Programação Orientada a Objetos

Aula 01: Motivação

Ivo Calado

Instituto Federal de Educação, Ciência e Tecnologia de Alagoas

28 de Maio de 2016

Roteiro

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA ALAGOAS

Relembrando um pouco programação estruturada

Suponha que seja solicitado o desenvolvimento de um programa em Python para gerência de uma biblioteca com as seguintes funcionalidades:

1. Cadastrar funcionários (nome e salário)
2. Apagar funcionários cadastrados
3. Listar funcionários
4. Exibir salário com gratificação (25%)
5. Sair

Como seria uma possível solução?

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA ALAGOAS

Solução I

```
while True:
        print("### Controle de Cadastro de Funcionarios
            ###")

        print("1 - Cadastrar")
        print("2 - Apagar")
        print("3 - Mostrar")
        print("4 - Exibir salario com gratificacao")
        print("5 - Sair")

        opc = int(input("Digite o numero da sua opcao:
            "))

        if opc==1:
                nomes.append(input("Digite o nome a ser
                    cadastrado:"))
```

Solução II

```
                salarios.append(float(input("Digite o
                    salario:")))
        elif opc==2:
                nome = input("Informe o nome a ser
                    apagado:")
                i=0
                for n in nomes:
                        if n==nome:
                                del(nomes[i])
                                del(salarios[i])
                        i=i+1
(...)
```

Mas como evoluir o software?

E se após o desenvolvimento, fosse solicitado que algumas funcionalidades extras fossem adicionadas:

- Como seria possível adicionar mais 3 atributos (endereço, CPF e data de nascimento)?

Mas como evoluir o software?

E se após o desenvolvimento, fosse solicitado que algumas funcionalidades extras fossem adicionadas:

- Como seria possível adicionar mais 3 atributos (endereço, CPF e data de nascimento)?
- E se fosse pedido para adicionar mais algumas entidades como Livro (titulo, edição, ISBN), Autor (nome, endereço, CPF), Locador (nome, endereço, CPF), Locações, ...?

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA ALAGOAS

Mas como evoluir o software?

E se após o desenvolvimento, fosse solicitado que algumas funcionalidades extras fossem adicionadas:

- Como seria possível adicionar mais 3 atributos (endereço, CPF e data de nascimento)?
- E se fosse pedido para adicionar mais algumas entidades como Livro (titulo, edição, ISBN), Autor (nome, endereço, CPF), Locador (nome, endereço, CPF), Locações, ...?
- E se alguns dos atributos fosse na verdade uma lista (ex.: lista de telefones de uma pessoa)
- E se fosse solicitado que o sistema adicionasse tipos específicos de funcionários, como Gerente (gratificacaoAdicional), Faxineiro (setorDeTrabalho), Bibliotecário (titulacaoAdicional) com algumas propriedades específicas?

Quais as consequências? I

Em termos de código, a mais imediata seria a criação de vários vetores para armazenar as informações:

```
nomesFunc = [] #dados do funcionário
salariosFunc = []
enderecosFunc = []
cpfsFunc = []
dataNascFunc = []
titulos = [] #dados dos livros
edicoes = []
isbns = []
nomesAutores = [] #dados dos autores
enderecosAutores = []
cpfsAutores = []
[...]
```

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA ALAGOAS

Quais as consequências? II

Quais os problemas de tal abordagem?

- Faz-se necessário a definição de vários vetores para armazenar as informações das diversas entidades
- Por todos os vetores estarem no mesmo escopo, faz-se necessário inventar nomes diferentes para o mesmo tipo de propriedade
- Não há uma ligação explícita entre os vetores para compor as entidades (Funcionario, Livro, Autor, Locador, Locações)
- Como seria a implementação de relacionamento entre essas entidades?
- Para as entidades derivadas não há uma maneira direta de representar as entidades sem duplicar os vetores de dados

Quais as consequências? III

O que podemos concluir?

- Conforme o software aumenta de tamanho a maior complexidade se concentra mais na modelagem dos dados e em seus relacionamentos e menos no algoritmos
- Programação estruturada não provê alguns elementos básicos para possibilitar a modelagem dos dados

Definição I

Definição

POO é um paradigma de programação onde um programa consiste de uma rede de objetos se comunicando

- O programa passa a ser composto por um conjunto de **entidades** que se comunicam
- Cada **entidade** é chamada de **objeto** no domínio de POO

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA ALAGOAS

Definição II

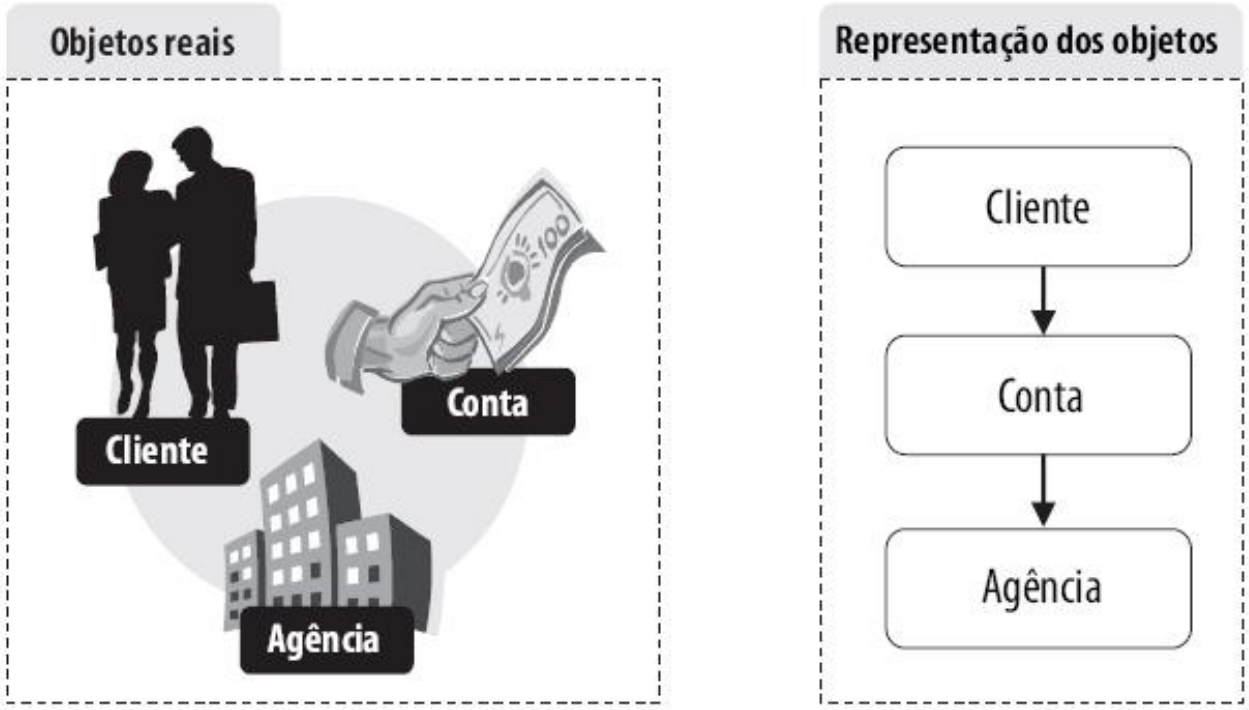


Figura: Linguagens OO oferecem suporte explícito para representar objetos do mundo real em software

Definição III

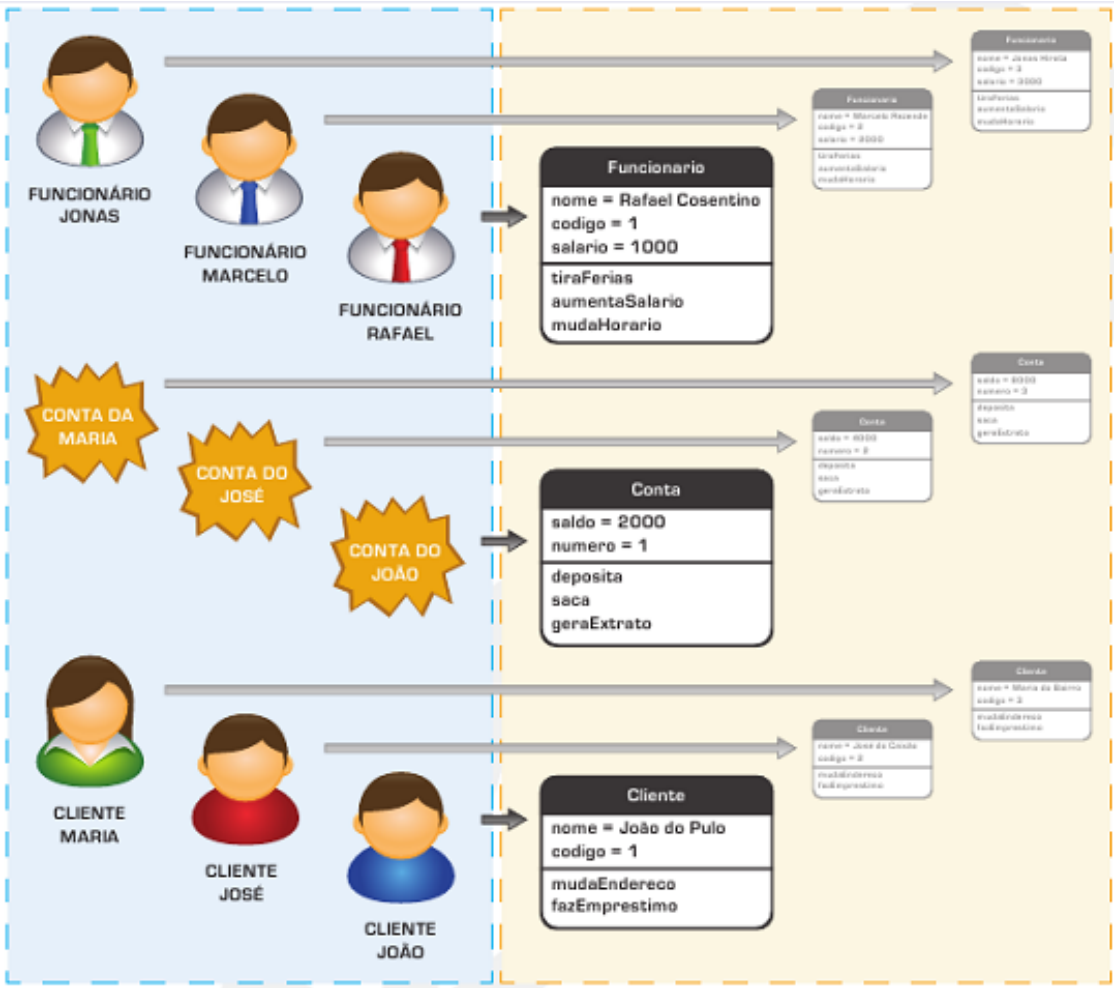


Figura: Objetos reais mapeados para objetos no paradigma OO

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA ALAGOAS

Mas como podemos representar um objeto?

Linguagens OO geralmente oferecem dois recursos para representarmos objetos, que são **atributos** e **métodos**

- **Atributos:** armazenam dados do objetos. Também conhecidos como estados do objetos
- **Métodos:** correspondem as **funções** do objeto
  - Podem manipular os valores dos atributos
  - Possbilitam interagir com outros objetos
  - Uma requisição de execução de um método é chamada **mensagem**

Mas como podemos representar um objeto?

Exemplo 1: como poderíamos representar modelar uma lâmpada em termos de atributos e métodos?

- **Atributos:**
  - EstaLigado: [Verdadeiro/Falso]
- **Métodos:**
  - Ligar
  - Desligar

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA ALAGOAS

Mas como podemos representar um objeto?

Exemplo 2: como poderíamos representar modelar uma lâmpada com suporte a ajuste de potência, em termos de **atributos** e **métodos**?

Mas como podemos representar um objeto?

Exemplo 2: como poderíamos representar modelar uma lâmpada com suporte a ajuste de potência, em termos de **atributos** e **métodos**?

- **Atributos:**
  - EstaLigado: [Verdadeiro/Falso]
  - PotênciaAtual: [*Watts*]
- **Métodos:**
  - Ligar
  - Desligar
  - AjustarPotência(novaPotência)

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA ALAGOAS

Mas como podemos representar um objeto?

Qual a diferença entre uma função na programação estruturada e um método de um objeto na OO?

Funções na programação estruturada não estão associadas a nenhum objeto enquanto que métodos **pertencem** a um objeto

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA ALAGOAS

Quais informações modelar sobre um objeto?

Suponha que fosse solicitado a modelagem de uma pessoa. Quais atributos deveriam ser considerados?

Quais informações modelar sobre um objeto?

Suponha que fosse solicitado a modelagem de uma pessoa. Quais atributos deveriam ser considerados?
Algumas propriedades "óbvias" poderiam ser:

- 1 - nome
- 2 - endereco
- 3 - cpf
- 4 - dataNascimento

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA ALAGOAS

Quais informações modelar sobre um objeto?

Porém, poderíamos considerar algumas outras propriedades, como:

- 5 - peso
- 6 - altura
- 7 - cor dos olhos
- 8 - cor dos cabelos
- 9 - dias que não toma banho
- 10 - tamanho da unha do dedão do pé

Quais propriedades deveríamos considerar?

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA ALAGOAS

Quais informações modelar sobre um objeto?

Porém, poderíamos considerar algumas outras propriedades, como:

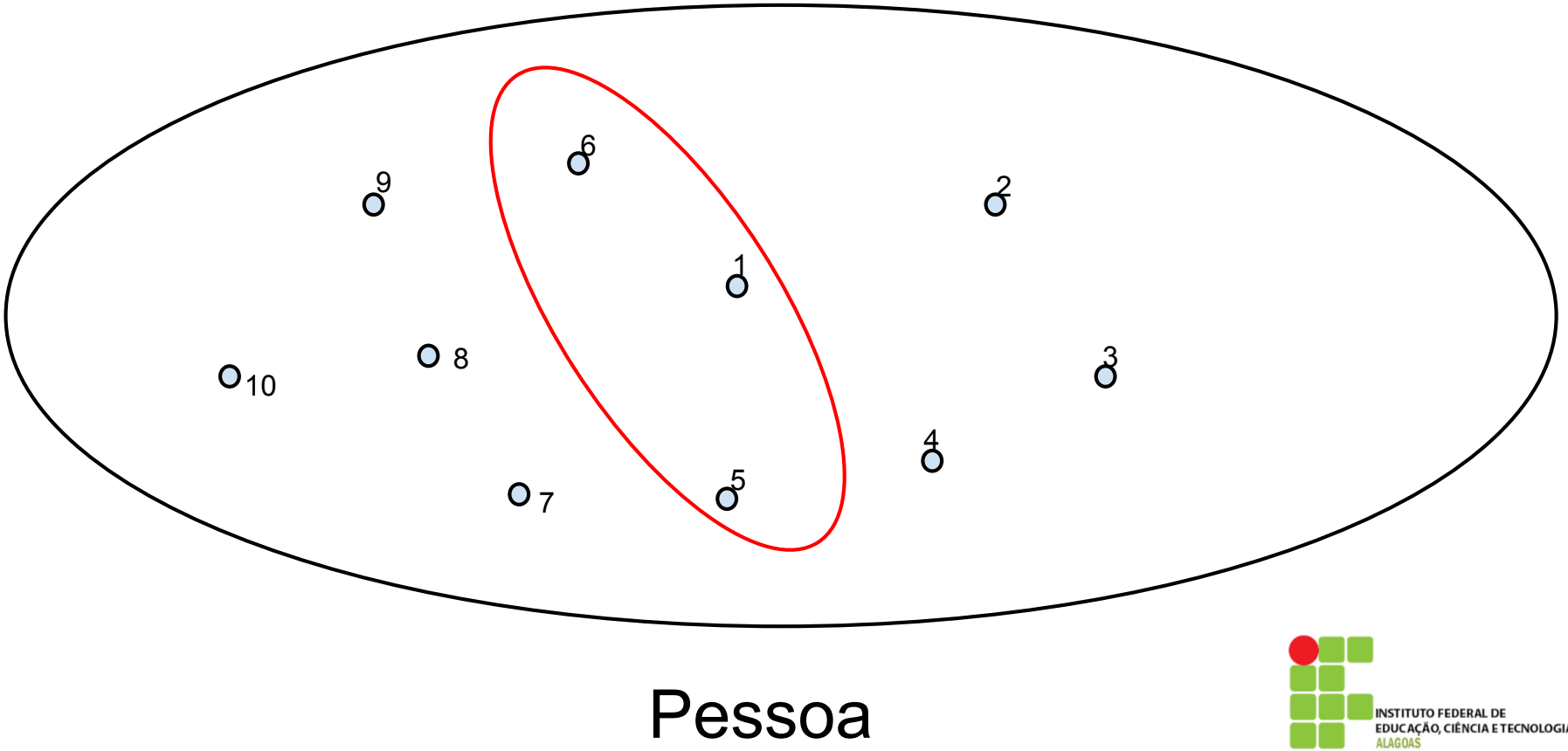
- 5 - peso
- 6 - altura
- 7 - cor dos olhos
- 8 - cor dos cabelos
- 9 - dias que não toma banho
- 10 - tamanho da unha do dedão do pé

Quais propriedades deveríamos considerar?

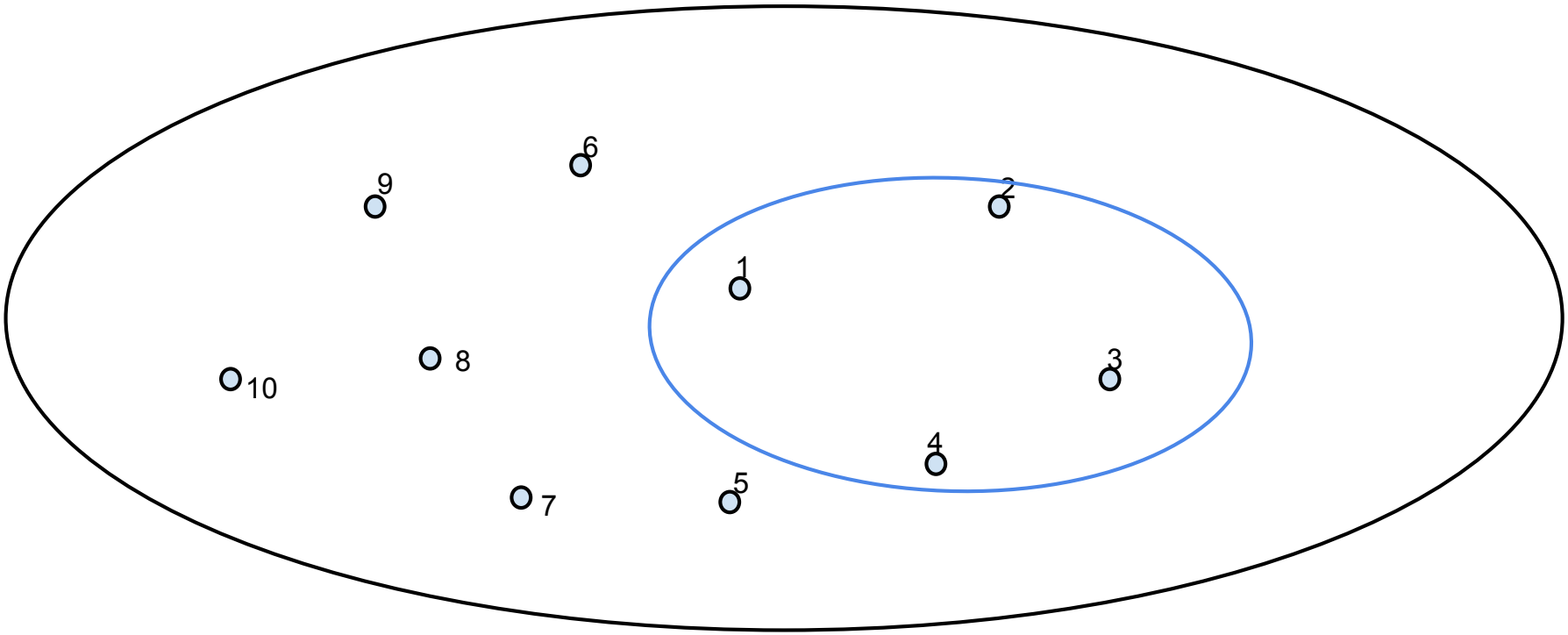
Depende do **domínio** do problema!

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA ALAGOAS

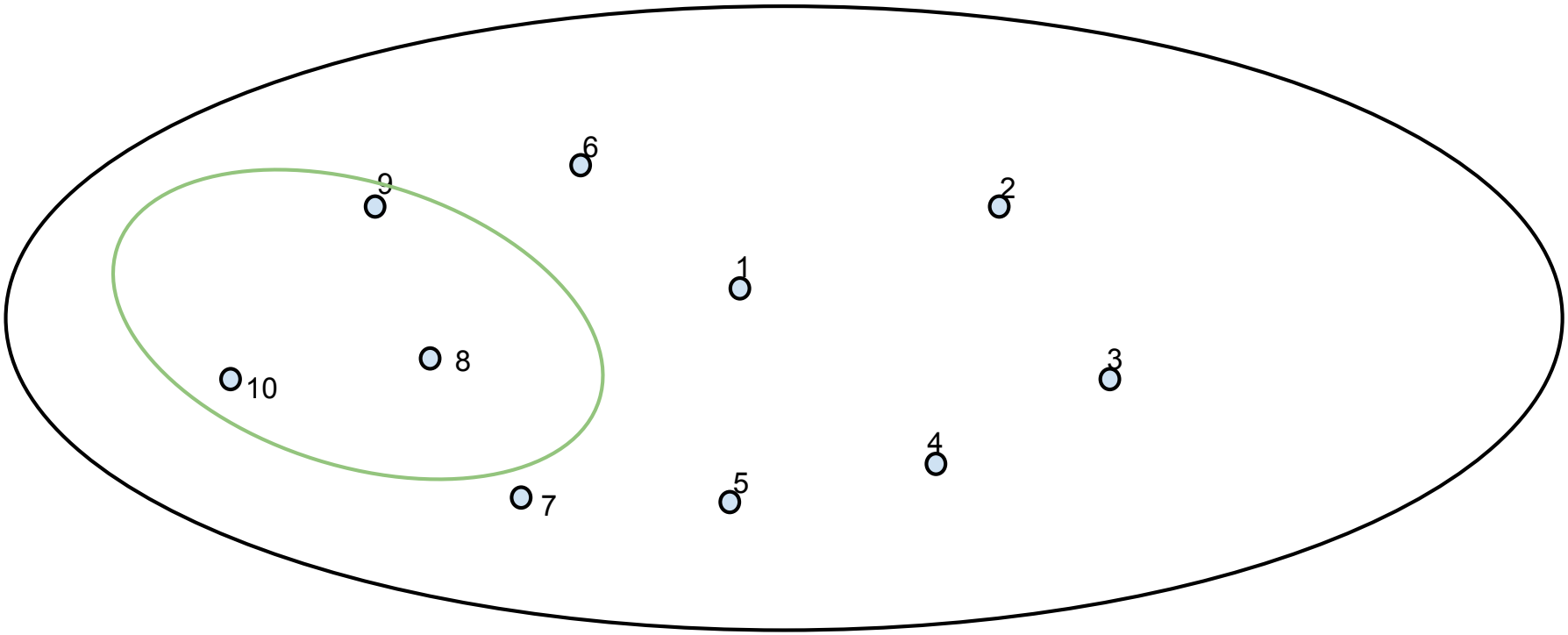
Abstraindo atributos I



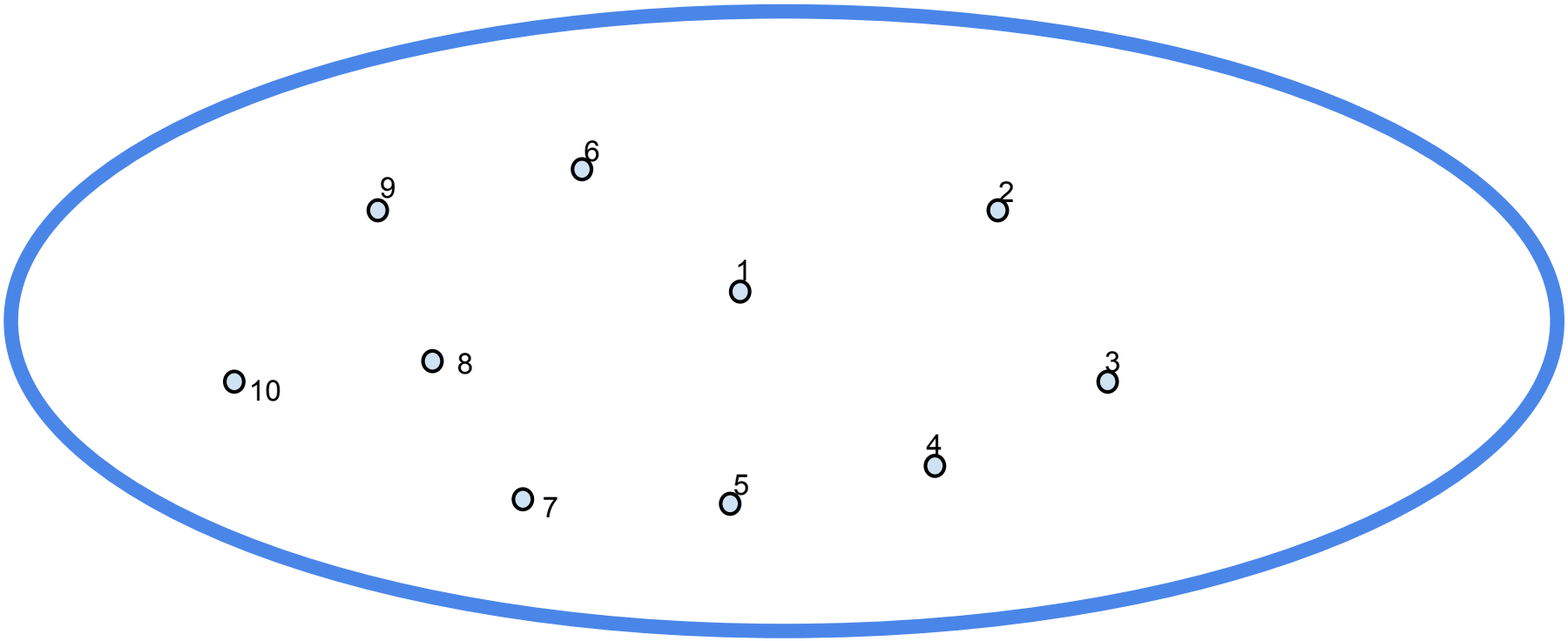
Abstraindo atributos II



Abstraindo atributos III



Abstraindo atributos IV



Pessoa

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA ALAGOAS

Seleção das propriedades importantes = Abstração

Abstração

"A **abstração** *é o processo de filtragem de detalhes sem importância do objeto, para que apenas as* **características apropriadas** *que o descrevem permaneçam"*

Seleção das propriedades importantes = Abstração

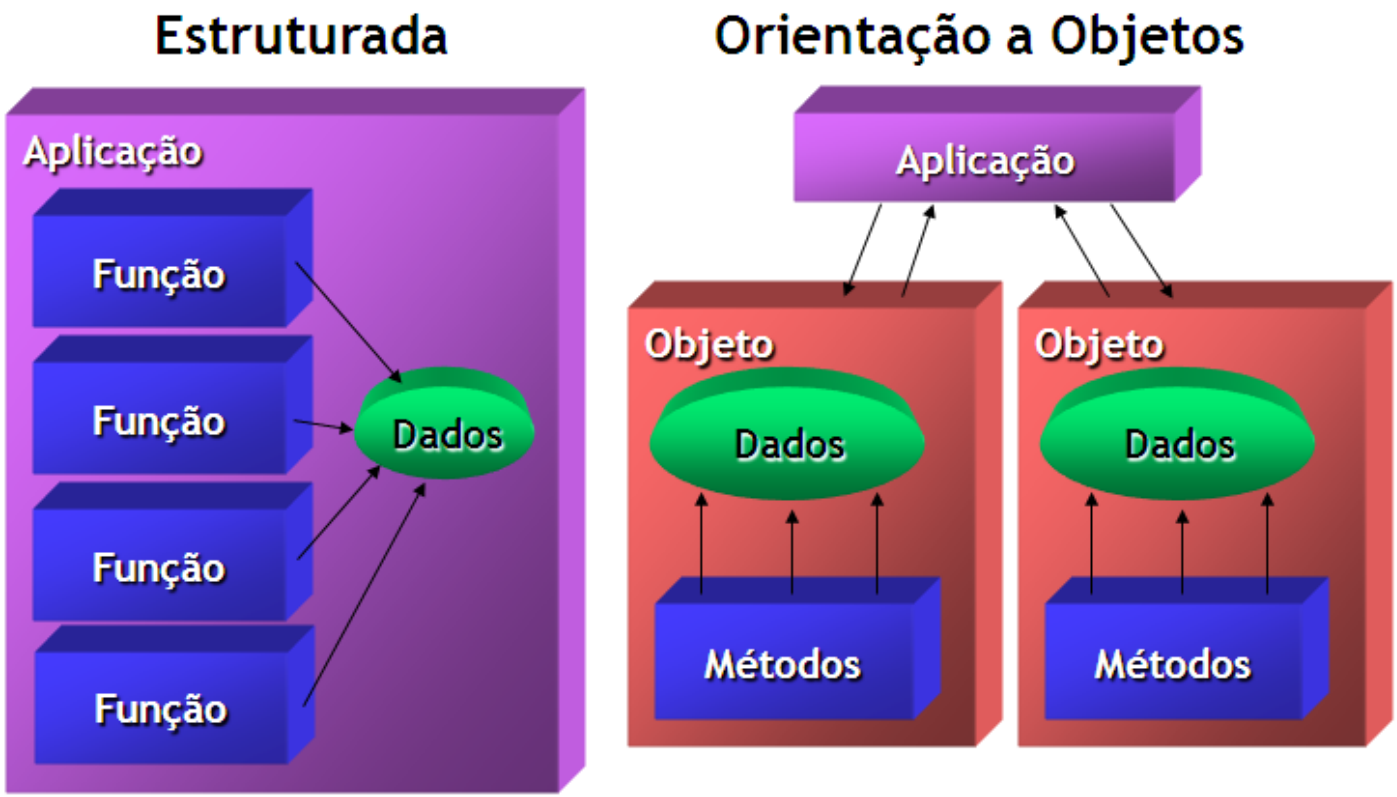
Abstração

"A **abstração** *é o processo de filtragem de detalhes sem importância do objeto, para que apenas as* **características apropriadas** *que o descrevem permaneçam"*

- As **características apropriadas** sempre dependem do **domínio** que está sendo trabalhado

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA ALAGOAS

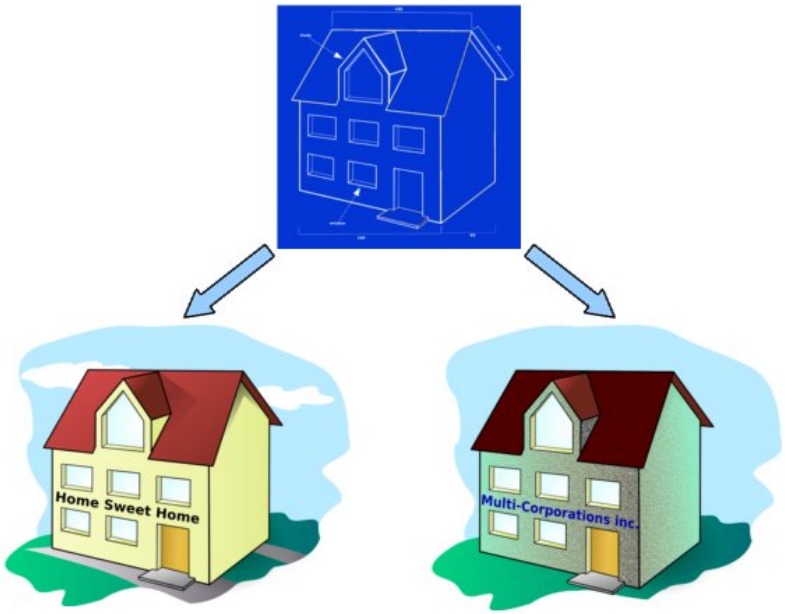
Resumo: POO x Programação Estruturada



INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA ALAGOAS

Classes I

- Antes de um objeto ser criado devemos projetá-lo



INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA ALAGOAS

Classes II

- Cada classe deve pode ter um conjunto de **atributos** e **métodos**
  - Atributos representam as **propriedades** do tipo a ser criado
  - Métodos representam as **ações** que o tipo pode realizar
- Um objeto é a "concretização de uma classe" **Um objeto é uma instância de uma classe**

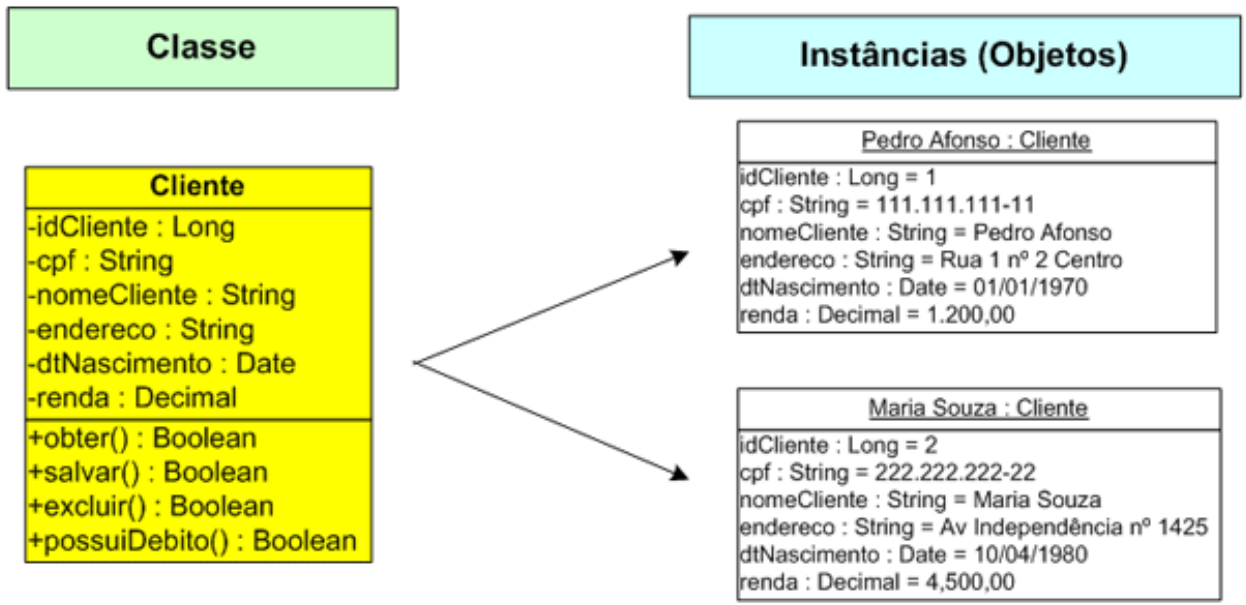
Fazendo uma analogia com uma tabela

- A **tabela** seria a **classe**
- Cada linha da tabela (**tupla**) seria um **objeto**

- Todas as instâncias de uma classe têm os mesmos métodos e atributos mas podem ter valores diferentes

Como representar classes e objetos?

- Podemos representar classes e objetos com o diagrama de classes UML
- Neste tipo de diagrama é possível modelar diversos detalhes da implementação como navegabilidade, tipo dos atributos, etc



Objetos (Instâncias) da classe Cliente

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA ALAGOAS

Exemplo de UML

- O diagrama UML se assemelha ao modelo ER, porém nele é possível modelar os métodos de cada classe