos**
  - Veremos mais a frente que as classes na orientação acomodam os métodos, equivalente às funções em uma programação estruturada como em C

```
s_umidade = new Sensor();
```

s\_umidade



Id:	0
Grand:	""
Valor:	0.0

# Discussões: Dinâmica Formulário

- Imagine o seguinte cenário, um professor precisou fazer um levantamento dos alunos e seus e-mails e para isso, pensando que ele não considerou recursos tecnológicos, ele preparou em word o seguinte formulário:

Matrícula	<input type="text"/>
Nome:	<input type="text"/>
Email:	<input type="text"/>

- Depois ele imprimiu (ou reproduziu) 20 cópias deste formulário e pediu que seus alunos preenchessem:

Matrícula	20140234
Nome:	Teobaldo Bisânio
Email:	teo@gmail.com

Matrícula	20140246
Nome:	Juvenilda Clementina
Email:	juvezinha@hotmail.com

- Neste exemplo, considerando o molde (ou template) do formulário e os formulários preenchidos, o que representariam as classes e os objetos?

Matrícula

20140246

Nome:

Juvenilda Clementina

Email:

juvezinha@hotmail.com

# Atributos e Métodos

- Atributos
- Métodos



# Atributos

- Atributos

Matrícula	20140246
Nome:	Juvenilda Clementina
Email:	juvezinha@hotmail.com

- Armazenam algum dado ou informação associado a um objeto.  
Análogo ao que ocorre com estruturas em C.

- Exemplos (Aluno): Matrícula, Nome, Email e outros
- (Conta): Titular e Saldo
- (Bola): Peso, Diâmetro, Material

```
public class Sensor
{
    int    id;
    String grandeza;
    float  valor;
}
```

- Conceito de Estado de um objeto

- Cada objeto em determinado instante apresenta um estado que é definido pelos valores de seus atributos

- Cada formulário preenchido apresenta seus valores
- Uma conta bancária apresenta o seu saldo em determinado dia

Id:	0
Grand:	""
Valor:	0.0

# Métodos



- Os métodos desempenham um papel parecido com as funções em uma programação estruturada
- A diferença é que eles são parte de uma classe, isto é, a sua definição ocorre no escopo ou contexto de uma classe
- Eles são responsáveis por alguma ação a ser disparada para um objeto daquela classe
- Exemplos:
  - Em conta, podemos ter os métodos sacar e depositar
  - Em carro, podemos ter os métodos mover, ligar, desligar, abrir portamalas.

```
class Conta {  
    ...  
    double Saldo;  
    public void Sacar(double pValor)  
    {  
        this.Saldo -= pValor;  
    }  
    ...  
}
```

# Exercícios

- Para fixar a ideia de Classes e Objetos, vale discutir outros cenários.
- Cenário sugerido 1: Agricultura 4.0:
- Cenário sugerido 2: Aplicativo de Mobilidade
- Cenário sugerido 3: Loja de Material de Construção
- Pense(m) agora você(s) em pelo menos três cenários
  - Quais classes e atributos?      Quais métodos?
- Escolha alguns dos métodos escolhidos e tente indicar quais atributos do objeto/classe ele acaba modificando

# Discussão: Estruturas e Classes

- Apenas lembrando, é possível fazer uma analogia entre funções em programação estrutura C e métodos em Java.
- Métodos funcionaria como “funções que pertencem a uma certa classe”
- Mas vale reforçar o conceito, o método é associado a uma classe e seu conceito é que os objetos podem receber mensagens ou comandos que alteram suas informações
  - Uma conta número 23 com saldo 300, pode receber a chamada do método Sacar(100) e ficar com saldo de 200



# Discussão: Estruturas e Classes

- Em estruturas
  - Sacar definido em uma função que acessa a estrutura
  - Problema:
    - Sacar fica num escopo mais geral e o campo saldo pode ser acessado de qualquer parte do programa
    - Qualquer parte do programa pode acessar o campo saldo e isso pode gerar inconsistências.
    - Por exemplo, se alguém abate algum valor de saldo sem o devido uso da função Sacar, o saldo da conta fica inconsistente.
- Em classes e objetos
  - O Sacar é implementado como um **método de uma classe conta**
  - Com isso o valor de saldo pode ficar dentro de uma classe Conta e seu valor pode ser regido pelo método Sacar
  - Se usado os devidos mecanismos para isolar (verá na frente o encapsulamento), haverá uma maior qualidade da informação quanto ao valor contido no campo Saldo.

# Como programar?

- Observe o código a seguir:
- Quais métodos estão codificados?
- Qual atributo eles alteram?
- Qual semelhança em relação às funções em C?
- Qual diferença?

```
13  class Conta {
14      int Numero;
15      int Agencia;
16      String Titular;
17      double Saldo;
18
19      public void Sacar(double pValor)
20      {
21          Saldo -= pValor;
22      }
23
24      public void Depositar(double pValor)
25      {
26          Saldo += pValor;
27      }
28
29      public void Imprimir()
30      {
31          System.out.println("Numero : " + Numero);
32          System.out.println("Agencia : " + Agencia);
33          System.out.println("Titular : " + Titular);
34          System.out.println("Saldo : " + Saldo);
35      }
36  }
```

# Como programar?

- Como invocar os métodos?

- Perceba que o método é chamado atrelado ao objeto

- Neste momento é que conceitualmente o objeto recebe a mensagem que precisa executar o método?

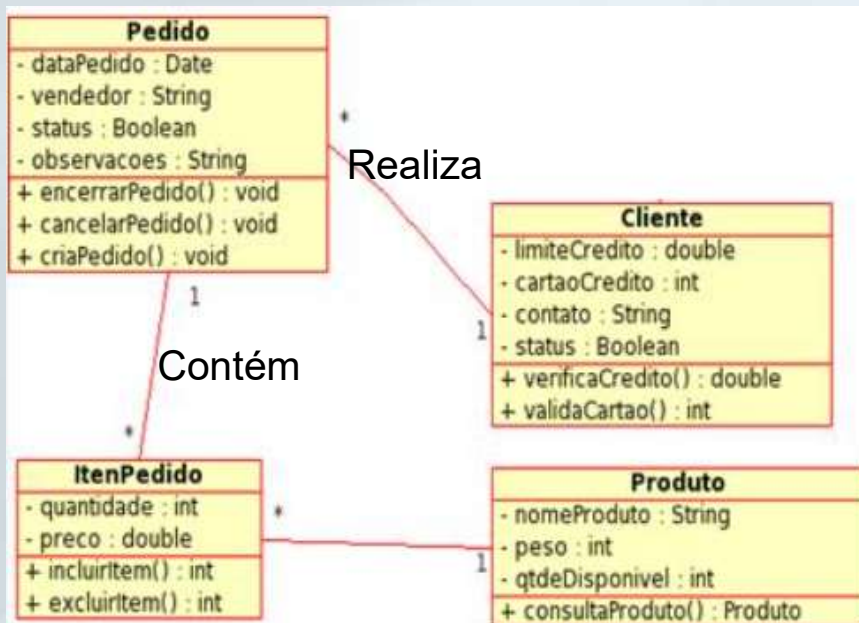
```
1 package v1_Basico;
2
3 public class Principal {
4
5     public static void main(String[] args)
6     {
7         Conta c1, c2;
8
9         c1 = new Conta();
10        c1.Numero = 10;    c1.Agencia = 25;
11        c1.Titular = "Carla";
12        c1.Saldo = 0.0;
13
14        c1.Depositar(1000.0);
15        c1.Sacar(300);
16        c1.Imprimir();
```

- Lembra que falamos que na Orientação a Objetos os objetos trocam mensagens entre si, é como se o objeto em C1 recebesse as mensagens de Depositar, Sacar e Imprimir

Matrícula	20140246
Nome:	Juvenilda Clementina
Email:	juvezinha@hotmail.com

## Aprofundando conceitos

- O pensar em entidades!



# Discussões finais

- Este paralelo Entidade e seus Campos vão estar presentes em uma série de coisas na computação
- Por que precisamos pensar em “Entidades” (Classes, Estruturas, Tabelas, ...) e seus Atributos (Campos ou partes)?
- Nosso papel como programados será criar sistemas que lidam com informações, e pensar nas entidades e seus campos é natural para estruturarmos estas informações
- Daí, em diversas partes ou componentes que formam um sistema ou em diversas disciplinas do curso que aprendemos sobre esta parte, teremos esta ideia
  - Em programação (LP1 e LP2)      Estruturas
  - Em LPIII      Classes
  - Em banco de dados      Entidades, Relações, Tabelas



## Conclusões

- Retomando Plano de Aula
- Revisão
- Para saber mais



# Conquistamos a aula

- ***Objetivos***

- Iniciar o aprendizado de Orientação a Objetos
- Compreender os conceitos básicos de Classes, Objetos, Atributos e Métodos

- ***Tópicos***

- Contexto da disciplina
- Foco agora e próximos passos
- Prog. estruturada versus Orientação a Objetos
- Classes e Objetos
- Atributos e Métodos
- Paralelo com conceitos anteriores  
Estruturas e funções
- Exercícios





# Sugestão de estudos

- **Reforço com algumas pesquisas**
  - Pesquisa sobre Projeto e Arquivos no NetBeans.
- **Faça uma leitura de um livro ou de um bom site da Internet para ampliar sua compreensão**

Apostila do Alura      Livro do Deitel      w3schools.com
- **Se quiser assuntos independentes mas curiosos, veja algum vídeo sobre cenários de tecnologia e discuta quais Classes e Objetos estão relacionados**
  - Agricultura 4.0
  - Internet das Coisas
  - Inteligência Artificial
- **Após estas “viagens” e explorações, faça as listas de exercícios mais diretas, práticas para explorar os assuntos do dia a dia da disciplina.**

Departamento de Engenharias e Computação  
Colegiado de Ciência da Computação  
Disciplina: Linguagem de Programação III



Cap 2 – Conceitos Básicos de Orientação a Objetos

## **Parte 1 – Classes, Objetos, Atributos e Métodos**

Professor: Otacílio José Pereira