

Introdução à disciplina

Prof. Allan Rodrigo Leite
ESTD – Estruturas de dados



Estruturas de dados

- Definem modelos para organizar e armazenar dados em sistemas computacionais
 - Também estabelecem políticas para acesso aos dados armazenados
- Cada estrutura de dados possui características distintas, aplicáveis em diferentes propósitos
 - Buscam a eficiência no armazenamento e na recuperação de dados

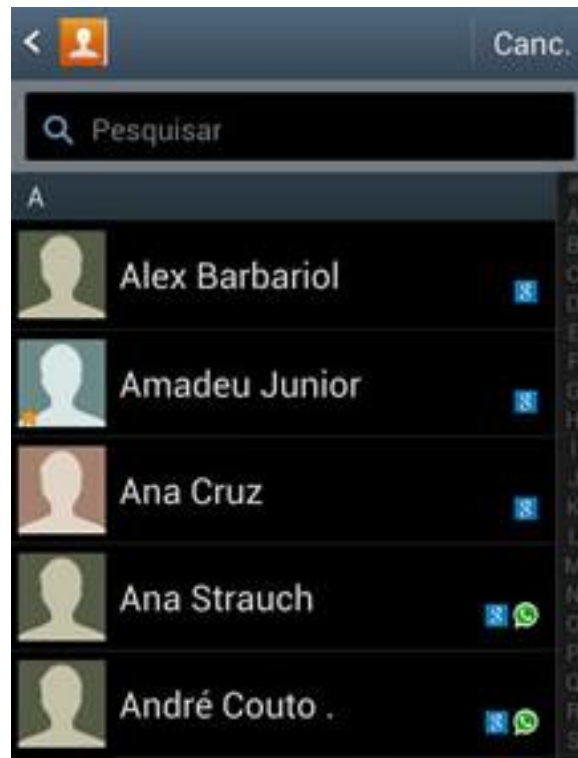


Estruturas de dados

- Existem diferentes tipos de estruturas de dados, sendo as principais:
 - Lineares
 - Pilhas, Filas, Listas
 - Hierárquicas
 - Árvores
 - Complexas
 - Grafos, Mapas, Tabelas

Estruturas de dados

- Exemplo 1: armazenar e localizar contatos telefônicos em uma agenda



Estruturas de dados

- Exemplo 2: encontrar o brasileiro mais velho na base de dados do IBGE



Estruturas de dados

- Exemplo 3: encontrar a melhor rota de viagem para um operador logístico





Metodologia da disciplina

- Aulas teóricas

- Conceitos e definições de estruturas de dados e algoritmos

- Aulas práticas

- Desenvolvimento de estruturas em Java
 - Utilização da orientação a objetos



Ementa da disciplina

- Estruturas de dados lineares
 - Pilhas estáticas e dinâmicas
 - Filas estáticas e dinâmicas
 - Listas
- Estruturas de dados hierárquicas
 - Conceitos de árvores
 - Árvores binárias, AVL e B
 - Técnicas de busca



Ementa da disciplina

- Métodos de ordenação
 - Complexidade algorítmica
 - Análise assintótica
 - Algoritmos de ordenação
- Estruturas de dados avançadas
 - Mapas e tabelas
 - Grafos
 - Técnicas de busca



Referências bibliográficas

- GOODRICH, Michael T.; TAMASSIA, Roberto.
Estruturas de dados e algoritmos em Java. 4. ed. Porto Alegre: Bookman, 2007.
- LAFORE, Robert. Estruturas de dados e algoritmos em Java. São Paulo: Ciência Moderna, 2005.
- SZWARCFITER, Jayme; MARKENSON, Lilian.
Estruturas de dados e seus algoritmos. 3. ed. São Paulo: LTC, 2010.



Referência complementar

- CORMEN, Thomas H.; et al. Algoritmos: Teoria e Prática. São Paulo: Editora Campus, 2012.

Estruturas básicas

- É comum a necessidade de armazenar um conjunto de informações relacionadas
 - Últimos vinte resultados das partidas de um time de futebol
 - Últimas dez melhores pontuações em um jogo de videogame
- Para não utilizar uma variável para cada valor do conjunto, pode-se utilizar vetores que são coleções numeradas de variáveis do mesmo tipo de dados
 - Cada variável em um vetor tem um índice, que identifica a posição onde o valor está armazenado
 - A numeração do índice se inicia do 0 até um limite predefinido
 - Arranjos podem ser definidos em Java utilizando o objeto `Array`



Vetores

■ Definição

- Um vetor é uma coleção homogênea de dados lineares e, isto é, pode armazenar um determinado tipo de dados
- Um vetor possui um tamanho finito e predefinido
- A velocidade de acesso aos dados de um vetor é praticamente instantânea e considerado constante

Vetores

■ Declaração de um vetor em Java

```
int[] valores = new int[10];
```



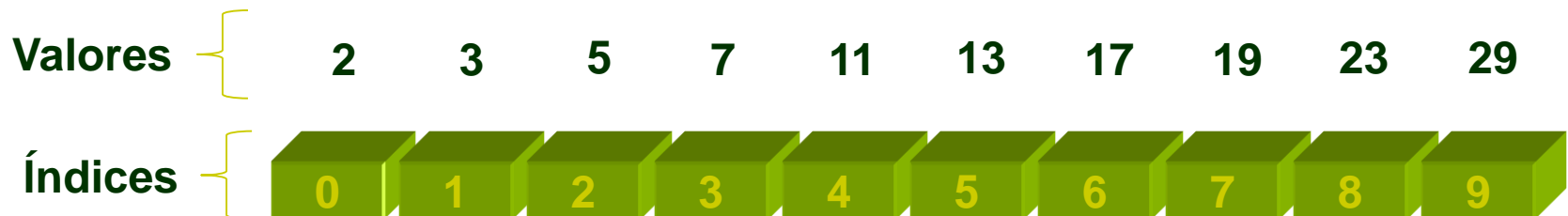
Vetores

- É possível recuperar o tamanho máximo de um vetor pelo atributo `length`

```
if (valores.length < 10) { ... };
```

- Vetores também podem ser iniciados no momento da sua declaração

```
int[] primos = { 2, 3, 5, 7, 11, 13, 17, 19, 23, 29 };
```



Exercício

- Crie uma classe para armazenar o score de um jogador, de acordo com a definição abaixo
- Em seguida, crie uma estrutura (vetor) para armazenar 10 scores, não é necessário que os scores estejam ordenados
- Por fim, crie um método para listar todos os scores

GameEntry

name : String

score : int

GameEntry(n : String,s : int)

getName() : String

getScore() : int

toString() : String

