

Definição e Sintaxe de Classes

Curso de Engenharia de Controle e Automação
DPEE1090 - Programação orientada a objetos para automação

Prof. Renan Duarte

1º semestre de 2024

Sumário

Definição e Sintaxe de Classes

- Conceitos fundamentais de classes
- Estrutura e sintaxe de declaração de classes
- Atributos, métodos e construtores

Exercício 1 – Resolvendo um problema sem orientação à objetos

Fazer um programa para ler as medidas dos lados de dois triângulos X e Y (suponha medidas válidas). Em seguida, mostrar o valor das áreas dos dois triângulos e dizer qual dos dois triângulos possui a maior área.

Fórmula de Herão:

$$Area = \sqrt{p(p - a)(p - b)(p - c)}$$

$$p = \frac{a + b + c}{2}$$



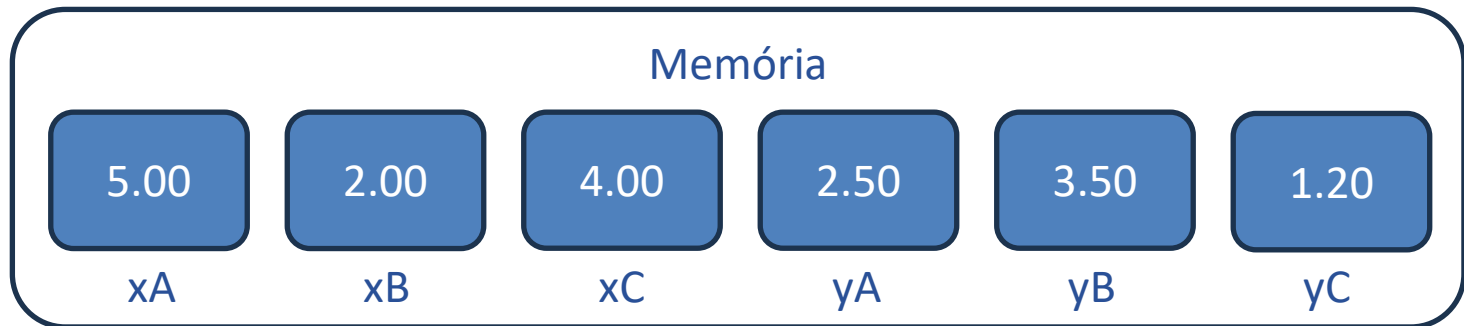
Herão de Alexandria
10 d.C. - 80 d.C.

Introdução

Quais os problemas da solução desenvolvida?

- Triângulo é uma entidade com três atributos: a, b, c
- Estamos usando três variáveis distintas para representar cada triângulo:

Λάσiάwêiς ράσά ôς λάδôς δôς ηςiάngulôς
δôυčlê yA yB yC òA òB òC



Classe

Definição

É um tipo estruturado que pode conter (membros):

- Atributos (dados/campos)
- Métodos (funções/operações)

A classe também pode prover muitos outros recursos, tais como:

- Construtores
- Sobrecarga
- Encapsulamento
- Herança
- Polimorfismo

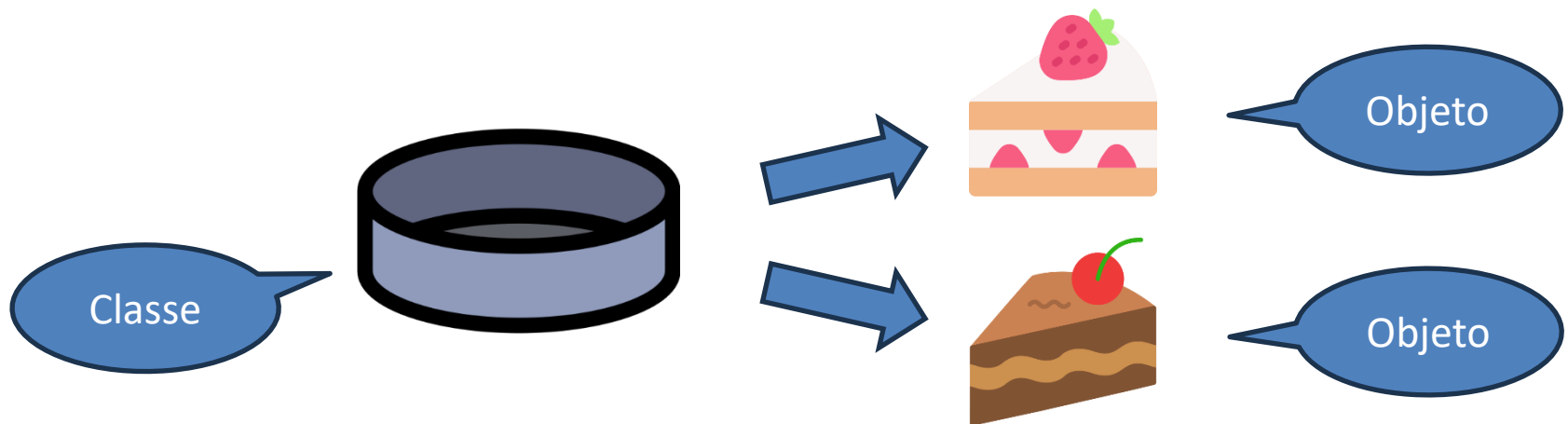
É o “elemento base” da orientação à objetos

Classe

Definição

Classes podem representar inúmeras “coisas”:

- Entidades: Produto, Cliente, Triângulo
- Serviços: ClienteService, EmailService, StorageService
- Controladores: ProdutoController, ClienteController
- Utilitários: Calculadora, Compactador
- Outros (views, repositórios, gerenciadores, etc.)



Exercício 2 – Melhorando a solução com atributos de uma classe

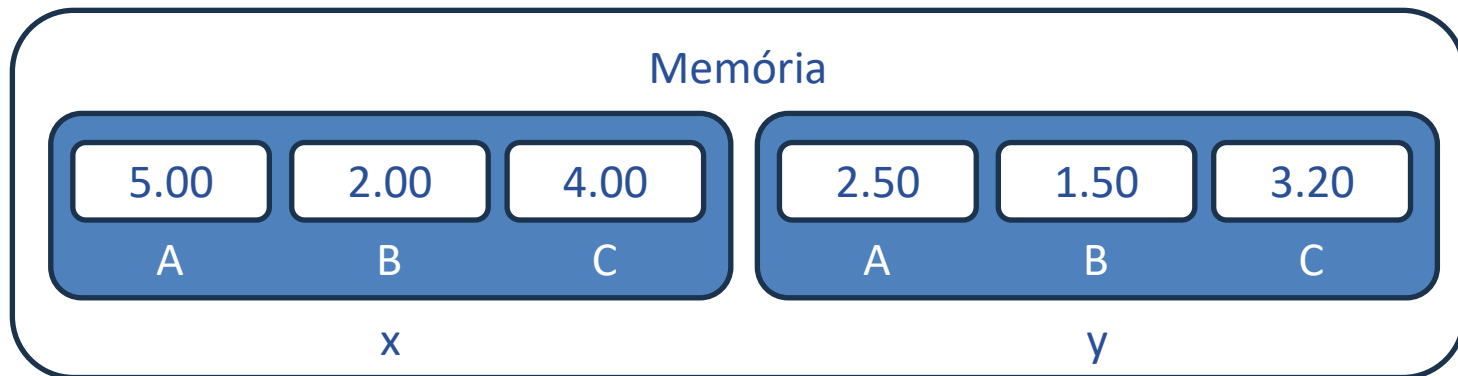
Objeto Classe

Instância classe

Objeto A

Objeto B

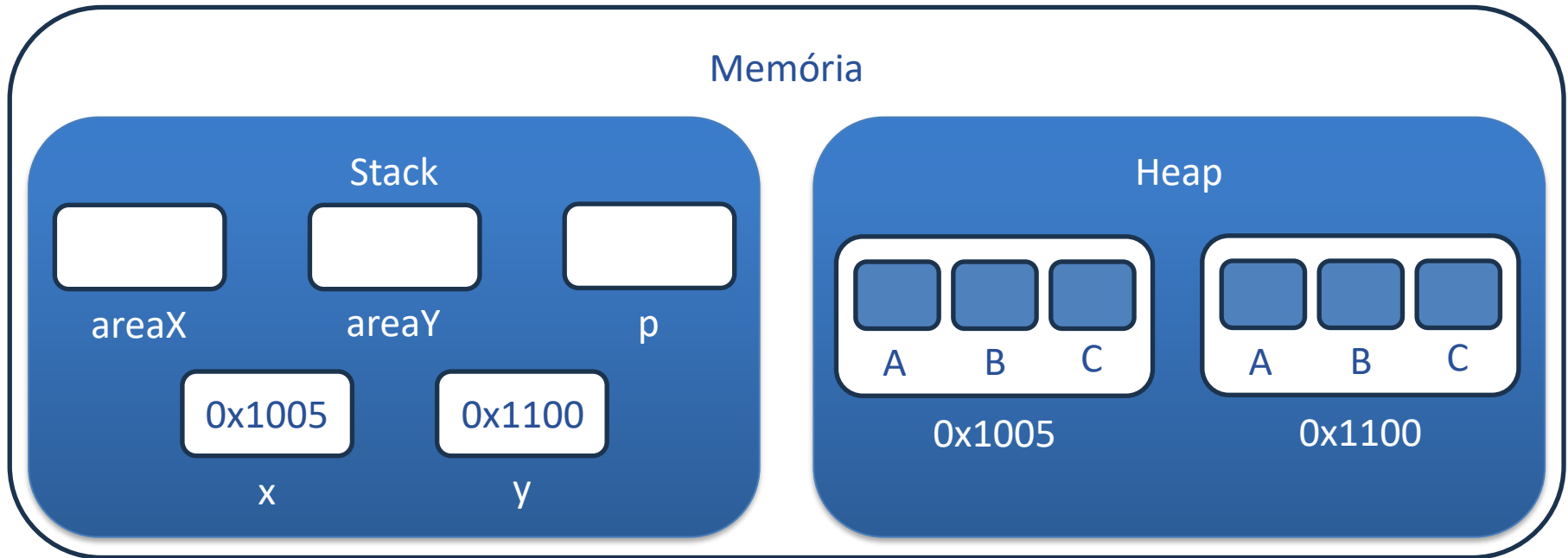
Objeto C



Classe

Instanciação - Alocação dinâmica de memória

`double *x, *y;` } Declaração → Alocado espaço na stack
`x = (double *) malloc(100);` } Instanciação → Alocado espaço na heap
`y = (double *) malloc(100);`



Classe

Vantagens da nova solução

Agora, atributos ligados a entidade triângulo ficam agrupados dentro de uma única variável → Objeto composto que contém as variáveis A, B e C.

Dessa forma, nosso código fica mais coeso, facilitando a leitura e interpretação, especialmente à medida que os problemas se tornam mais complexos.

Além disso, aumentamos a modularidade do código, visto que o uso de uma classe permite reaproveitar unidades de código sem a necessidade de grandes modificações na solução como um todo.



Exercício 3

Fazer um programa para ler os dados de duas pessoas, depois mostrar o nome da pessoa mais velha.

Exemplo:

```
Dãđđộş đấ ấsĩnêĩsấ ấềşşộấ  
Nộặ Nấsĩấ  
Íđấđê ,
```

```
Dãđđộş đấ sêgụđấ ấềşşộấ  
Nộặ Kộấộ  
Íđấđê ,
```

```
Ầềşşộấ ấấĩş ấêlặấ Nấsĩấ
```



Exercício 4 – Melhorando o exercício 2 com métodos de uma classe

Na solução anterior para o problema dos triângulos, melhoramos o código com o uso de atributos de uma classe “Triângulo”. Contudo, ainda temos repetição de código no programa:

```
Cálculo da área de Y
doyle r y A y B y C ,
doyle áreaY Ntjh Srstj r r y A r y B r y C

Cálculo da área de Y
r y A y B y C ,
doyle áreaY Ntjh Srstj r r y A r y B r y C
```

Vamos utilizar métodos de uma classe para obter os benefícios de reaproveitamento e delegação

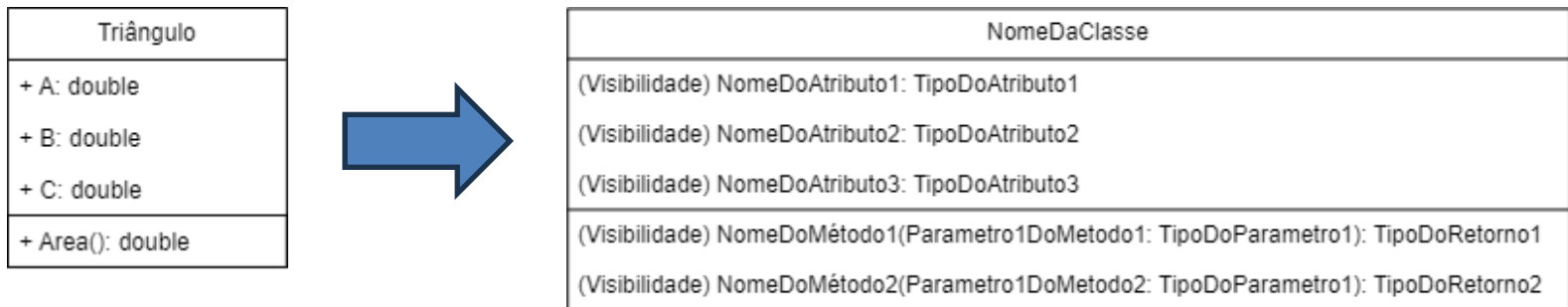
Classe

Projeto de uma classe

UML: Linguagem de modelagem unificada (do inglês *Unified Modeling Language*)

- É uma linguagem-padrão para a elaboração da estrutura de projetos de software

Representação da nossa classe triângulo:



<https://www.youtube.com/watch?v=rDidOn6KN9k>
<https://www.youtube.com/watch?v=WnMQ8HlmeXc>

Classe

Construtor

É uma operação especial da classe, que executa no momento da instanciação do objeto → `Isiăngûlô y nêx Isiăngûlô`

Usos comuns:

- Iniciar valores dos atributos
- Permitir ou obrigar que o objeto receba dados / dependências no momento de sua instanciação (injeção de dependência)

Se um construtor customizado não for especificado, a classe disponibiliza o construtor padrão:

- `Produto p = new Produto();`

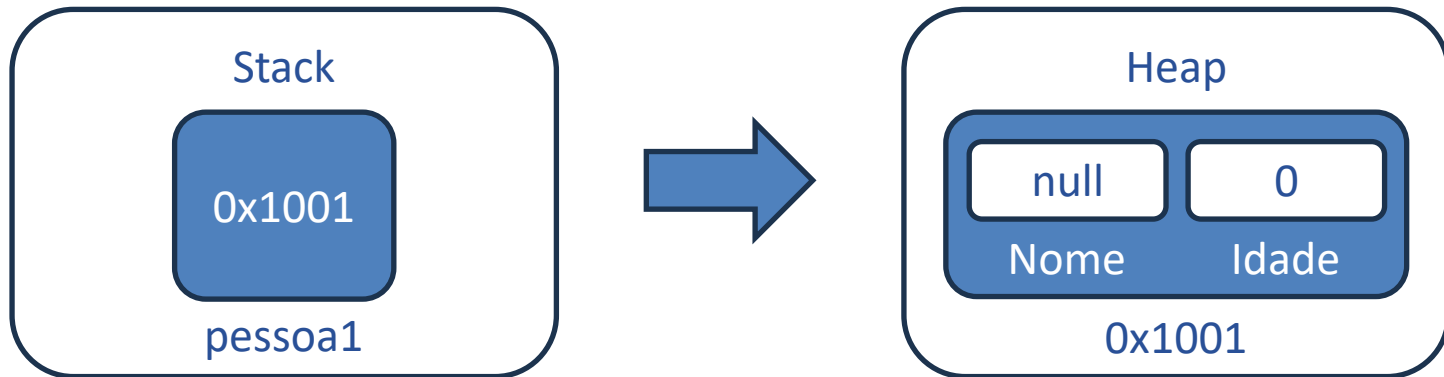
É possível especificar mais de um construtor na mesma classe

Classe

Construtor

Usando o Exercício 3 como exemplo:

Rêşşôă řêşşôă, **ñêx** Rêşşôă



Faz sentido uma pessoa que não tenha nome ou “idade = 0” no programa?

Como evitar essa situação?



Exercício 5 – Criando um construtor para o exercício 3

Cópiamos o código da classe
e adicionamos o construtor

```
Nome nome;  
Idade idade;
```

```
Instanciando o objeto: Nome rs, idade rs  
rs, rs, rs, rs → Inválido: Parâmetros nulos
```

```
rs, rs, rs, rs, rs → Ok
```

Programador é obrigado a passar os parâmetros nome e idade na instanciação de rs,



Exercício 6 – Construtor com sobrecarga

Podemos declarar mais de um construtor para a mesma classe com diferentes parâmetros em cada um → **Sobrecarga**

Dessa forma podemos fazer com que um ou mais atributos sejam opcionais na instanciação da classe

Exemplo: Modificar o Exercício 5 de forma que um novo atributo `altura` seja opcional na instanciação da classe `Reptil`

```
class Reptil:
    def __init__(self, nome, idade, peso, altura):
```

```
        self.nome = nome
        self.idade = idade
        self.peso = peso
        self.altura = altura
```


Classe

A palavra *this*

No exemplo abaixo, como diferenciar os atributos da classe dos parâmetros do construtor?

```
Atsıcütöş đã rêşşöă  
řučlíc şťsingŋ ñệñê  
řučlíc інť idăđê
```

```
Cộşťşutöş  
řučlíc Rêşşöă şťsingŋ ñệñê інť idăđê
```

```
ñệñê ñệñê  
idăđê idăđê
```

Classe

A palavra *this*

Nestes casos, deve-se usar a palavra **this** para se referir aos atributos internos da classe:

Atsĩčũťôş đấ rêşşốấ
řũčlĩç şťşĩngê ặặê
řũčlĩç ỉnť ỉđấđê

Cộşťşũťôş
řũčlĩç Rêşşốấ şťşĩngê ặặê ỉnť ỉđấđê

ťhĩş ặặê ặặê
ťhĩş ỉđấđê ỉđấđê

Cores iguais
representam
variáveis iguais



Exercício 7

Fazer um programa para ler os dados de um funcionário (nome, salário bruto e imposto). Em seguida, mostrar os dados do funcionário (nome e salário líquido). Em seguida, aumentar o salário do funcionário com base em uma porcentagem dada (somente o salário bruto é afetado pela porcentagem) e mostrar novamente os dados do funcionário.

Use a classe projetada abaixo:

Funcionario
+ Nome: string
+ SalarioBruto: double
+ Imposto: double
+ SalarioLiquido(): double
+ AumentarSalario(porcentagem: double): void



Exercício 7 - Continuação

Exemplo:

Nome: **Renan Duarte**

Salário bruto: **6000.00**

Imposto: **1000.00**

Funcionário: Renan Duarte, \$5000.00

Digite a porcentagem para aumentar o salário: **10.0**

Dados atualizados: Renan Duarte, \$5600.00

Revisão

Próxima aula

Instâncias de classes - Objetos

- Compreensão dos conceitos de objetos e instância de classes
- Métodos padrão de classes
- Membros estáticos
- Exemplos práticos

Dúvidas?

renan.duarte@gedre.ufsm.br

GEDRE – Prédio 10 – CTLAB