

Algoritmos e Programação III

Aula 4 – Estruturas de Dados

Parte 1

Profa. Dr^a. Eduarda Rodrigues Monteiro

Porto Alegre

O que vimos até agora ...

- **Cenário de dados atual: Big Data**
 - Importância/Desafio de manipular dados com ênfase no cenário atual
- **Processo de Compilação/Interpretação:**
 - Execução de uma programa escrito em uma linguagem de alto nível
- **Alocação de Memória – Endereçamento dos dados**
 - Estática vs Dinâmica
- **Linguagem de Programação Python**
 - Conceitos Básicos
 - Métodos
 - Classes e Objetos

O que vimos até agora ...

- **Cenário de dados atual: Big Data**
 - Importância/Desafio de manipular dados com ênfase no cenário atual
- **Processo de Compilação/Interpretação:**
 - Execução de uma programa escrito em uma linguagem de alto nível
- **Alocação de Memória – Endereçamento dos dados**
 - Estática vs Dinâmica
- **Linguagem de Programação Python**
 - Conceitos Básicos
 - Métodos
 - Classes e Objetos

O que vimos até agora ...

- **Cenário de dados atual: Big Data**
 - Importância/Desafio de manipular dados com ênfase no cenário atual



Dados no
mundo....

Representação do
dados no
mundo....



O que vimos até agora ...

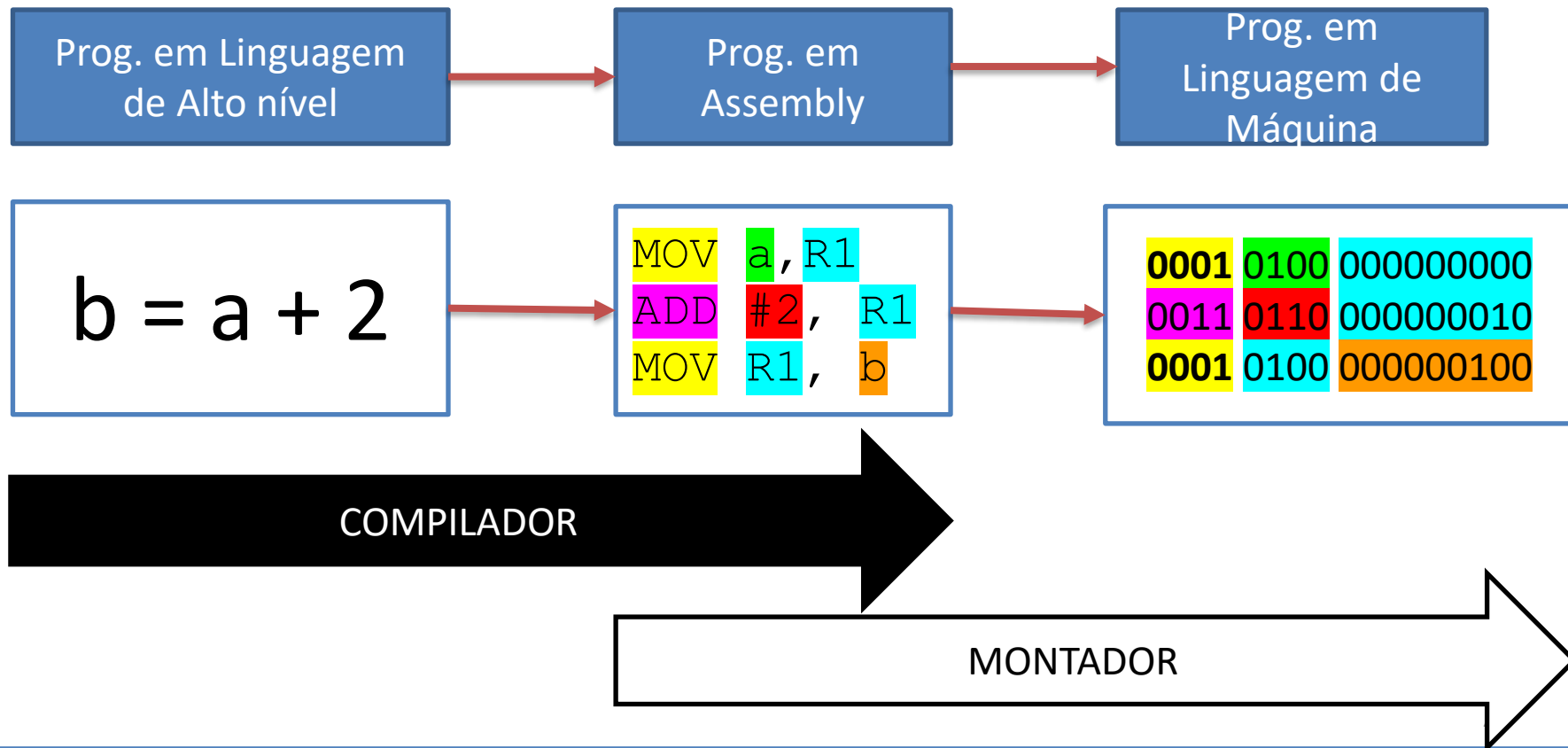
- **Cenário de dados atual: Big Data**
 - Importância/Desafio de manipular dados com ênfase no cenário atual
- **Processo de Compilação/Interpretação:**
 - Execução de uma programa escrito em uma linguagem de alto nível
- **Alocação de Memória – Endereçamento dos dados**
 - Estática vs Dinâmica
- **Linguagem de Programação Python**
 - Conceitos Básicos
 - Métodos
 - Classes e Objetos

O que vimos até agora ...

- **Cenário de dados atual: Big Data**
 - Importância/Desafio de manipular dados com ênfase no cenário atual
- **Processo de Compilação/Interpretação:**
 - Execução de uma programa escrito em uma linguagem de alto nível
- **Alocação de Memória – Endereçamento dos dados**
 - Estática vs Dinâmica
- **Linguagem de Programação Python**
 - Conceitos Básicos
 - Métodos
 - Classes e Objetos

O que vimos até agora ...

- **Processo de Compilação/Interpretação:**
 - Execução de uma programa escrito em uma linguagem de alto nível



O que vimos até agora ...

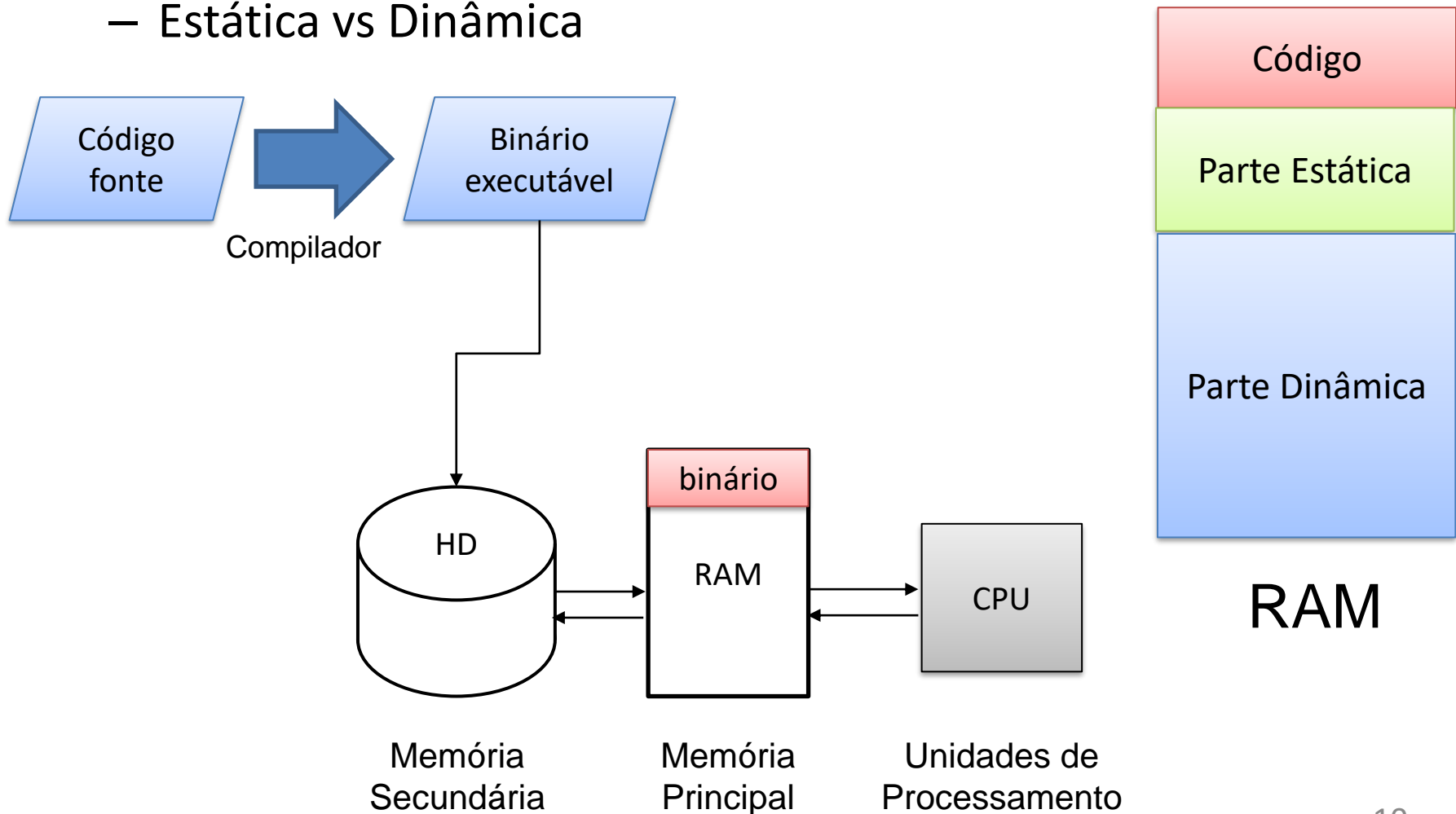
- **Cenário de dados atual: Big Data**
 - Importância/Desafio de manipular dados com ênfase no cenário atual
- **Processo de Compilação/Interpretação:**
 - Execução de uma programa escrito em uma linguagem de alto nível
- **Alocação de Memória – Endereçamento dos dados**
 - Estática vs Dinâmica
- **Linguagem de Programação Python**
 - Conceitos Básicos
 - Métodos
 - Classes e Objetos

O que vimos até agora ...

- **Cenário de dados atual: Big Data**
 - Importância/Desafio de manipular dados com ênfase no cenário atual
- **Processo de Compilação/Interpretação:**
 - Execução de uma programa escrito em uma linguagem de alto nível
- **Alocação de Memória – Endereçamento dos dados**
 - Estática vs Dinâmica
- **Linguagem de Programação Python**
 - Conceitos Básicos
 - Métodos
 - Classes e Objetos

O que vimos até agora ...

- **Alocação de Memória – Endereçamento dos dados**
 - Estática vs Dinâmica



O que vimos até agora ...

- **Cenário de dados atual: Big Data**
 - Importância/Desafio de manipular dados com ênfase no cenário atual
- **Processo de Compilação/Interpretação:**
 - Execução de uma programa escrito em uma linguagem de alto nível
- **Alocação de Memória – Endereçamento dos dados**
 - Estática vs Dinâmica
- **Linguagem de Programação Python**
 - Conceitos Básicos
 - Métodos
 - Classes e Objetos

O que vimos até agora ...

- **Cenário de dados atual: Big Data**
 - Importância/Desafio de manipular dados com ênfase no cenário atual
- **Processo de Compilação/Interpretação:**
 - Execução de uma programa escrito em uma linguagem de alto nível
- **Alocação de Memória – Endereçamento dos dados**
 - Estática vs Dinâmica
- **Linguagem de Programação Python**
 - Conceitos Básicos de Programação Orientada a Objetos
 - Métodos
 - Classes e Objetos

O que vimos até agora ...



- **Linguagem de Programação Python**

- Conceitos Básicos de Programação Orientada a Objetos
- Métodos
- Classes e Objetos

- Linguagem de Programação Interpretada
- Linguagem orientada a objetos
- Interativa
- Alta produtividade:
 - Foco no problema
- Ampla comunidade

Exercício



O que vimos até agora ...



- **Linguagem de Programação Python**

- Conceitos Básicos de Programação Orientada a Objetos
- Métodos
- Classes e Objetos

- Linguagem de Programação Interpretada
- Linguagem orientada a objetos
- Interativa
- Alta produtividade:
 - Foco no problema
- Ampla comunidade

O que vimos até agora ...



- **Linguagem de Programação Python**

- Conceitos Básicos de Programação Orientada a Objetos
- Métodos
- Classes e Objetos

- Linguagem de Programação Interpretada
- Linguagem orientada a objetos
- Interativa

- A

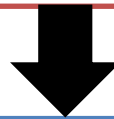
Síntese:

- - Estruturas de Dados são fundamentais
- A - Linguagens de Programação: Compilar vs Interpretar
- Alocação dos dados Dinâmica

Abre Parênteses...

COMO O ALUNO COMEÇA?!

Foco inicial – Lógica + linguagem de programação (várias linguagens)

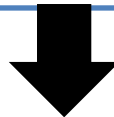


Dando Continuidade...

Foco Principal: precisamos dar atenção para questões mais genéricas, como exemplo, tipos de dados 'próprios' (POO), estrutura de dados lineares

Estrutura de dados – tipicamente tem como principal objetivo **organizar** dados.

Ideia Geral de ED: **armazenar** esses dados de uma forma eficiente conseguimos **recuperar** (buscar) por esses dados de uma forma mais trivial (custo computacional)



Estrutura de Dados: realizar o armazenamento da forma mais organizada possível (dependendo do domínio do problema) para que possamos posteriormente realizar a recuperação dos dados mais trivial (menor) custo computacional

Por que estudar Estruturas de Dados?

- Organização da Informação
- Melhora no Desempenho
- Proporciona o reuso de código
- Proporciona interoperabilidade
- Diminui custos

Pergunta-Chave: Como são os dados manipulados são representados?!

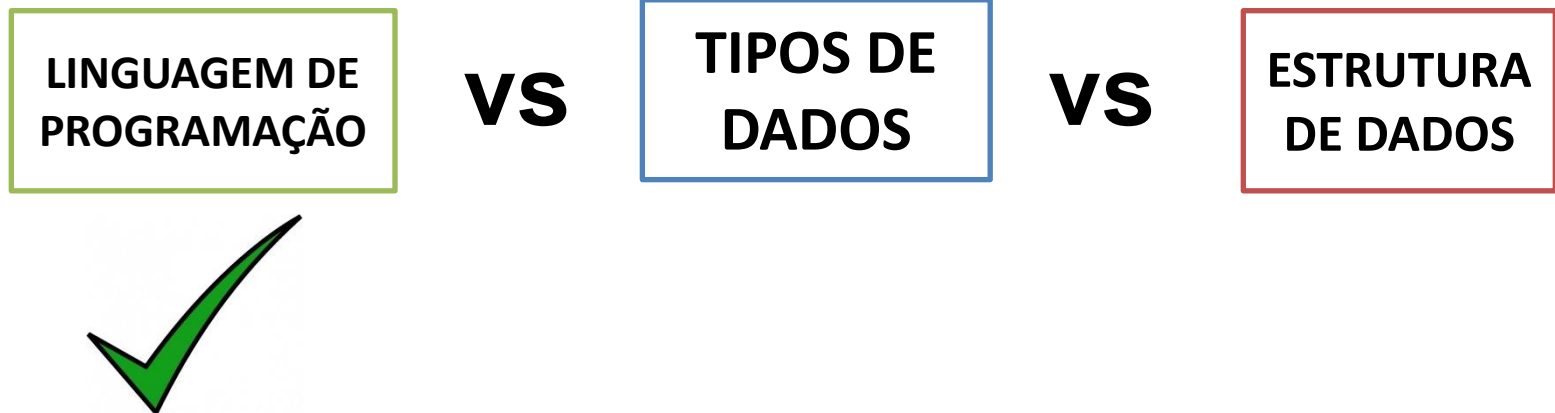
Estrutura de Dados

- Conhecimentos necessários para entender diferentes estruturas de dados...
 - O que vimos até agora?
- **Primeiro Foco:** entender a **DISTINÇÃO** entre:



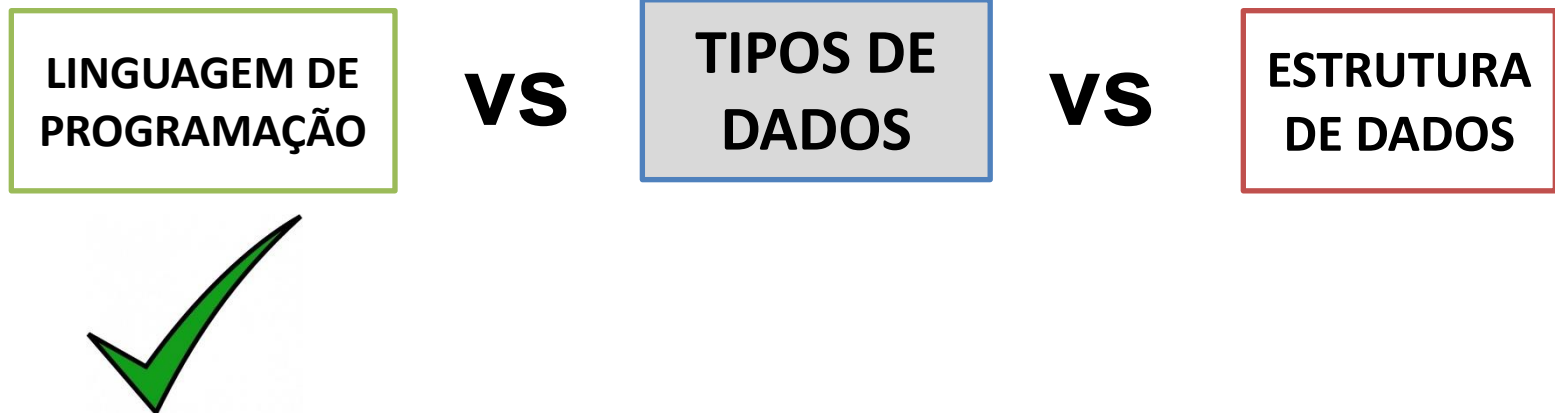
Estrutura de Dados

- Conhecimentos necessários para entender diferentes estruturas de dados...
 - O que vimos até agora?
- **Primeiro Foco:** entender a **DISTINÇÃO** entre:



Estrutura de Dados

- Conhecimentos necessários para entender diferentes estruturas de dados...
 - O que vimos até agora?
- **Primeiro Foco:** entender a **DISTINÇÃO** entre:



Estrutura de Dados

- Tipos de Dados

Tipo	Tamanho	Descrição
boolean	-----	Booleano
char	16 bits	Caracter
byte	8 bits	Byte
int	32 bits	Inteiro
long	64 bits	Longo
float	32 bits	Float
Double	64 bits	Double
void	-----	void

Estrutura de Dados

- Tipos de Dados

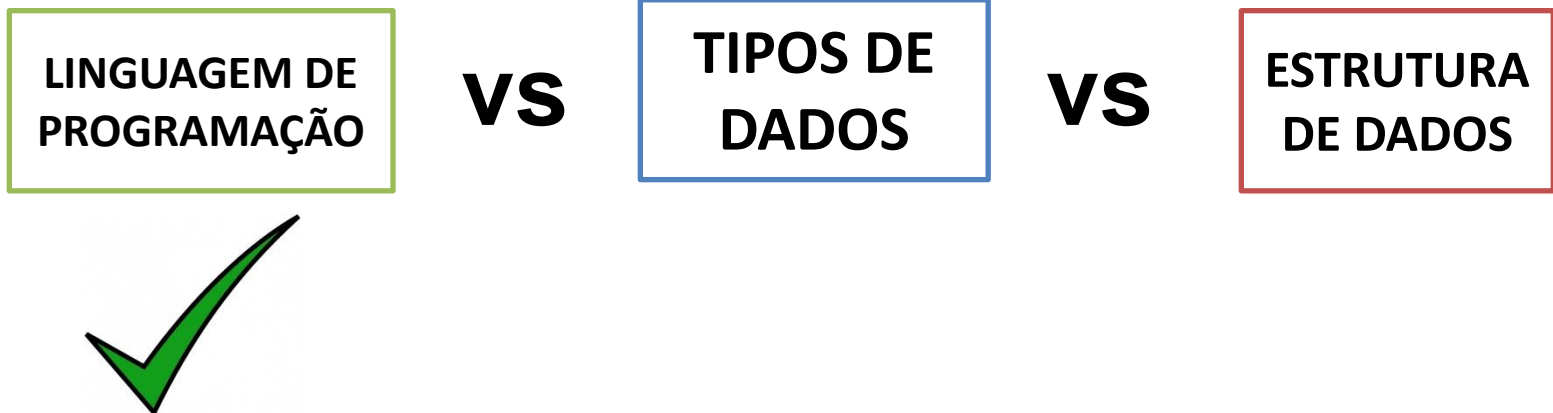
Classificação	Tipo	Descrição
Lógico	boolean	Pode possuir os valores true (verdadeiro) ou false (falso)
Inteiro	byte	Abrange de -128 a 127 (8 bits)
	short	Abrange de -32768 a 32767 (16 bits)
	int	Abrange de -2147483648 a 2147483647 (32 bits)
	long	Abrange de -2^{63} a $(2^{63})-1$ (64 bits)
Ponto Flutuante	float	Abrange de 1.40239846^{-46} a 3.40282347^{+38} com precisão simples (32 bits)
	double	Abrange de $4.94065645841246544^{-324}$ a $1.7976931348623157^{+308}$ com precisão dupla (64 bits)
Caracter	char	Pode armazenar um caracteres unicode (16 bits) ou um inteiro entre 0 e 65535

Estrutura de Dados

- **Tipos de Dados em Python...?**
 - Linguagem Fracamente ou Fortemente Tipada?!
 - Por que?!

Estrutura de Dados

- Conhecimentos necessários para entender diferentes estruturas de dados...
 - O que vimos até agora?
- **Primeiro Foco:** entender a **DISTINÇÃO** entre:



Estrutura de Dados

- **Tipos de dados e estruturas de dados**

- Primeiro Objetivo: entender a **DISTINÇÃO** entre:

**TIPOS DE
DADOS**

VS

**ESTRUTURA
DE DADOS**

Consiste da definição do:

- **conjunto de valores** (denominado domínio) que uma variável pode assumir ao longo da execução de um programa **e do**
- **conjunto de operações** que podem ser aplicadas sobre ele.

Estrutura de Dados

- **Tipos de dados e estruturas de dados**

- Primeiro Objetivo: entender a **DISTINÇÃO** entre:

**TIPOS DE
DADOS**

VS

**ESTRUTURA
DE DADOS**

Consiste da definição do:

- **conjunto de valores** (denominado domínio) que uma variável pode assumir ao longo da execução de um programa **e do**
- **conjunto de operações** que podem ser aplicadas sobre ele.

Exemplo:

Tipo da dado Inteiro:

Exemplo: ...-2,-1, 0, 1, 2...

Operações: [+] [-] [/] [*]

Classificação: básico

Estrutura de Dados

- Tipos de dados e estruturas de dados
 - Primeiro Objetivo: entender a **DISTINÇÃO** entre:

TIPOS DE
DADOS

VS

ESTRUTURA
DE DADOS

Quando um tipo de dado é **básico**?

Quando **NÃO** possuem uma estrutura sobre os seus valores, isto é, **NÃO** é possível decompor em partes menores.

Estrutura de Dados

- Tipos de dados e estruturas de dados

– Primeiro Objetivo: entender a **DISTINÇÃO** entre:

TIPOS DE
DADOS

VS

ESTRUTURA
DE DADOS

Quando um tipo de dado é **básico**?

Quando **NÃO** possuem uma estrutura sobre os seus valores, isto é, **NÃO** é possível decompor em partes menores.

Portanto os tipos de dados básicos, em geral, são **INDIVISÍVEIS**

Tipos de dados básicos (em geral):

- Lógico
- Inteiro
- Real
- Caractere

Estrutura de Dados

- Tipos de dados e estruturas de dados

– Primeiro Objetivo: entender a **DISTINÇÃO** entre:

TIPOS DE
DADOS

VS

ESTRUTURA
DE DADOS

Quando um tipo de dado é **básico**?

Quando **NÃO** possuem uma estrutura sobre os seus valores, isto é, **NÃO** é possível decompor em partes menores.

Portanto os tipos de dados básicos, em geral, são **II**

Tipos de dados básicos (em geral):

- Lógico
- Inteiro
- Real

E o tipo de dados que criamos
Data?!

Estrutura de Dados

- **Tipos de dados e estruturas de dados**

- Primeiro Objetivo: entender a **DISTINÇÃO** entre:



Tipos de dados Estruturados:

- Permitem **agregar** mais de uma valor em uma variável, existindo uma **relação ESTRUTURAL** entre os elementos que o compõem.
- **Dados** compostos com **mais de um campo**
- Linguagens de programação fornecem mecanismos para que possamos **estruturar dados mais complexos**, os nossos dados

Estrutura de Dados

- **Tipos de dados e estruturas de dados**

- Primeiro Objetivo: entender a **DISTINÇÃO** entre:

TIPOS DE
DADOS

VS

ESTRUTURA
DE DADOS

Tipos de dados Estruturados:

- Permitem **agregar** mais de um valor em uma variável, existindo uma **relação ESTRUTURAL** entre os elementos.
- **Dados** compostos por outros dados.
- Linguagens de programação possuem mecanismos para criar **dados mais complexos**.

Exemplos de Tipos de dados Estruturados Primitivos (em Geral):

- **Arranjos (vetores, matrizes)**
 - **Tamanho pré-definido**
 - **Vetores:** coleções ordenadas do mesmo tipo
 - **Matrizes:** vetores de mais de uma dimensão

Estrutura de Dados

- Tipos de dados e estruturas de dados

- Primeiro Objetivo: entender a **DISTINÇÃO** entre:



Tipos de dados DEFINIDOS PELO USUÁRIO:

- Também conhecidos como dados estruturados
- Constituído por um conjunto de componentes que podem ser diferentes tipos e agrupados em um único nome (variável)

Estrutura de Dados

- **Tipos de dados e estruturas de dados**

- Primeiro Objetivo: entender a **DISTINÇÃO** entre:



Tipos de dados DEFINIDOS PELO USUÁRIO:

- Também conhecidos como dados estruturados
- Constituído por um conjunto de componentes que podem ser diferentes tipos e agrupados em um único nome (variável)

A partir deste momento a gente precisa focar em:
COMO CRIAR OS NOSSOS TIPOS DE DADOS ESTRUTURADOS!?!?

Estrutura de Dados

- **Tipos de dados e estruturas de dados**

- Primeiro Objetivo: entender a **DISTINÇÃO** entre:



Tipos de dados DEFINIDOS PELO USUÁRIO:

- Também conhecidos como dados estruturados
- Constituído por um conjunto de componentes que podem ser diferentes tipos e agrupados em um único nome
- **Construção Hierárquica:** manipulação dos dados de forma organizada.

Estrutura de Dados

- Tipos de dados e estruturas de dados
 - Primeiro Objetivo: entender a **DISTINÇÃO** entre:

TIPOS DE
DADOS

Tipos de dados DEFINIDOS PELO USUÁRIO

- Também conhecidos como dados estruturados
- Constituído por um conjunto de componentes que podem ser diferentes tipos e agrupados em um único nome
- **Construção Hierárquica:** manipulação de dados de forma organizada.

ESTRUTURA
DE DADOS

Estrutura de Dados

- **Introdução**

- **Estrutura de dados:**

- Especificam “conceitualmente” os dados
 - Refletem um **relacionamento** lógico entre os **dados e o domínio do problema**
 - Incluem **operações** para manipulação dos seus dados:
 - Essas operações também tem um papel na caracterização do problema/domínio

Estrutura de Dados

- **Estrutura de dados:**
 - **Tipos de Dados Abstratos (TADs)**
 - São estruturas de dados capazes de representar os tipos de dados que NÃO foram previstos no núcleo das linguagens de programação
 - E
 - Que são necessários para aplicações específicas.

Estrutura de Dados

- **Estrutura de dados:**
 - **Tipos de Dados Abstratos (TADs)**
 - São divididas em duas partes:
 - Os dados
 - E as operações

A especificação de um TAD requer a:

- escolha de uma **maneira adequada de armazenar os dados**
- à definição de um **conjunto adequado de operações** para atuar sobre eles.

Estrutura de Dados

- **Estrutura de dados:**
 - **Tipos de Dados Abstratos (TADs)**
 - Característica Essencial de um TAD
 - É a separação entre conceito e implementação, ou seja, existe uma distinção entre:
 - » a definição do tipo **e**
 - » a sua representação **e**
 - » a implementação das operações.

Estrutura de Dados

- **Estrutura de dados:**
 - **Tipos de Dados Abstratos (TADs)**
 - Característica Essencial de um TAD
 - É a separação entre conceito e implementação, ou seja, existe uma distinção entre:
 - » a definição do tipo e
 - » a sua representação e
 - » a implementação das operações.

Um TAD é, portanto, uma forma de definir um novo tipo de dado **ESTRUTURADO** juntamente com as operações que manipulam esse novo tipo de dado estruturado criado por nós!

Estrutura de Dados

- **Estrutura de dados:**
 - **Tipos de Dados Abstratos (TADs)**
 - Formalmente um TAD é definido por um par:
 - (V, \circ) :
 - » onde V representa o conjunto de valores
 - » E \circ representa o conjunto de operações aplicáveis sobre V

Estrutura de Dados

- **Estrutura de dados:**
 - **Tipos de Dados Abstratos (TADs)**
 - Exemplo de um tipo de dado utilizado com frequência e que não é definido primitivamente nas linguagens de programação:
 - **Tipo Data**

Estrutura de Dados

- **Estrutura de dados:**
 - **Tipos de Dados Abstratos (TADs)**
 - Exemplo de um tipo de dado utilizado com frequência e que não é definido primitivamente nas linguagens de programação:
 - **Tipo Data**

Importância do Tipo Data: exemplos de necessidade/aplicações:
Seguidamente precisamos fazer cálculos envolvendo datas e nem sempre isto é fácil. Por exemplo, calendários, agendas (em diferentes contextos).

Estrutura de Dados

- **Estrutura de dados:**
 - **Tipos de Dados Abstratos (TADs)**
 - Exemplo de um tipo de dado utilizado com frequência e que não é definido primitivamente nas linguagens de programação:
 - **Tipo Data**

Importância do Tipo Data: exemplos de necessidade/aplicações:
Seguidamente precisamos fazer cálculos envolvendo datas e nem sempre isto é fácil. Por exemplo, calendários, agendas (em diferentes contextos).

Vamos começar a definir hierarquicamente o nosso tipo de dado estruturado
DATA!

Estrutura de Dados

- Estrutura de dados:

- Tipos de Dados Abstratos (TADs)

- **Tipo Data:** para representar uma data pode ser dado pelo par (v, o) , onde:

- v é uma **tripla** formada por: **dia-mês-ano**
 - o são **operações** aplicáveis sobre o tipo Data

- » Exemplos de operações:

- Verificar se uma data é válida
 - Calcular o dia da semana de uma determinada data
 - Calcular a data do Carnaval de um determinado ano
 - Entre outras ...

Estrutura de Dados

- Estrutura de dados:
 - Tipos de Dados Abstratos (TADs)
 - **Tipo Data:** para representar uma data pode ser dado pelo par (v, \circ) , onde:
 - v é uma **tripla** formada por: dia-mês-ano
 - \circ são operações aplicáveis sobre o tipo Data
 - » Exemplos de operações:
 - Verificar se uma data é válida
 - Calcular o dia da semana de uma determinada data
 - Calcular a data do Carnaval de um determinado ano
 - Entre outras ...

Vamos desenvolver nosso TAD Data?!

Estrutura de Dados

- Estrutura de dados:
 - Tipos de Dados Abstratos (TADs)
 - **Tipo Data:** para representar uma data pode ser dado pelo par (v, \circ) , onde:
 - v é uma **tripla** formada por: dia-mês-ano
 - \circ são operações aplicáveis sobre o tipo Data
 - » Exemplos de operações:
 - Verificar se uma data é válida
 - Calcular o dia da semana de uma determinada data
 - Calcular a data do Carnaval de um determinado ano
 - Entre outras ...

Vamos desenvolver nosso TAD Data?!

Estrutura de Dados

- Estrutura de dados:

- Tipos de Dados Abstratos (TADs)

- Tipo Data:

- Passos:

- 1. Caracterizar o tipo de dado

```
Data: estrutura de dado / registro  
Dia: inteiro  
Mês: inteiro  
Ano: inteiro
```

- 2. Criando a estrutura de dado DATA (*DateStruct*):

- Três campos do tipo inteiro para representar dia, mês e ano

Estrutura de Dados

- Estrutura
- Tipos

```
class DateStruct:  
    def __init__(self, day, month, year):  
        self.day = day  
        self.month = month  
        self.year = year
```

- Tipo Data:

- Passos:

1. Caracterizar o tipo de dado ✓
2. Criando a estrutura de dado DATA (*DateStruct*): ✓
 - Três campos do tipo inteiro para representar dia, mês e ano

```
Data: estrutura de dado / registro  
Dia: inteiro  
Mês: inteiro  
Ano: inteiro
```

Estrutura de Dados

- Estrutura de dados:
 - Tipos de Dados Abstratos (TADs)
 - Tipo Data:
 - Passos:
 1. Caracterizar o tipo de dado ✓
 2. Criando a estrutura de dado DATA (*DateStruct*): ✓
 - Três campos do tipo inteiro para representar dia, mês e ano

```
Data: estrutura de dado / registro  
Dia: inteiro  
Mês: inteiro  
Ano: inteiro
```
 3. Definir as operações (em andamento)

Estrutura de Dados

Data: estrutura de dado / registro

Dia: inteiro

Mês: inteiro

Ano: inteiro

- **Estrutura de dados:**

- **Tipos de Dados Abstratos (TADs)**

- **Tipo Data:**

- Passos:

- 3. Definir de um conjunto de operações (métodos) que podem ser executados sobre o novo tipo de dado

Método ***inicializarData***

Entradas: dia, mês, ano -> inteiro

Saída: D(data)

Objetivo: método que recebe três parâmetros inteiros, informando Dia, Mês e Ano e retorna D, com a data inicializada

Estrutura de Dados

Data: estrutura de dado / registro

Dia: inteiro

Mês: inteiro

Ano: inteiro

- **Estrutura de dados:**

- **Tipos de Dados Abstratos (TADs)**

- **Tipo Data:**

- Passos:

- 6. Especificação de um conjunto de operações (métodos) que podem ser executados sobre o novo tipo de dado

Método ***acrescentarDias***

Entradas: D (data), Dias (inteiro)

Saída: (Data)

Objetivo: método que recebe soma um determinado número de *Dias* a uma data recebida como parâmetro e retorna o resultado. Caso não seja possível realizar a operação, o método retorna uma data cujo Dia seja -1

Estrutura de Dados

Data: estrutura de dado / registro

Dia: inteiro

Mês: inteiro

Ano: inteiro

- **Estrutura de dados:**

- **Tipos de Dados Abstratos (TADs)**

- **Tipo Data:**

- Passos:

- 6. Especificação de um conjunto de operações (métodos) que podem ser executados sobre o novo tipo de dado

Método ***escreverExtenso***

Entrada: D (data)

Saída: Lógico

Objetivo: método que recebe uma data e a escreve por extenso. Por exemplo: 10/03/2020 deve ser escrito como 10 de março de 2020. Retorna verdadeiro se a operação foi realizada com sucesso e falso, caso contrário.

Estrutura de Dados

- **Estrutura de dados:**
 - **Tipos de Dados Abstratos (TADs)**

Coesão e Coerência:

- A forma como um TAD é implementado torna-se um detalhe de implementação, que **não deve interferir** no uso do TAD em outros contextos.
- A separação entre a **definição** do TAD e sua **implementação** permite que alterações de implementação NÃO influenciem nas aplicações que utilizam o TAD.