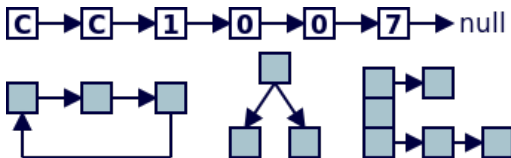



Estruturas de Dados

Pedro Ribeiro

DCC/FCUP

2020/2021



- Site: <http://www.dcc.fc.up.pt/~pribeiro/aulas/edados2021/>
- Slack: <http://edados2021.slack.com>  slack
(usado para a comunicação nesta UC)
- Aulas **teóricas** com E@D (e vídeos pré-gravados):
 - ▶ 3^a Feira: 17:00-18:00
 - ▶ 6^a Feira: 15:00-16:00
- Aulas **práticas** (9 diferentes TPs com 4 diferentes docentes):
 - ▶ João Resende
 - ▶ Luis Lopes
 - ▶ Patrícia Sousa
 - ▶ Ricardo Rocha

Para obter frequência é necessário:

- Não exceder limite de faltas nos **quizzes** online (máximo de 6 faltas)
 - ▶ Estão previstos 12 quizzes
 - ▶ São obrigatórios, mas não contam para nota
 - ▶ Cada um estará online para efeitos de frequência durante uma semana (00:01 de Domingo a 23:50 de Sábado)
(depois ficam disponíveis para treino, sem contar para frequência)
 - ▶ São constituídos por perguntas de escolha múltipla
 - ▶ Podem ver os resultados, saber que opção estava correcta e voltar a submeter quantas vezes quiserem

Fórmula de Cálculo da Avaliação

- **P:** nota prática, valendo **30%** da nota final, obtida através de 3 componentes:

- ▶ 2 testes práticos de programação (2.5 valores cada)
- ▶ resolução de exercícios ao longo do semestre (1 valor).

Nota mínima: $P \geq 1.5$ (escala da nota: 0 a 6).

- **EN:** nota do exame de época normal, valendo 70% da nota final, obtida através de um exame escrito (presencial) com nota de 0 a 20
- **ER:** na época de recurso será feito um único exame (presencial), valendo 70% da nota final, não sendo possível repetir a componente prática

Classificação da época normal: $C = EN * 0.7 + P \geq 9.5$

Classificação da época de recurso: $C = ER * 0.7 + P \geq 9.5$

Sobre a componente prática

- Será usada a linguagem **Java**, e terão acesso a um computador
- Terão objectivos específicos divulgados para os testes
- Poderão vir a ter acesso a código meu e/ou vosso (pré-submetido)
(compensa por isso terem feito os exercícios antes, perceberem bem o que fizeram e terem código organizado)

Datas dos testes práticos de programação (duração: 2h):

- **1º teste:** previsto ser em abril
- **2º teste:** previsto ser no final do semestre

Estruturas de Dados nos últimos 5 anos:

- Total de Alunos:
254 (15/16), **272** (16/18), **285** (17/18), **259** (18/19), **236** (19/20)
- Alunos aprovados:
74 (15/16), **76** (16/17), **133** (17/18), **135** (18/19), **132** (19/20)

Distribuição de notas no ano passado:



Estruturas de Dados 2020/2021

- Total de Alunos inscritos: **244**
- Nº de inscrições: **170** (1^a), **42** (2^a), **13** (3^a)
9 (4^a), **7** (5^a), **2** (6^a), **1** (7^a)
- Total por curso: **134** (MI:ERS), **100** (L:CC)
6 (L:M), **2** (L:F), **1** (L:B)
1 (*aluno em mobilidade*)

Objectivos da Unidade Curricular

Competências de **programação**

- Proeficiência na linguagem **Java**
- Paradigma de **Programação Orientada a Objectos**

Conhecimentos de **estruturas de dados e algoritmos** associados

- Principais estruturas de dados básicas: **arrays, matrizes, listas ligadas e árvores binárias.**
- Principais tipos abstractos de dados e suas implementações: **filas, pilhas, conjuntos, dicionários e filas de prioridade.**

Competências básicas na concepção e **análise de algoritmos**

- Competência básica na análise de **complexidade de algoritmos** e e compreensão das principais **classes de complexidade.**
- Enriquecimento do conhecimento sobre técnicas algorítmicas como **recursividade, pesquisa com retrocesso e dividir para conquistar.**
- Experiência prática de aplicação a **problemas concretos.**

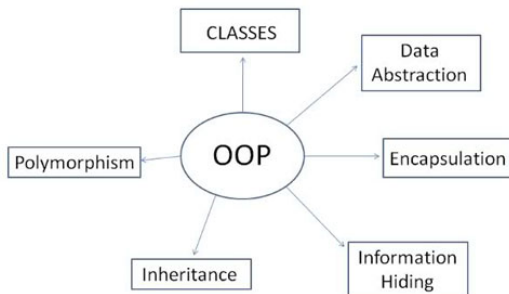
Conceitos fundamentais de Java:

- Classes, objectos, atributos e métodos
- Tipos primitivos, strings, wrappers, arrays e tipos enumerados
- Expressões, operadores e instruções de controle de fluxo
- Input/Output e a classe Scanner
- Pacotes e biblioteca padrão do Java
- Princípios de desenvolvimento de software, estilo e documentação



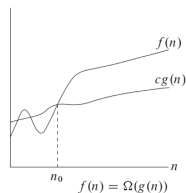
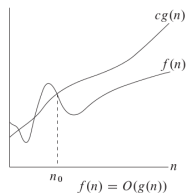
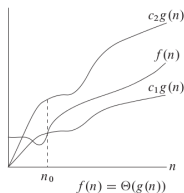
Programação Orientada a Objectos

- Objectivos, princípio, padrões e mecanismo de herança
- Interfaces e Tipos Abstractos de Dados (TADs)
- Uso de genéricos e de iteradores



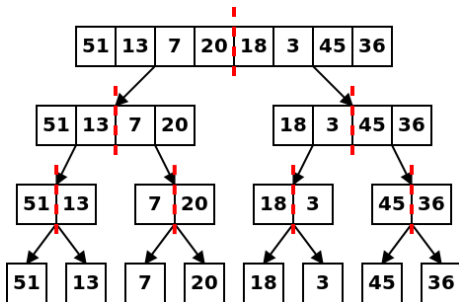
Conceitos de Análise Assintótica

- Noções de análise assintótica
- Classes de complexidades típicas e sua comparação
- Exemplos de análise de algoritmos



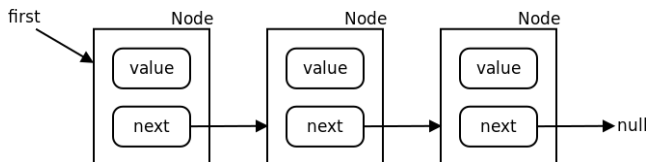
Técnicas de Desenho de Algoritmos

- Programação estruturada
- Recursividade
- Pesquisa exaustiva e *backtracking*
- Dividir para conquistar



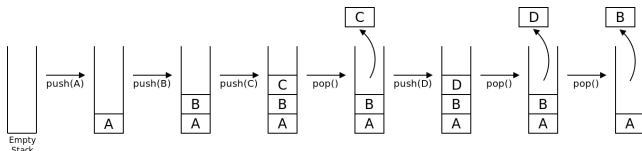
Estruturas de Dados Fundamentais

- Arrays e matrizes
- Listas ligadas simples, circulares e duplamente ligadas
- Árvores binárias, árvores de pesquisa e heaps



Tipos Abstratos de Dados e suas possíveis Implementações

- Sequências, pilhas, filas e dequeus
- Contentores associativos: conjuntos e dicionários
- Filas de prioridade



Funcionamento das aulas

- **Teóricas:** vídeos (pré-gravados) com uso de: slides + livecoding + quadro + visualizações + ... (+ quizzes no Mooshak)
- **Práticas:** guião com exercícios (html) + implementação em código (Java) (+ submissões no Mooshak)
- **Material auxiliar:** divulgado no site, em português e/ou inglês videos, slides, apontamentos, animações, ...

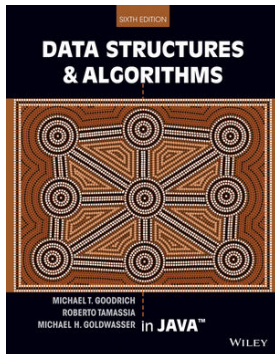
Importante trabalhar fora das aulas!

As aulas são "apenas" as **horas de contacto**. É importante estar presente mas não chega apenas lembrar-se de E. Dados durante as aula práticas....

Livro Principal:

- **Data Structures and Algorithms in Java**

M Goodrich, R Tamassia and M Goldwasser, 6th Edition, Wiley, 2014.



Outros Livros Recomendados:

- **Introduction to Algorithms**

TH Cormen, CE Leiserson, RL Rivest and C Stein, 3rd Edition, MITPress, 2009.

- **Introduction to Programming in Java: An Interdisciplinary Approach**

R. Sedgewick, K Wayne, 2nd Edition, Addison-Wesley, 2017.

- **Think Java: How To Think Like a Computer Scientist**

A Downey, O'Reilly, 2016.

- **Think Data Structures**

A Downey, O'Reilly, 2017.