



Ciência da **Computação**

Programação Orientada a Objetos
Prof. Luciano Rodrigo Ferretto



2025/01 - Org. e Abs. na Programação

Grupo do WhatsApp



Nossa Aula de hoje

- Estrutura do AVA
- Um pouco sobre nós
- Como será nosso semestre
- Paradigmas de Programação
- Paradigmas Imperativos
- Entregáveis
- Questões de Estudo

Introdução e Plano de Ensino - AVA

- Nosso Plano de Ensino está disponível no nosso AVA, na seção Introdução
- Nesta seção também temos os contatos dos professores da disciplina.

<https://imed.mrooms.net/course/section.php?id=187972#section-0>

Feedback

- Seu feedback é essencial para nós continuarmos aprimorando nossas aulas e proporcionando a melhor experiência de aprendizado possível. Sua opinião sobre o conteúdo, metodologia e dinâmica das aulas é inestimável para nós.
- Cada comentário que vocês compartilham é uma oportunidade para melhorarmos juntos. Sejam honestos e específicos em suas observações, pois é através delas que podemos identificar pontos fortes e áreas de melhoria.
- Lembrem-se de que sua voz tem o poder de moldar o rumo do nosso curso. Ao expressarem suas opiniões, estão contribuindo para o desenvolvimento de uma comunidade de aprendizado mais eficaz e inclusiva.
- Juntos, podemos criar um ambiente onde o diálogo aberto e construtivo nos guie para alcançar excelência acadêmica.

Feedback

→ 👁 ⋮

Feedback Constante das aulas de CC 2025.1 - Organização e Abstração na Programação

Não é necessário estar logado com uma conta google para responder esse formulário.
Assim, sua identidade é mantida em sigilo.

Em uma escala de 1 a 5, avalie os itens abaixo referente as aulas:

[Faça login no Google](#) para salvar o que você já preencheu. [Saiba mais](#)

* Indica uma pergunta obrigatória

E-mail *

Selecione o Professor: *

Escolher ▼

Próxima

Limpar formulário

Nunca envie senhas pelo Formulários Google.

Before - AVA

<https://imed.mrooms.net/course/section.php?id=187972#section-1>

A seção **Before** tem por finalidade oferecer aos alunos conteúdos essenciais para um bom aprendizado deste módulo. Estes materiais não são de consumo obrigatório, porém são premissas indicadas para um bom aprendizado. **Todas as trilhas indicadas são gratuitas e estão no YouTube.**

- **Obs.: recomendado abrir no YouTube para ver as demais aulas da trilha.**

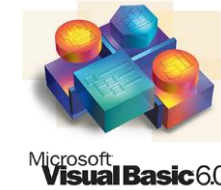
Apresentação - AVA

- Em nosso AVA temos a seção LRF - Apresentação, no qual consta os contatos que vocês podem estar utilizando para falar com o professor, assim como o link para no nosso grupo de WhatsApp, e também o Link para o repositório GIT da turma no GitHub

<https://imed.mrooms.net/course/section.php?id=187972#section-4>

Luciano Rodrigo Ferretto

- Academia
 - Bacharel em Sistemas de Informação
 - Mestre em Computação Aplicada
 - Sistemas de Recomendação
- Atuação Profissional (na área de desenvolvimento)
 - Início em meados de 2001 com Microsoft Visual Basic 5.0, passando depois para o VB 6.0
 - Algumas “aventuras” no C# com a chegada do .NET
 - Atualmente programando JavaScript e principalmente com TypeScript para aplicações node js (REST Web Services)
 - Também atuando em ReactNative
 - E por fim, “sempre” (kkkk) desenvolvendo em Java
 - JSP
 - REST





E você ????

- Fale um pouco sobre você para nos conhecermos:
 - Sua experiência com programação?
 - Quais as linguagens?
 - Estás trabalhando além de estudar?
 - Trabalhas na área da computação?
 - Trabalhas com programação?
- Sua expectativa com a disciplina?

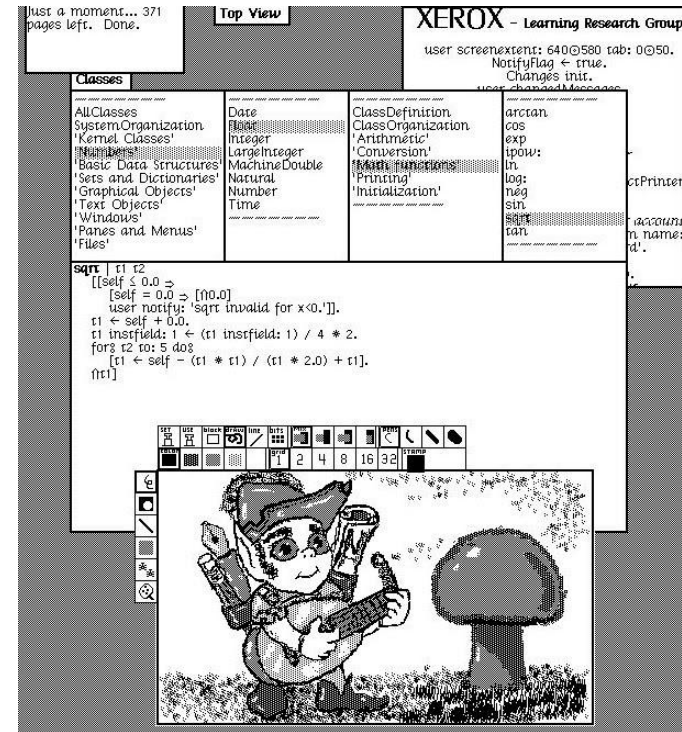


Agora sim! Vamos ao que interessa

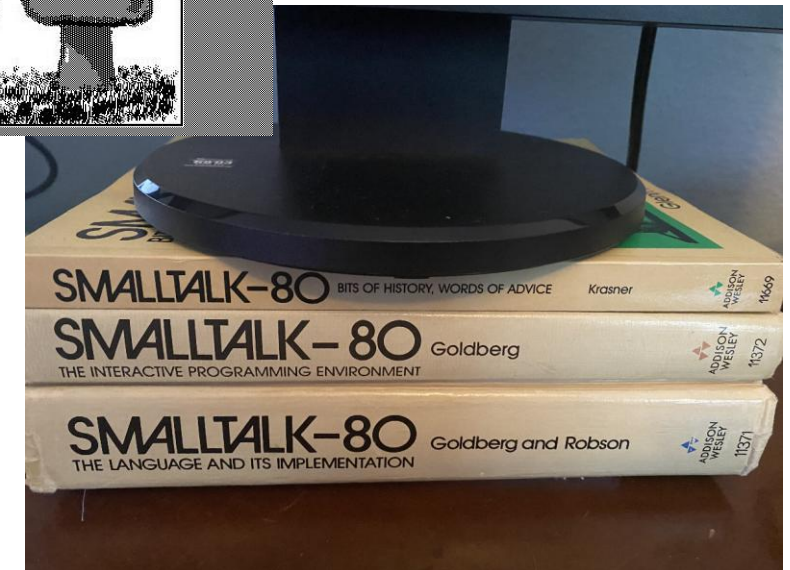


POO - Origem

- Linguagem: Smalltalk-80
 - Introduziu o conceito de IDE
- Criador: Alan Kay



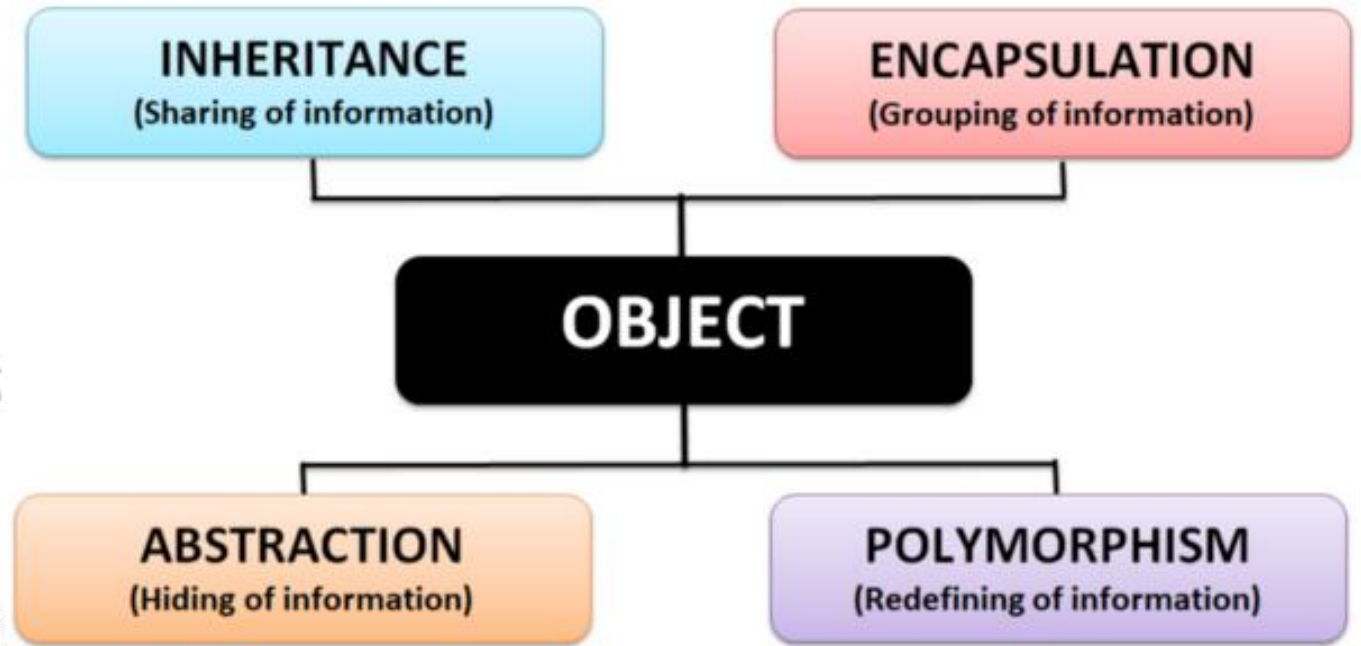
1950 – 1960 Era do Caos	1970 – 1980 Era da Estruturação	1990 até agora Era dos Objetos
Saltos, gotos, variáveis não estruturadas, variáveis espalhadas ao longo do programa	If-then-else Blocos Registros Laços-While	Objetos Mensagens Métodos Herança





https://www.youtube.com/watch?v=KIIL63MeyMY&list=PLHz_AreHm4dkqe2aR0tQK74m8SFe-aGsY&index=1

The Four Pillars

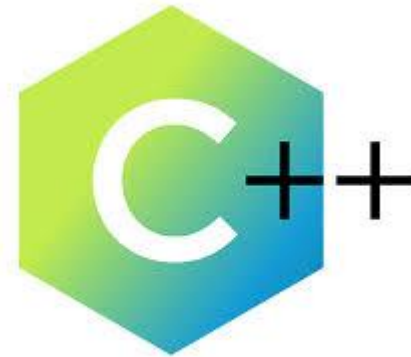




https://www.youtube.com/watch?v=pbb0jzXt_xA

Computação

Linguagens Orientadas a Objeto





Mas calma...

Vamos lembrar um pouco

Calma... Antes vamos lembrar um pouco do que já foi visto!

1.Algoritmo:

Um algoritmo é uma sequência de instruções finita, ordenada e precisa que descreve um método para resolver um problema ou realizar uma tarefa.

Plano ou roteiro para alcançar um objetivo específico.

Algoritmos existem independentemente de qualquer linguagem de programação específica e podem ser expressos de várias maneiras, como texto, fluxogramas ou pseudocódigo.

2.Programação:

Programação é o processo de escrever código em uma linguagem de programação específica para instruir um computador a realizar uma tarefa ou resolver um problema.

Envolve a tradução de algoritmos e lógica de programação em código executável.

- Em resumo, a programação é o processo de transformar um algoritmo em código de computador executável
- A programação é a aplicação prática dos conceitos de algoritmos na criação de software.



VSCode



Visual Studio Code

- Visual Studio Code (VS Code) é um dos ambientes de desenvolvimento integrado (IDE) mais populares e amplamente utilizados disponíveis atualmente. Desenvolvido pela Microsoft, o VS Code é uma ferramenta gratuita e de código aberto que oferece uma ampla gama de recursos e extensões para facilitar o desenvolvimento de software em diversas linguagens de programação.

Python



- Python é uma linguagem de programação de alto nível, interpretada, de propósito geral e fácil de aprender.
- Uma das principais características de Python é sua ênfase na legibilidade do código
- Seu uso extensivo de indentação (espaços em branco) para delimitar blocos de código, em vez de usar chaves ou palavras-chave como em outras linguagens, torna o código Python intuitivo e fácil de entender.
- Python é uma linguagem interpretada, o que significa que o código-fonte é executado diretamente por um interpretador, sem a necessidade de compilação prévia.

Declaração de Variáveis em Python:

- Em Python, as variáveis são criadas atribuindo um valor a um nome. Não é necessário declarar explicitamente o tipo de uma variável; o tipo é inferido automaticamente com base no valor atribuído. Para declarar uma variável em Python, basta atribuir um valor a um nome usando o operador de atribuição '='. Por exemplo:

```
1  # Declaração de variáveis
2  idade = 25
3  nome = "João"
4  salario = 1500.50
```



Declaração de Funções (Procedimentos) em Python:

- Em Python, uma função é um bloco de código reutilizável que executa uma tarefa específica. Para definir uma função em Python, você usa a palavra-chave 'def', seguida pelo nome da função, parênteses contendo os parâmetros da função (se houver) e dois pontos ':'. O corpo da função, que contém as instruções a serem executadas, é indentado. Por exemplo:

```
1  # Declaração de função
2  def saudacao(nome):
3      ... print(f"Olá, {nome}! Bem-vindo!")
4
5  saudacao("Raul Seixas")
6  "
```



Estrutura Condicional: if, elif e else

- O "if" é usado para executar um bloco de código se uma condição for verdadeira. O "elif" é usado para verificar múltiplas condições após o "if". O "else" é usado para executar um bloco de código se nenhuma das condições associadas ao "if" ou "elif" for verdadeira.

```
1  nota = 85
2  if nota ≥ 90:
3      print("Você tirou A.")
4  elif nota ≥ 80:
5      print("Você tirou B.")
6  elif nota ≥ 70:
7      print("Você tirou C.")
8  else:
9      print("Você tirou F.")
```



Estrutura de Repetição: while

- O loop "while" é usado para repetir um bloco de código enquanto uma condição específica for verdadeira. A condição é verificada antes de cada iteração do loop. Se a condição for verdadeira, o bloco de código dentro do loop é executado. Quando a condição se torna falsa, a execução do loop é interrompida.

```
1  contador = 0
2  while contador < 5:
3      | ... print(contador)
4      | ... contador += 1
```



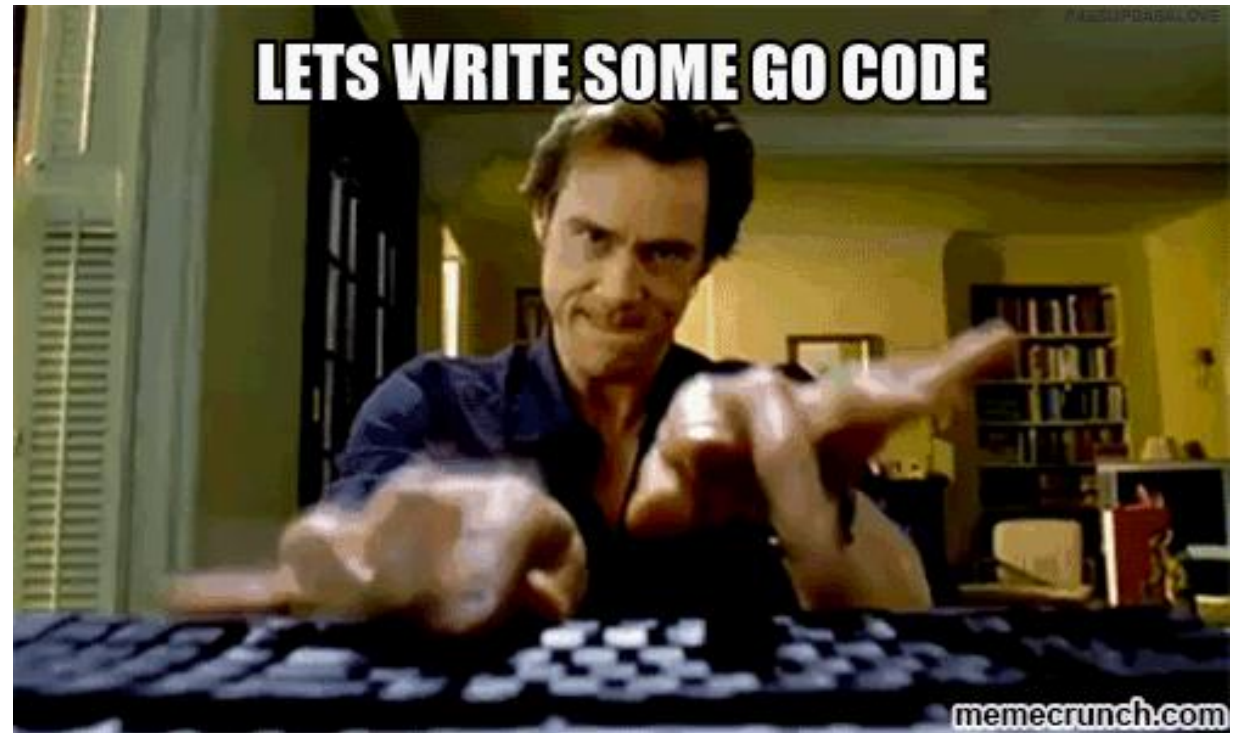
Estrutura de Repetição: for

- O loop "for" é usado para iterar sobre uma sequência (como uma lista, tupla, dicionário, etc.) ou um iterável. Ele executa um bloco de código para cada item na sequência ou iterável.

```
1  lista = ["maçã", "banana", "laranja"]
2  for fruta in lista:
3      print(fruta)
```




Let's code



Desafio: Jogo de Adivinhação em Python

Você foi contratado para desenvolver um jogo de adivinhação simples em Python! O programa deve gerar um número aleatório entre 1 e 100, e o jogador precisa adivinhar qual é esse número.

-  Regras do jogo:
 - O programa escolhe um número aleatório entre 1 e 100.
 - O jogador digita um palpite.
 - Se o palpite for menor que o número secreto, o programa exibe: "Tente um número maior."
 - Se for maior, exibe: "Tente um número menor."
 - O jogo continua até que o jogador acerte.
 - Ao acertar, o programa deve mostrar uma mensagem de parabéns e o número de tentativas utilizadas.

