

Universidade Federal do Rio Grande do Norte
Instituto Metr pole Digital

AULA 10

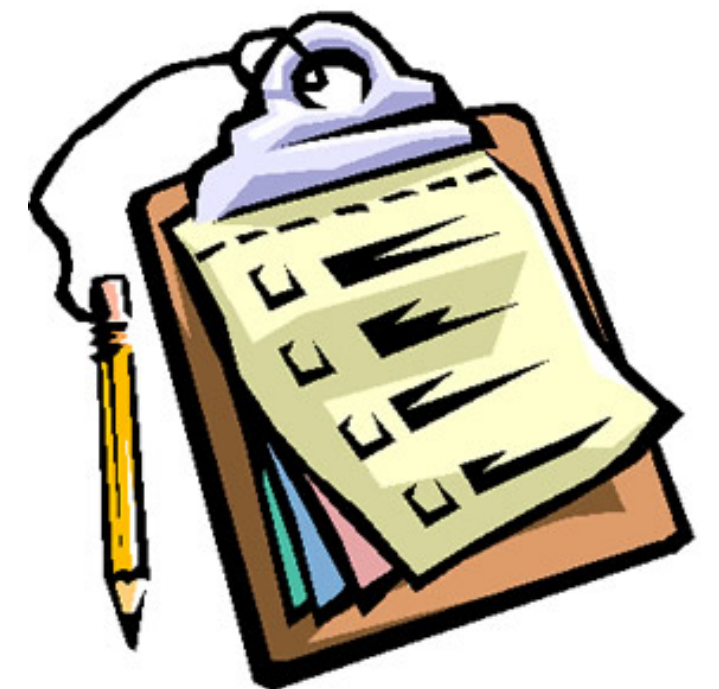
Estrutura de dados b sico I (EDB1)

Prof. Msc. Janiheryson Felipe (Felipe)

Natal, RN
2023

OBJETIVO DA AULA

- Introduzir os tipos abstratos de dados:
 - Definição;
 - Vantagens;
 - Desvantagens;
 - Usos pelos programas

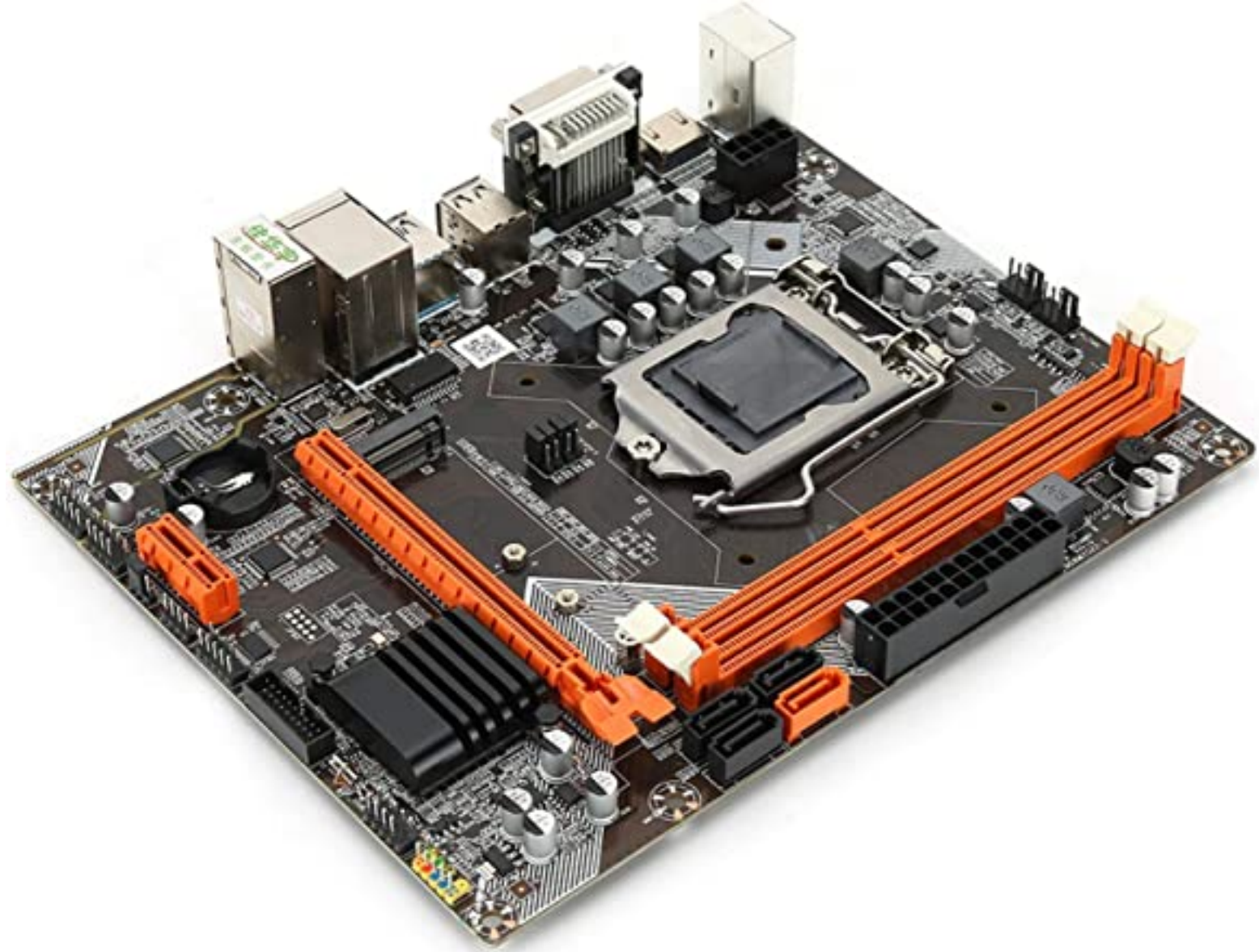




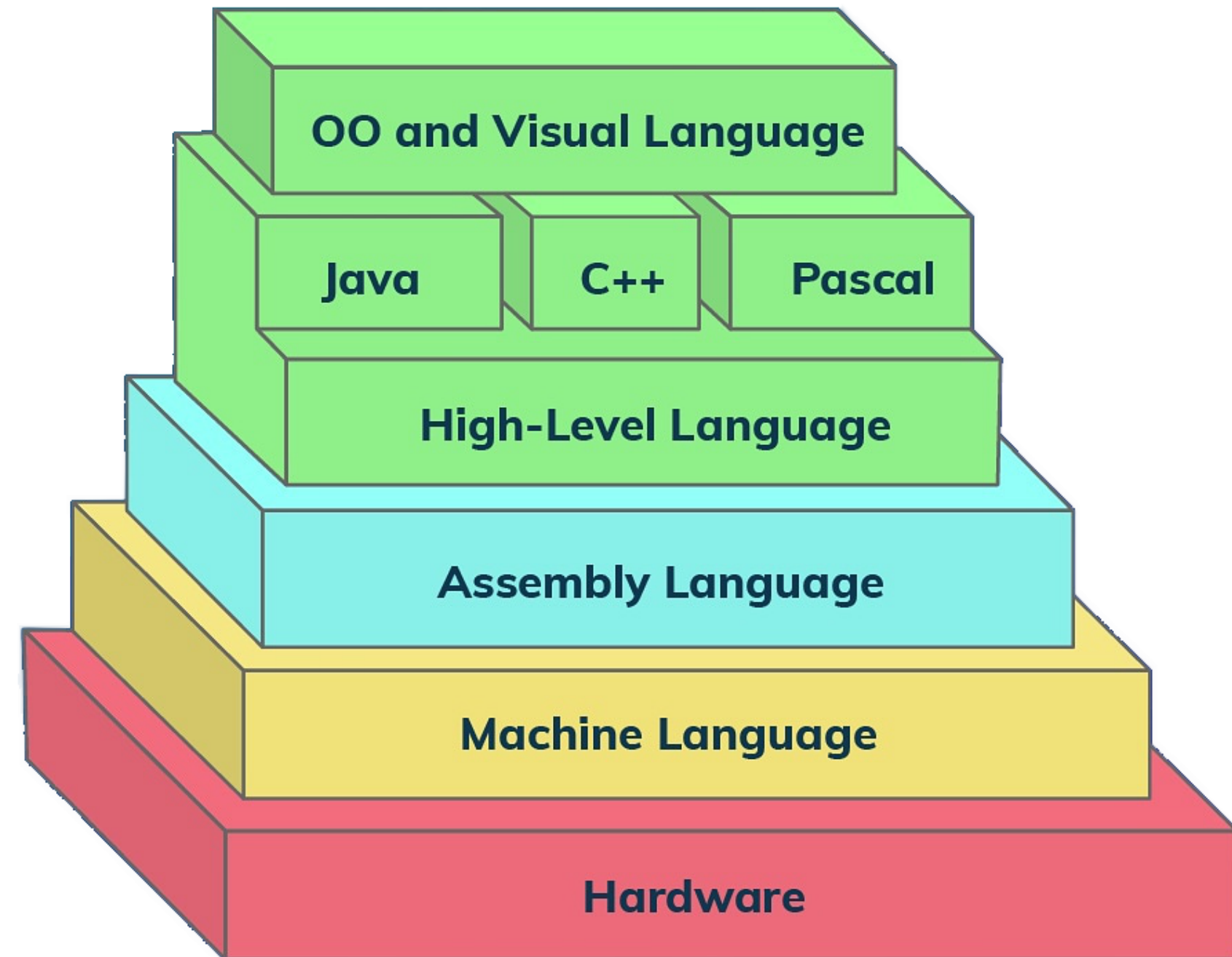
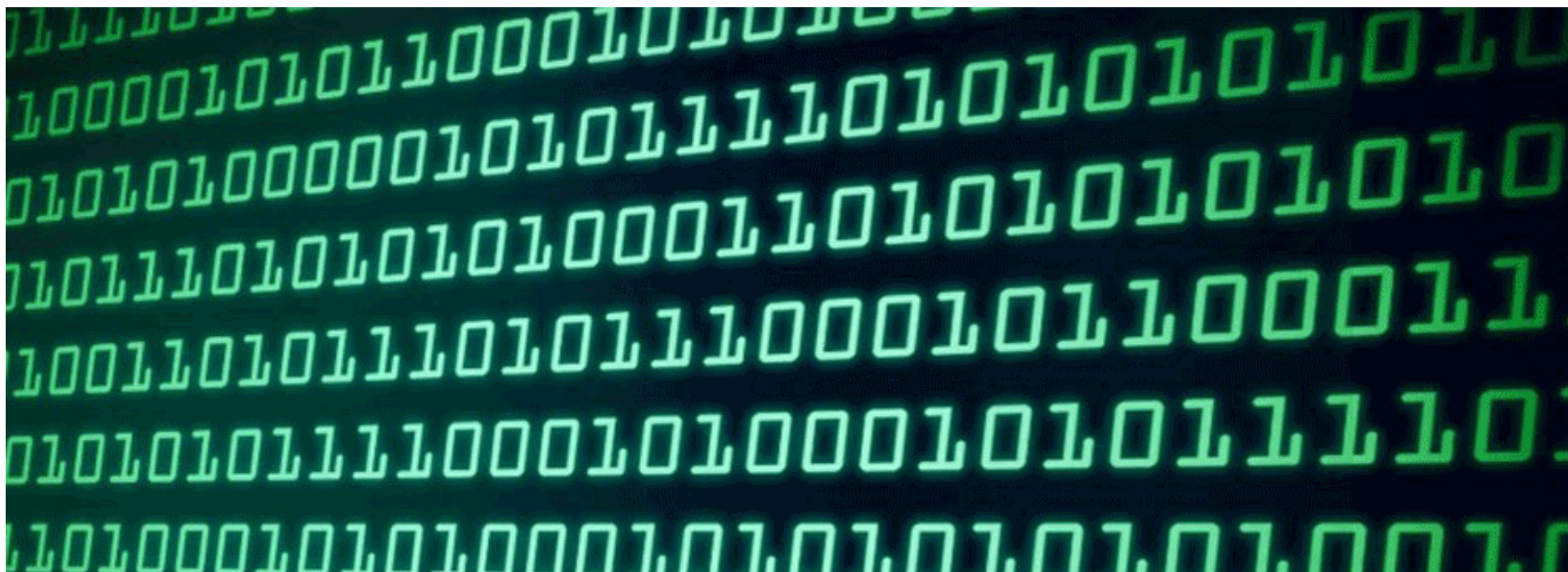
INTRODUÇÃO AOS TADS

**"A ciência da
computação é a ciência
da abstração"**





"A ciencia da
computação é a ciencia
da abstração"



TIPOS ABSTRATOS DE DADOS (TADS)

Os Tipos Abstratos de Dados (TADs) são um conceito fundamental na ciência da computação que se refere a uma abstração de uma estrutura de dados, independentemente da linguagem de programação usada.

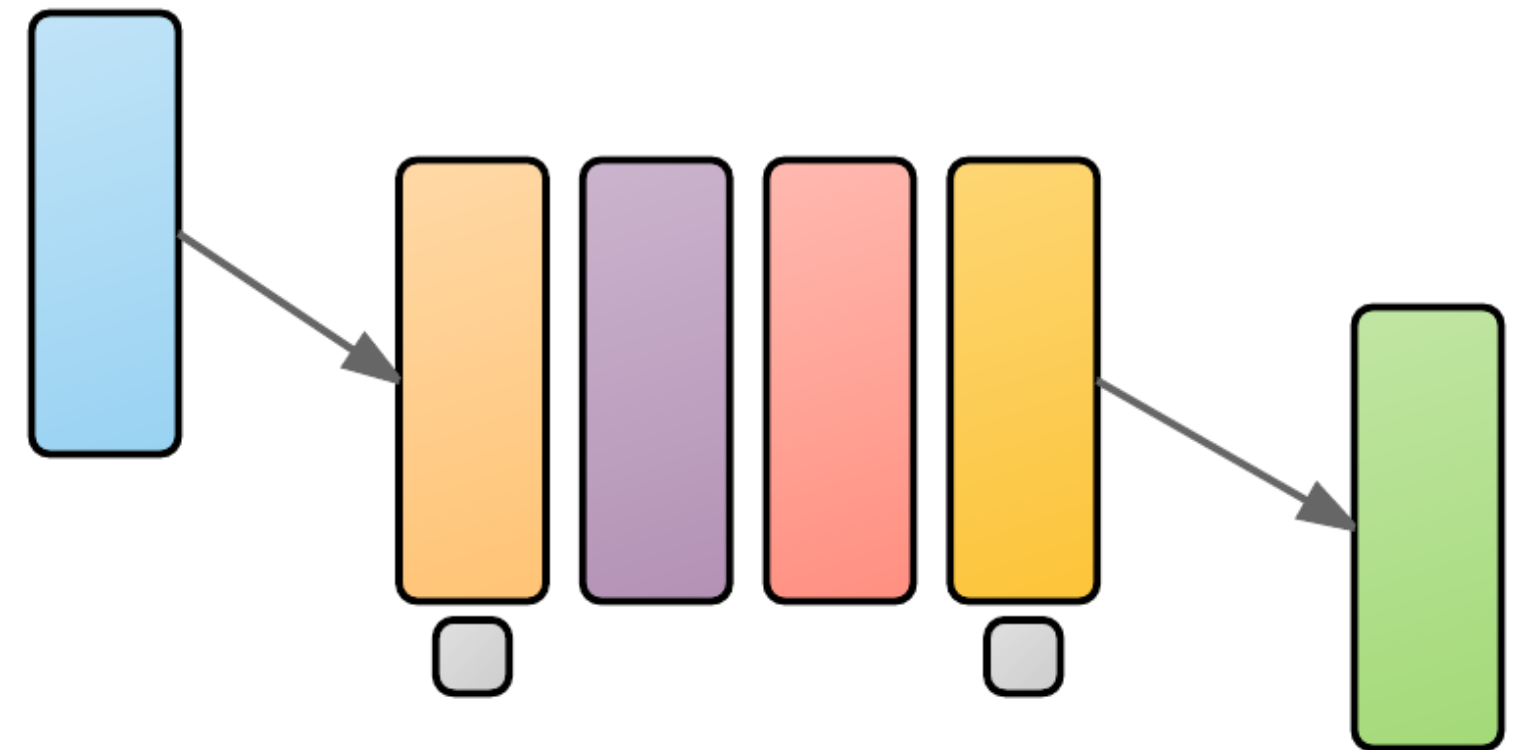
Os TADs fornecem uma descrição de como uma determinada estrutura de dados pode ser manipulada, sem se preocupar com a implementação detalhada dessa estrutura.

TIPOS ABSTRATOS DE DADOS (TADS)

As pilhas são comumente implementadas utilizando arrays ou listas ligadas, mas, uma implementação desnecessariamente complicada usando uma árvore de pesquisa binária, por exemplo, ainda é uma implementação possível e válida. Em outras palavras, é incorreto dizer que as pilhas são arrays ou vice-versa.

TIPOS ABSTRATOS DE DADOS (TADS)

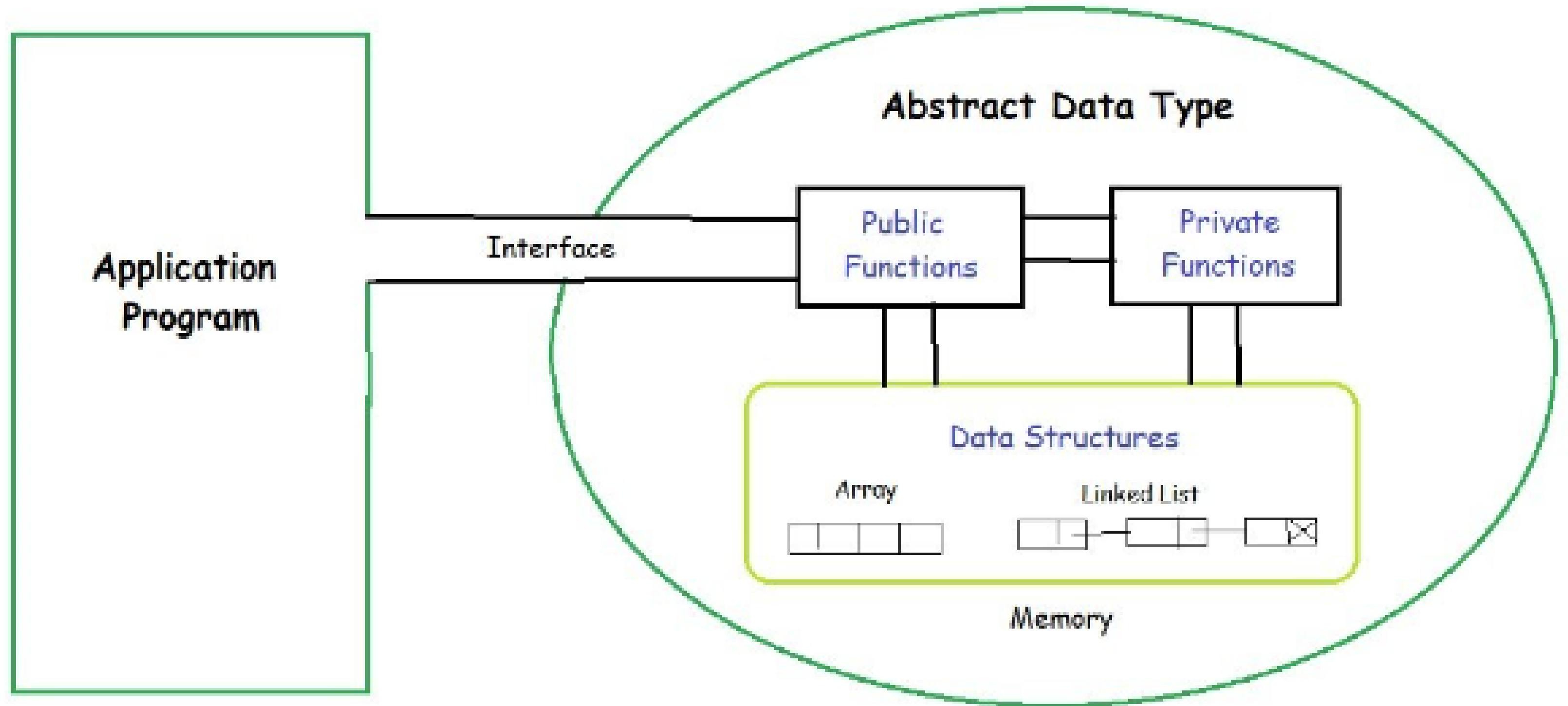
Os tipos de dados abstratos não especificam uma implementação, isso significa que também é incorreto falar sobre a complexidade de tempo de um determinado tipo de dados abstratos.



TIPOS ABSTRATOS DE DADOS (TADS)

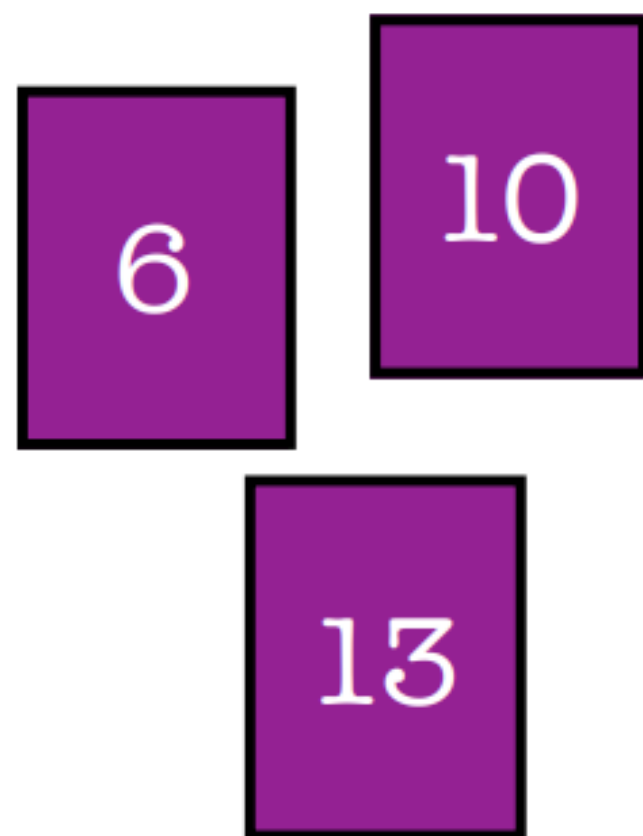
Abstract Data type (ADT) é um tipo (ou classe) para objetos cujo comportamento é definido por um conjunto de valores e um conjunto de operações. A definição de ADT menciona apenas quais operações devem ser executadas, mas não como essas operações serão implementadas. Não especifica como os dados serão organizados na memória e quais algoritmos serão usados para implementar as operações. É chamado de “abstrato” porque oferece uma visão independente da implementação.

TIPOS ABSTRATOS DE DADOS (TADS)



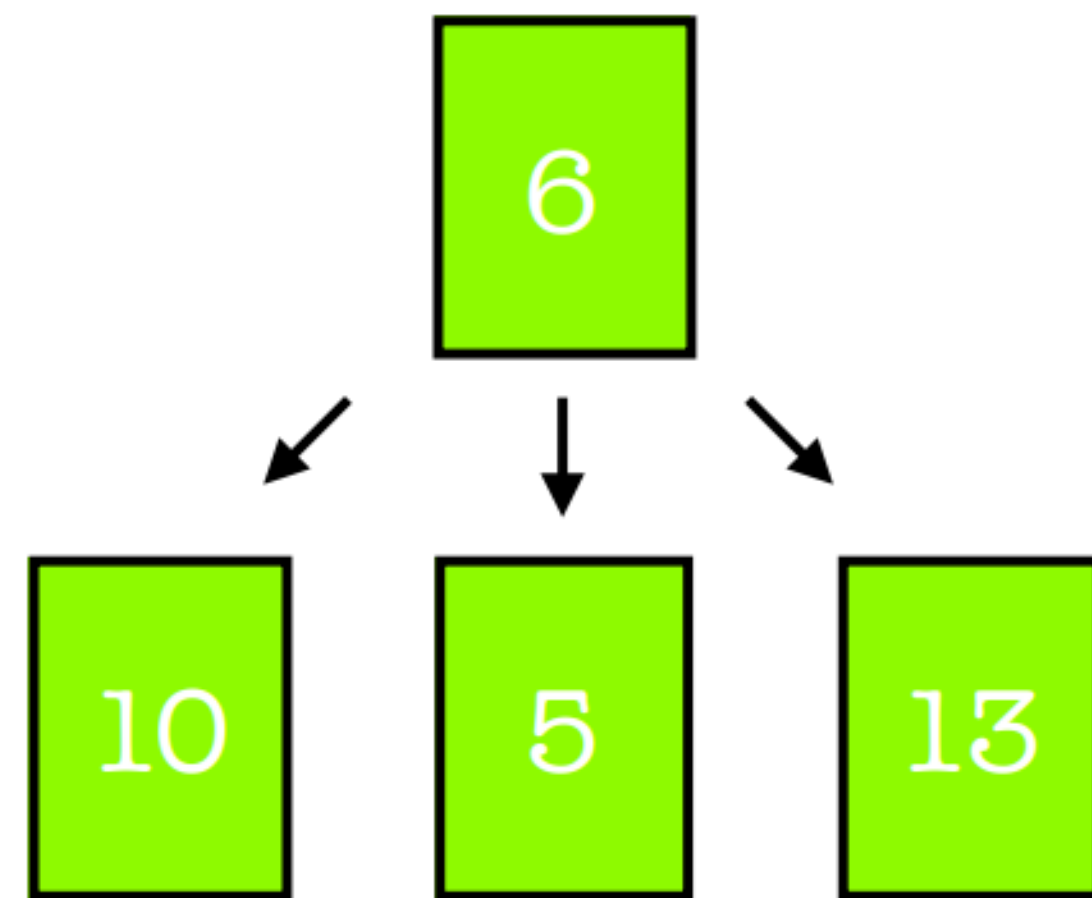
(TADS) : LISTAS - EXEMPLO DE MÉTODOS

- `get()` – Retorna um elemento da lista em qualquer posição.
- `insert()` – Insere um elemento em qualquer posição da lista.
- `remove()` – Remove a primeira ocorrência de qualquer elemento de uma lista não vazia.
- `removeAt()` – Remove o elemento em um local especificado de uma lista não vazia.
- `replace()` – Substitui um elemento em qualquer posição por outro elemento.
- `size()` – Retorna o número de elementos na lista.
- `isEmpty()` – Retorna verdadeiro se a lista estiver vazia, caso contrário, retorna falso.
- `isFull()` – Retorna verdadeiro se a lista estiver cheia, caso contrário, retorna falso.

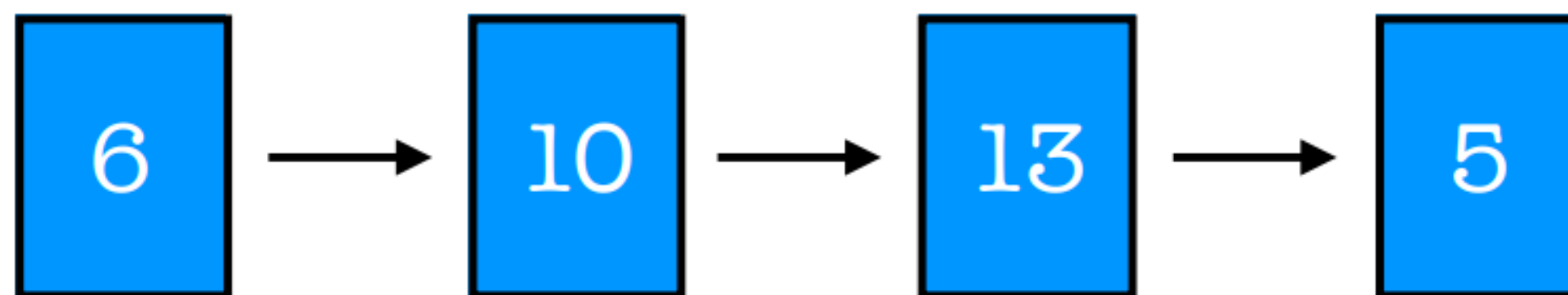


TADs

(não-relacionais)



(hierárquicos)



(lineares)

conjunto		dicionário
----------	--	------------

(não-relacional)

lista ordenada
manualmente

lista ordenada automaticamente

TADs

pilha

deque

fila

(linear)

Lista Circular

Lista Encadeada

Tabela de Dispersão

VANTAGENS DO USO DE TADS

- **Encapsulamento:** fornecem uma maneira de encapsular dados e operações em uma única unidade.
- **Abstração:** permitem que os usuários trabalhem com estruturas de dados sem precisar conhecer os detalhes da implementação, o que pode simplificar a programação e reduzir erros.
- **Independência da estrutura de dados:** os ADTs podem ser implementados usando diferentes estruturas de dados, o que pode facilitar a adaptação às necessidades e requisitos em constante mudança.

VANTAGENS DO USO DE TADS

- **Ocultação de informações:** os ADTs podem proteger a integridade dos dados, controlando o acesso e impedindo modificações não autorizadas.
- **Modularidade:** os ADTs podem ser combinados com outros ADTs para formar estruturas de dados mais complexas, o que pode aumentar a flexibilidade e a modularidade na programação.

DESVANTAGENS DO USO DE TADS

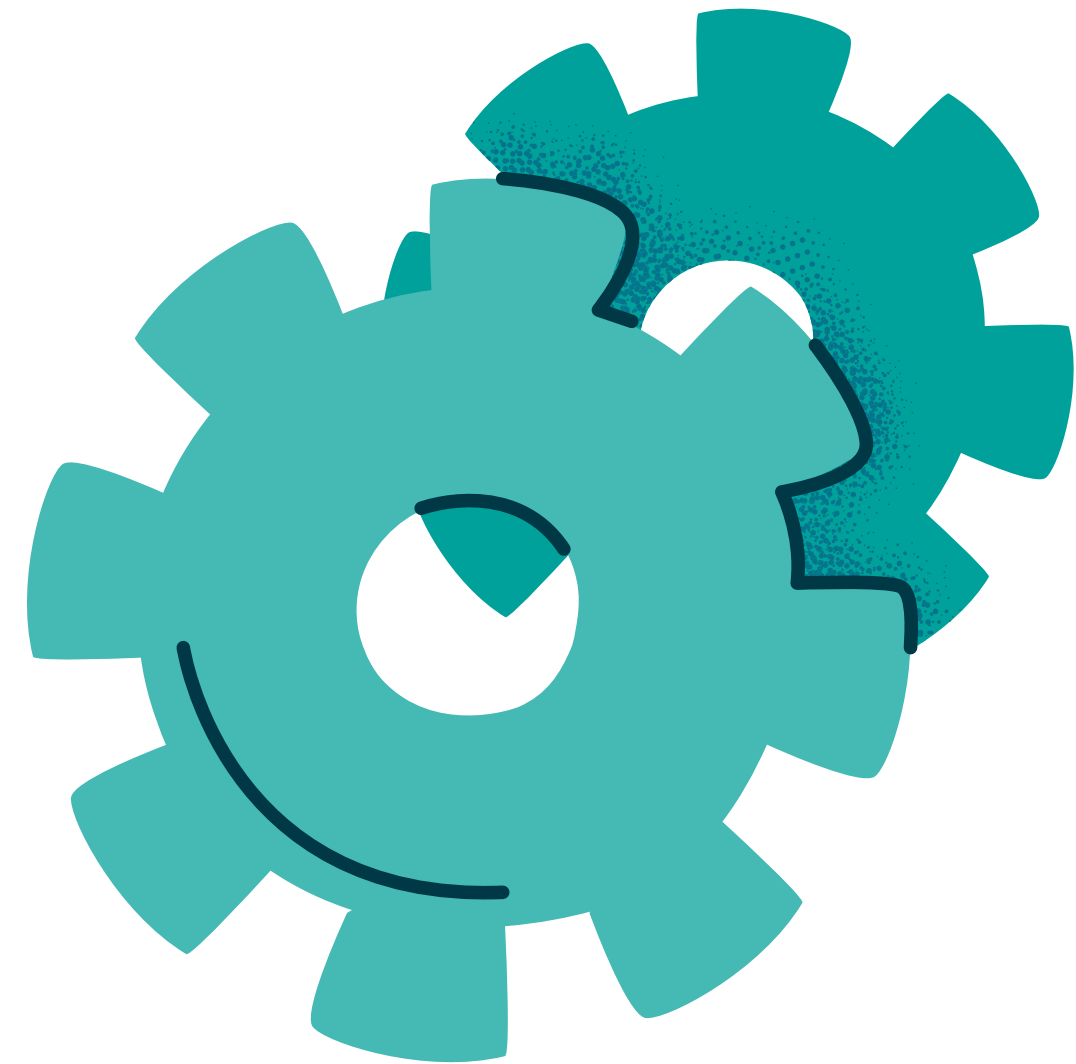
- **Sobrecarga:** A implementação de TADs pode adicionar sobrecarga em termos de memória e processamento, o que pode afetar o desempenho.
- **Complexidade:** TADs podem ser complexos de implementar, especialmente para estruturas de dados grandes e complexas.
- **Curva de aprendizado:** o uso de TADs requer conhecimento de sua implementação e uso, o que pode levar tempo e esforço para aprender.

DESVANTAGENS DO USO DE TADS

- **Flexibilidade Limitada:** Alguns TADs podem ser limitados em sua funcionalidade ou podem não ser adequados para todos os tipos de estruturas de dados.
- **Custo:** A implementação de TADs pode exigir recursos e investimentos adicionais, o que pode aumentar o custo de desenvolvimento.

TIPOS ABSTRATOS DE DADOS (TADS)

- Listas sequenciais;
- Listas simplesmente encadeadas;
- Listas duplamente encadeadas;
- filas, pilhas e dequeues, heap (fila prioritaria);
- Tabelas de dispersão;
- Conjuntos (sets);
- Dicionarios e Hash map;
- Vetores circulares e auto-ajustaveis.



QUESTÃO 01 - BANCA: FCC

Em relação a tipos abstratos de dados, é correto afirmar que

- (A) o TAD não encapsula a estrutura de dados para permitir que os usuários possam ter acesso a todas as operações disponibilizadas sobre esses dados.
- (B) algumas pilhas admitem serem declaradas como tipos abstratos de dados.
- (C) filas não permitem declaração como tipos abstratos de dados.
- (D) os tipos abstratos de dados podem ser formados pela união de tipos de dados primitivos, mas não por outros tipos abstratos de dados.
- (E) são tipos de dados que escondem a sua implementação de quem o manipula; de maneira geral as operações sobre estes dados são executadas sem que se saiba como isso é feito.

DÚVIDAS???

