

Programação Orientada a Objetos

CIÊNCIA DA COMPUTAÇÃO E ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

Aula 01

Prof.ª ERIKA MIRANDA E PROF. CLAUDIO BONASSI

Prof. ^a Erika Miranda

- Mestre em Ciência e Tecnologia em Saúde(UMC)
- Especialista em Consultoria em Informática na Educação (UNICSUL)
- Bacharel em Ciências da computação (UNICSUL)



Conhecendo a turma

Quem são vocês??







- DISCIPLINA Programação
 Orientada a Objetos
- PRESENCIAL 36 Aulas



Ementa

Compreensão dos conceitos e aplicações do paradigma da Programação Orientada a Objetos, por meio do desenvolvimento de programas.



Cognitivos

- Estudar e conhecer os conceitos básicos do paradigma da programação orientada a objeto;
- Estudar e conhecer técnicas de programação orientadas a objeto;
- Estudar e conhecer as funcionalidades de ambientes de desenvolvimento de programas utilizados;
- Estudar e conhecer ferramentas para o desenvolvimento de programas.

Habilidades

- Utilizar as potencialidades da linguagem de programação selecionada;
- Elaborar programas sob o paradigma da programação orientada a objetos;
- 3. Utilizar ferramentas computacionais para o desenvolvimento de programas;
- 4. Comparar as linguagens orientadas a objetos com as linguagens de outros paradigmas de programação.

Atitudes

- Realizar projetos atuando no processo de soluções de problemas organizacionais que abranjam sistemas de informação levando em conta as dimensões organizacionais, tecnológicas e humanas;
- Valorizar e saber utilizar as novas tecnologias de informação;

Atitudes

- Ter criatividade e iniciativa diante da solução de problemas ;
- 4. Abstrair e organizar o raciocínio lógico;
- Ser crítico, receptivo e estar preparado para o trabalho em equipes ou coletivos de pesquisadores e programadores;

Atitudes

- Saber procurar e avaliar informação técnico-científica nas diferentes fontes escritas e eletrônicas disponíveis.
- 7. Ser analítico e responsável.

Unidade 1

Apresentação da disciplina.

- Plano de Ensino e Conceitos Básicos Apresentação e discussão do Plano de Ensino, focando objetivos, conteúdos, estratégias, avaliação e bibliografia.
- Ambientes de desenvolvimento de programas (IDE). Paradigmas de Programação. Conceitos fundamentais da POO.

Unidade 2 Conceitos básicos de Programação Orientado a Objetos Classes.

- Objetos. Atributos e métodos de uma classe.
- Encapsulação.
- Modificadores de acesso na linguagem utilizada. Métodos construtores.
- Mensagens e métodos.

Unidade 3 Variáveis e Métodos Variáveis e escopo.

- Tipos de dados (primitivos e abstratos).
- Conceito de métodos.
- Passagem de parâmetros e retorno.

Unidade 4 Métodos com estrutura de decisão Expressões.

- Estruturas de decisão simples e composta.
- Estruturas de decisão aninhada

Unidade 5 Métodos com estrutura de repetição

- Variáveis de controle.
- Estruturas de repetição aninhada.
- Tipos de estruturas de repetição.

Unidade 6 Métodos que recebem e retornam vetores e matrizes

- Conceitos sobre vetores e matrizes. Processamento de vetores e matrizes.
- Atributos de classes como vetores e matrizes.
- Vetores de objetos.
- Recebendo e retornando vetores e matrizes em métodos.

Unidade 7 Conceitos avançados de POO

- Sobreposição de métodos.
- Interfaces.
- Classes Abstratas.
- Herança.
- Composição.
- Polimorfismo.

Unidade 8

- Avaliações
- Provas parcial e regimental, trabalhos em grupo e atividades individuais

Processo Avaliativo

O processo avaliativo compreenderá:

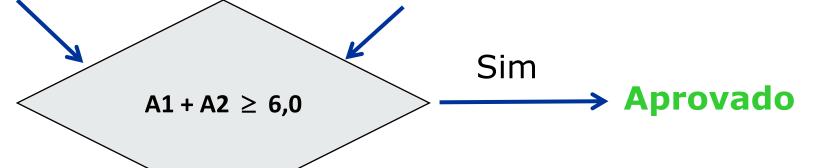
Avaliação Regimental (A1) no valor de 0,0 a 5,0.

Avaliações parciais e processuais (A2) no valor de 0,0 a 5,0.

A Nota Final (NF) resulta da soma destas duas notas (A1 A2).

A1 – Prova Regimental (5,0 pontos) A2 – Outras avaliações da disciplina (5,0 pontos)

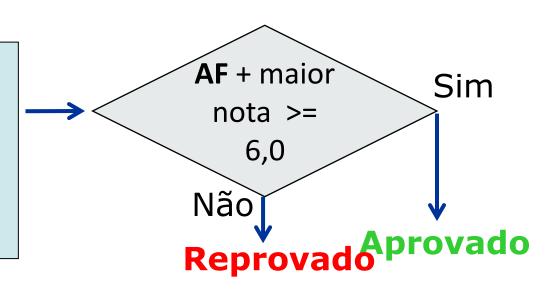
Processo Avaliativo



O aluno que não conseguir pelo menos 1 ponto na A1 ou A2 estará reprovado e não poderá fazer a AF

Avaliação Final
(AF), 5,0 pontos,
substituindo a menor
nota, seja A1 ou A2, caso
exista possibilidade de
aprovação com esta nota

↓ Não



Processo Avaliativo

É considerado aprovado na disciplina o aluno que obtiver Nota Final (NF) igual ou superior a 6,0 (seis) e que tenha, no mínimo, 75% (setenta e cinco por cento) de frequência às atividades acadêmicas.

Mais informações sobre o processo avaliativo podem ser obtidas no Manual do aluno e com os respectivos professores das disciplinas

Bibliografia Básica

- DEITEL, H. M. Java: Como Programar. 8. ed. Porto Alegre: Bookman, 2011.
- GAMMA, E. et al. Padrões de Projeto: Solucões Reutilizáveis de Software
 Orientado a Objetos. Porto Alegre: Bookman, 2007.
- HORSTMANN, C. S. Core **Java 2: Fundamentos. 7. ed**. São Paulo: Makron Books do Brasil, 2005.

Bibliografia Complementar

- AGUILAR, LUIS JOYANES Fundamentos de Programação: Algoritmos, estruturas de dados e objetos. Porto Alegre: Grupo A, 2001. (e-book).
- ARNOLD, Ken; GOSLING, James; HOLMES, David. A Linguagem de Programação Java 4ª edição. Porto Alegre: Grupo A, 2001. (e-book).
- PAGE-JONES, MEILIR. Fundamentos do Desenho Orientado a Objeto com UML. São Paulo: Pearson, 2001. (e-book).
- SINTES, ANTHONY Aprenda Programação Orientada a Objetos em 21 dias. São Paulo: Pearson, 2002. (e-book)

Programação Orientada a Objetos

Metodologia:

- Aulas expositivas
- Exercícios de aplicação prática
- Debates, Seminários, etc.



Softwares









Java SE SDK

http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html

NetBeans

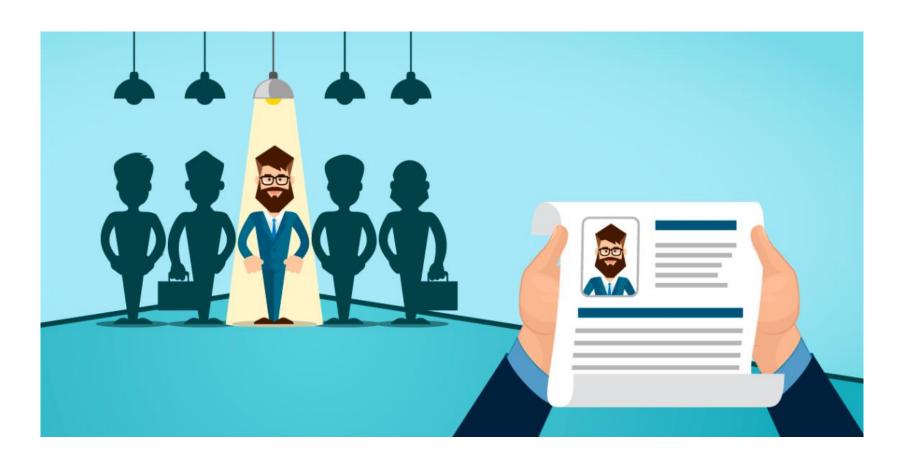
www.netbeans.org

Eclipsewww.eclipse.org

JCreator www.jcreator.com



Como está o Mercado de Trabalho



Para quem atua ou deseja atuar no setor de tecnologia, é pertinente saber que o mercado de trabalho é um dos menos atingidos pela crise dos últimos tempos.



Nesse cenário, uma das profissões que se mantém em alta é a de programador — profissional que atua diretamente na área de Tecnologia da Informação (TI).



Também há a possibilidade de aplicação desses conhecimentos em web, em dispositivos móveis, como celulares e aplicativos, e em jogos e videogames.



Um programador web, por exemplo, desenvolve sistemas e aplicativos para a internet, criando sites, lojas virtuais, portais etc.





Atualmente, a Catho, um dos maiores portais de empregos do país, mostra que entre os cargos com mais vagas abertas estão os de analista e técnico de suporte, além de desenvolvedor e de programador, com salários considerados acima da média.



As empresas brasileiras e internacionais que atuam no país estão em crescimento, o que torna o mercado favorável para os profissionais da Tecnologia da Informação.



As inovações tecnológicas crescem, e a procura por esse tipo de profissional também tende a receber uma demanda maior.

Por isso, é imprescindível ter um foco e se especializar o máximo possível.

Quais são os principais desafios da profissão?



Em boa parte das profissões, as pessoas adquirem um bom volume de conhecimento durante a faculdade, sentindo-se preparadas para o mercado de trabalho assim que saem dela.



Porém, na área de programação, o estudo precisa ser constante.

Quem escolhe entrar nesse setor precisa estar

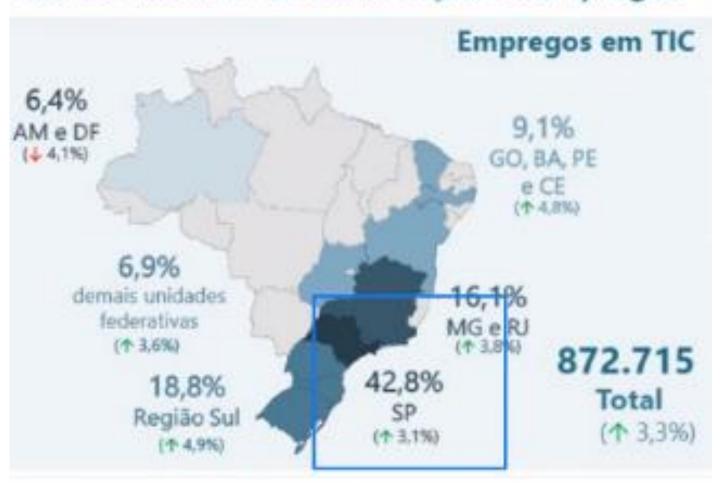
disposto a isso.



É importante ter em mente que ninguém nasce sabendo tudo e que as oportunidades de aperfeiçoamento existem, basta procurá-las.



Retrato nacional da distribuição de empregos





Um levantamento da Brasscom indica que o mercado de tecnologia no Brasil vai gerar 797 mil vagas de emprego até 2025. A média, segundo o levantamento, é de 150 mil postos gerados por ano no país pela área. Outro dado do estudo mostra que a remuneração dos profissionais de TIC (Tecnologia da Informação e Comunicação) é quase 3 vezes maior do que a média de trabalhadores de outros setores.

Está realmente mais difícil preencher as vagas na área de tecnologia?

O mercado de TI, Tecnologia da Informação, é de longe um dos que mais geram oportunidades de emprego. Seja pela sua variedade de setores, como pela necessidade de profissionais da área em praticamente todas as empresas, o que não falta são vagas para quem quer investir nessa carreira.

Está realmente mais difícil preencher as vagas na área de tecnologia?

Entretanto, há uma série de empecilhos que fazem com que surja uma enorme carência de profissionais capacitados.

Seja pela necessidade de investir em cursos e treinamentos para se qualificar às vagas, como pela falta de incentivo em escolas de nível Fundamental e Superior, para preparar os alunos para este caminho.

"...Em média o Brasil capacita 46.000 pessoas por ano para trabalhar em TI, isso em cursos superiores (graduação e tecnólogos) e técnicos, mas há uma demanda para 70.000 profissionais por ano.

O profissional de TI pode atuar em diversas áreas, por isso é necessária atualização profissional e um olhar dinâmico".

Dois grupos de habilidades em processos seletivos:



Hard Skills

- São habilidades que podem ser aprendidas e facilmente quantificadas.
- Em outras palavras, elas são tangíveis. Vistas na universidade:
 - Cálculo, programação, projetos, etc.

Soft Skills

- São competências subjetivas, muito mais difíceis de avaliar.
- Em geral habilidades com pessoas (interpessoais).
 - Empatia, resiliência, comunicação, criatividade, liderança, motivação, trabalho em equipe, solução de problemas, etc.

Vídeo: Entrevista com estagiário



https://www.youtube.com/watch?v=_IRD3Ejm0eM

Vídeo: Entrevista com estagiário

Alguns estudos apontam que boa parte das demissões são causadas por fatores comportamentais e emocionais.



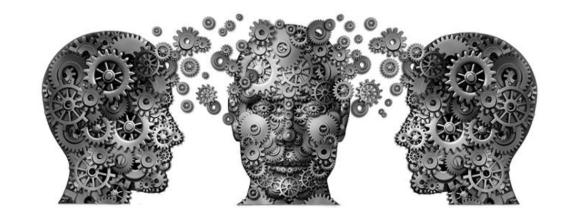
1. Unidade





Paradigmas de Programação

Um paradigma de programação fornece (e determina) a visão que o programador possui sobre a estruturação e execução do programa;



Paradigmas de Programação

Assim como ao resolver um problema podemos adotar uma entre variadas metodologias, ao criar um programa podemos adotar um determinado paradigma de programação para desenvolvê-lo.

Paradigmas de Programação

Vamos entender a diferença entre **Programação Estruturada** e **Programação Orientada a Objetos**.



Sequência: Uma tarefa é executada após a outra, linearmente.

Decisão: A partir de um teste lógico, determinado trecho de código é executado, ou não.

Iteração: A partir de um teste lógico, determinado trecho de código é repetido por um número finito de vezes.



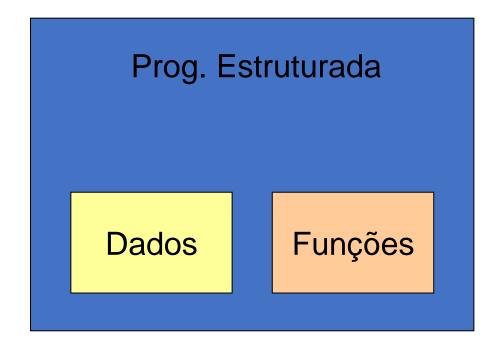
Vantagens

- É fácil de entender.
- Ainda muito usada em cursos introdutórios de programação.
- Execução mais rápida.

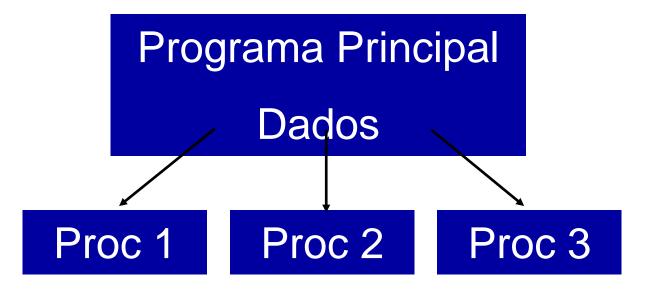
Desvantagens

- Baixa reutilização de código
- Códigos confusos: Dados misturados com comportamento.

Ênfase nos procedimentos, implementados em blocos estruturados, com comunicação entre procedimentos por passagem de dados;



Na programação estruturada o código é composto por vários processos ligados através de chamadas.



Exemplo:

Linguagens estruturadas(C, Pascal, Cobol, Basic, Clipper).

- Diminuiu a distância entre mundo real e solução computacional;
- Baseada em Classes e Objetos;
- Métodos e Atributos;
- Programação se dá pela comunicação entre objetos.

Vantagens

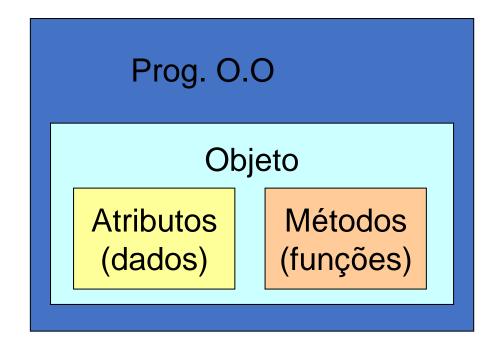
- Melhor organização do código ;
- Bom reaproveitamento de código.

Desvantagens

Mais difícil compreensão.

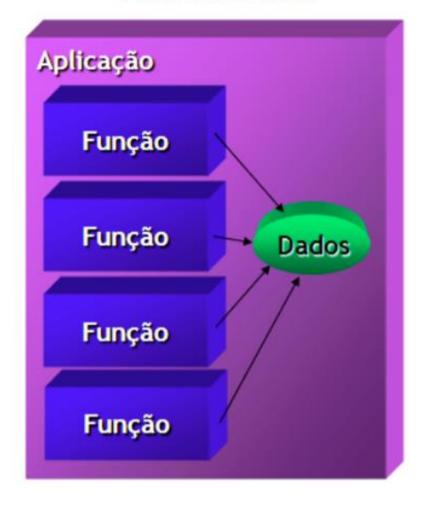
Dados e procedimentos fazem parte de um só elemento básico (objeto).

Os elementos básicos comunicam-se entre si por mensagens e tem ênfase nos dados e no agrupamentos dos mesmos.

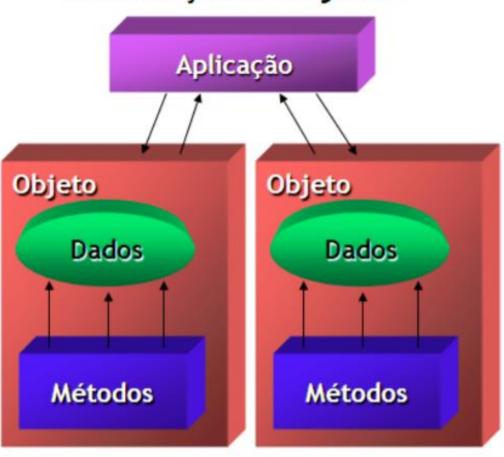


Programação Orientada a Objetos x Estruturada

Estruturada



Orientação a Objetos



Abstração:

Habilidade de se concentrar nos aspectos essenciais do sistema, ou um contexto qualquer, ignorando o que é supérfluo;

Objeto:

- Representação computacional de algo do mundo real
 - Concreto
 - Abstrato

Abstração

Transformar aquilo que observamos realidade para a virtualidade.

Objetos concretos

- Cão
- Moto
- Casa

Objetos abstratos

- Música
- Transação Bancária

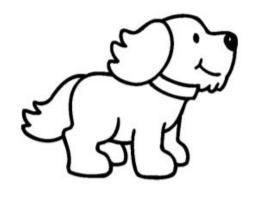
Modelo

Características + Comportamento

Estado - Atributos (Características)

Operações - Métodos (Comportamentos)

Atributos e Métodos



- Atributos
 - Raça: Poodle
 - Nome: Rex
 - Peso: 5 quilos

- Método
 - Latir
 - Comer
 - Dormir



- Potência: 500cc
- Modelo: Honda
- Ano: 1998

- Acelerar
- Frear
- Abastecer

Atividade

Cite 4 atributos de um aluno:

Cite 3 métodos de um aluno:



Por que usar Programação Orientada a Objetos

A metodologia Orientação a Objetos é baseada em "objetos do mundo real", e por este motivo é mais intuitiva, pois oferece recursos como: objetos e atributos, classes e membros, estruturas e componentes, ação e reação.

Em um sistema orientado a objetos, os dados e todas as operações, que manipulam esses dados, são agrupados em estruturas chamadas de "classes".



Os principais problemas do software hoje são:

- Diminuir o custo e o tempo da mudança;
- Aumentar a capacidade e facilidade de adaptação.

Objetos são especialmente bons para:

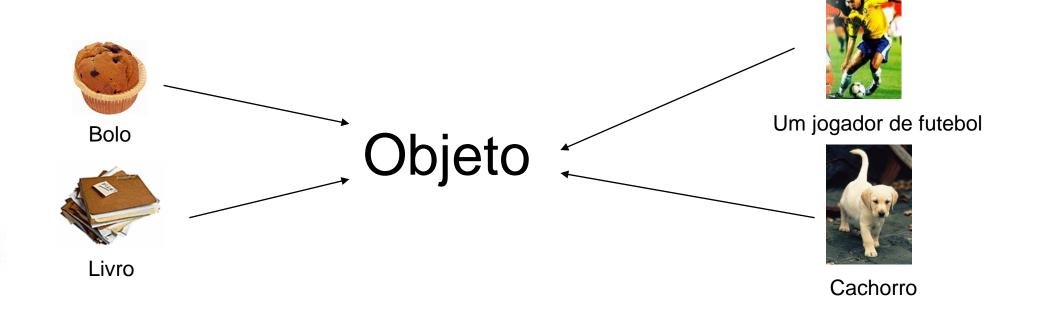
- Reduzir o tempo necessário para adaptar um sistema existente (reação mais rápida à mudanças no seu ambiente de negócio);
- Reduzir o esforço, a complexidade e os custos associados à mudança

O termo orientação a objetos significa organizar o mundo real como uma coleção de objetos que incorporam estrutura de dados e um conjunto de operações que manipulam estes dados.



Classe: Pessoa	Propriedades	Comportamentos
	Nome Profissão Data de Nascimento Peso Altura	Andar Correr Trabalhar Chorar Dançar
Classe: Pássaro	Propriedades	Comportamentos
	Espécie Cor das penas Tamanho Peso	Andar Correr Voar Pousar

Lembrando que objetos são quaisquer coisas na natureza que possuam propriedades (características) e comportamentos (operações).



Um objeto tem dados e comportamento.



Atributos





nome = Marina cpf = 022.200.708-12 idade = 27

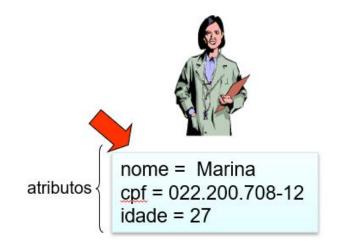
corre dorme nome = Felipe cpf = 039.217.908-22 idade = 42

corre dorme

Atributos

São características presentes nos objetos;

- Os valores de todos os atributos é chamado de estado do objeto;
- Somente atributos que são de interesse do sistema devem ser considerados.

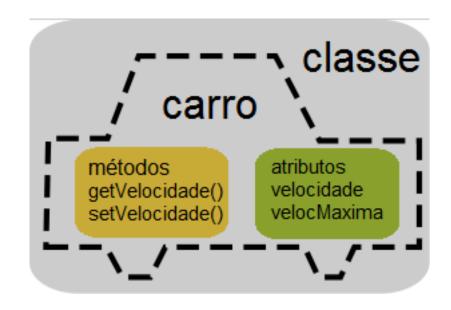


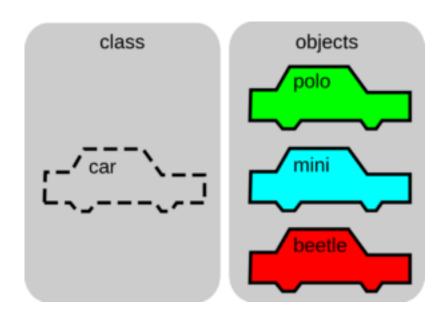
Atributos

São características presentes nos objetos;

Para criarmos objetos em um sistema precisamos abstraí-lo criando uma classe que o represente.

As **Classes** e os **Objetos** possuem uma relação de dependência, pois não existe objeto sem a classe;





A Classe representa a abstração das características comuns mais relevantes (atributos e métodos) de um conjunto de objetos. A classe passa a ser um molde para se criar objetos do mesmo tipo.

"Uma classe é uma entidade que descreve um conjunto de objetos com propriedades (atributos) e comportamentos (métodos) semelhantes e com relacionamentos comuns com outros objetos"

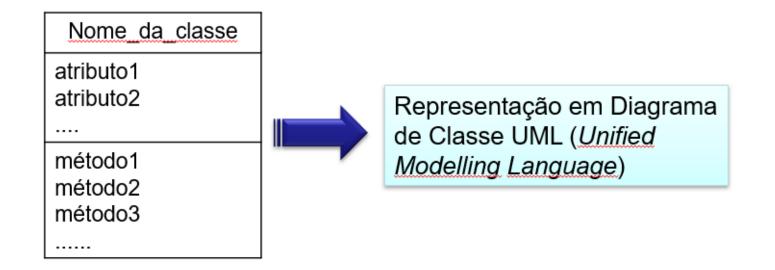


As classes são as partes mais importantes de qualquer sistema orientado a objetos.

Essas classes podem incluir abstrações que são parte do domínio do problema.



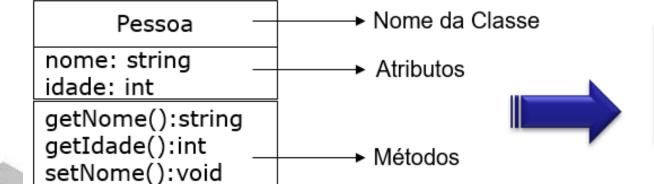
Graficamente, as classes são representadas por retângulos incluindo nome, atributos e métodos. Devem receber nomes representativos.



setIdade():void

Exemplo:

A classe Pessoa deverá ter atributos e métodos comuns



Representação em Diagrama de Classe UML (*Unified Modelling Language*)

Construindo classes em Java:

Nome_da_classe

atributo1
atributo2
....

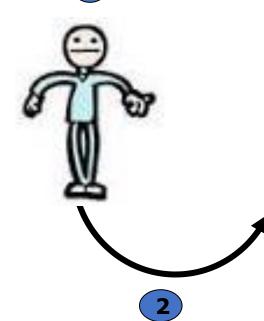
método1
método2
método3
.....

Modelo UML de classe

```
public class <Nome_da_classe> {
      de atributos>
      de métodos>
}
```

Um arquivo em Java precisa ser uma classe pública com o mesmo nome do arquivo

1 Pessoa



Pessoa

nome: string

idade: int

renda: double

public class Pessoa {
 //lista de atributos
 String nome;
 int idade;
 double renda;

//lista de métodos

Diagrama UML

Cliente

nome: string

endereço: string

telefone: string

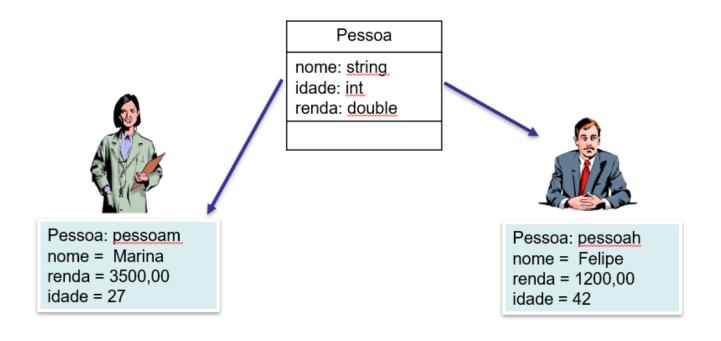
e-mail: string

Classe em Java

```
public class Cliente {
   //atributos
   String nome;
   String endereco;
   String telefone;
   String email;
   //lista de métodos
```

Instanciando um Objeto

A classe é um modelo para os objetos do sistema, desta forma após a definição da classe podemos instanciar (criar) objetos que serão do tipo da classe. Eles são chamados de instâncias.



Instanciando um Objeto

```
public class Pessoa {
    //lista de atributos
    String nome;
    int idade;
    double renda;
```

Atributos (em letras minúsculas)

Classe (iniciais em maiúsculas)

```
public class UsaClasses {
   public static void main(String[] args) {
       //Pessoa p = new Pessoa();
        Pessoa p;
       p = new Pessoa();
       p.nome = "Fulano";
                                     Objetos (em letras
       p.idade = 25;
                                     minúsculas)
       p.renda = 1000;
       System.out.println("Nome: " + p.nome);
       System.out.println("Idade: " + p.idade);
       System.out.println("Renda: " + p.renda);
       System.out.println("Nome: " + p1.nome);
       System.out.println("Idade: " + p1.idade);
       System.out.println("Renda: " + p1.renda);
```

Instanciando um Objeto

```
public class UsaClasses {
                                                                 Método principal por onde o Java
    public static void main(String[] args) {
                                                                 começa a execução do programa
        //Pessoa p = new Pessoa();
        Pessoa p;
        p = new Pessoa();
        Pessoa p1 = new Pessoa();
                                                                Instanciando 2 objetos da classe
                                                                Pessoa. Cada um com o seu
        p.nome = "Fulano";
                                                                identificador
        p.idade = 25;
        p.renda = 1000;
                                                               Estamos criando o estado do objeto
        System.out.println("Nome: " + p.nome);
                                                               p1, acessando os seus atributos
        System.out.println("Idade: " + p.idade);
                                                               através de um ponto após o nome
        System.out.println("Renda: " + p.renda);
                                                               do objeto
        System.out.println("Nome: " + p1.nome);
        System.out.println("Idade: " + p1.idade);
        System.out.println("Renda: " + p1.renda);
                                                               Podemos também acessar os
                                                               atributos e recuperar o estado dos
                                                               objetos para imprimir na tela o seu
                                                               conteúdo
```

OBS: Os arquivos de classe UsaClasses.java e Pessoa.java precisam estar na mesma pasta

Outros exemplos:

Triangulo

base: float

altura: float

Data

dia: int

mes: int

ano: int

Curso

nome: string

qtdealunos: int

turma: string

Exercícios

- 01-) Crie um diagrama de classes UML para abstrair os atributos dos seguintes objetos:
 - a) Eletrodoméstico
 - b) Carro
 - c) Caixa de Diálogo

Exercícios

- 01-) Crie um diagrama de classes UML para abstrair os atributos dos seguintes objetos:
 - a) Eletrodoméstico
 - b) Carro
 - c) Caixa de Diálogo
- 02-) Implemente as classes do exercício acima (cada uma em um arquivo).
- 01-) Crie uma classe com o método main e instancie um objeto de cada uma das classes acima, coloque valores nos atributos e os mostre na tela.

Obrigado!

Se precisar ...

Prof. ^a Erika Miranda

emiranda@cruzeirodosul.edu.br