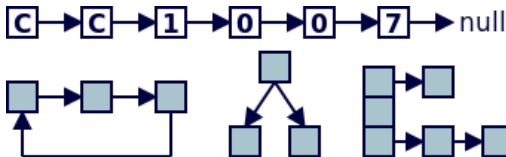


Estruturas de Dados

Pedro Ribeiro

DCC/FCUP

2019/2020



- Site: <http://www.dcc.fc.up.pt/~pribeiro/aulas/edados1920/>
- Piazza: <http://piazza.com/up.pt/spring2020/cc1007>
(usado para a comunicação nesta UC)
- Aulas **teóricas**:
 - ▶ 3ª Feira: T1 (11:00-12:00, FC1027) | T2 (12:00-13:00, FC1003)
 - ▶ 6ª Feira: T1 (11:00-12:00, FC1007) | T2 (12:00-13:00, FC1003)
- Aulas **práticas** (9 diferentes TPs com 5 diferentes docentes):
 - ▶ João Barbosa
 - ▶ João Resende
 - ▶ João Vinagre
 - ▶ Patrícia Sousa
 - ▶ Pedro Ribeiro
- **Atendimento**:
 - ▶ 6ª feira: 14:00 às 15:00 (gabinete 1.47)
 - ▶ Almoço EDados 6ª: a seguir às aulas teóricas

Obtenção de Frequência

Para obter frequência é necessário:

- Não exceder limite de faltas nas **aulas práticas** (máximo de 4 faltas)
 - ▶ Estão previstas 13 aulas práticas (feriados não contam para faltas)
- Não exceder limite de faltas nos **quizzes** online (máximo de 4 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 testes práticos de programação (2 valores cada)Nota mínima: 25% (minimo de 1.5 numa nota de 0 a 6 valores)
($< 25\%$ na parte prática \rightarrow reprovação por falta de componente)
- **E:** nota de exame, valendo **70%** da nota final
(teste escrito com nota de 0 a 20)
- **R:** na época de recurso será feito um único exame
(teste com nota de 0 a 20, não é possível repetir testes práticos)

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

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

Sobre a componente prática

- Será usada a linguagem **Java**, e terão acesso a um computador
- Terão objectivos específicos divulgados
- 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 (4ª feiras, duração: 2h):

- **1º teste:** 18 de Março
- **2º teste:** 22 de Abril
- **3º teste:** 20 de Maio (*data poderá mudar*)

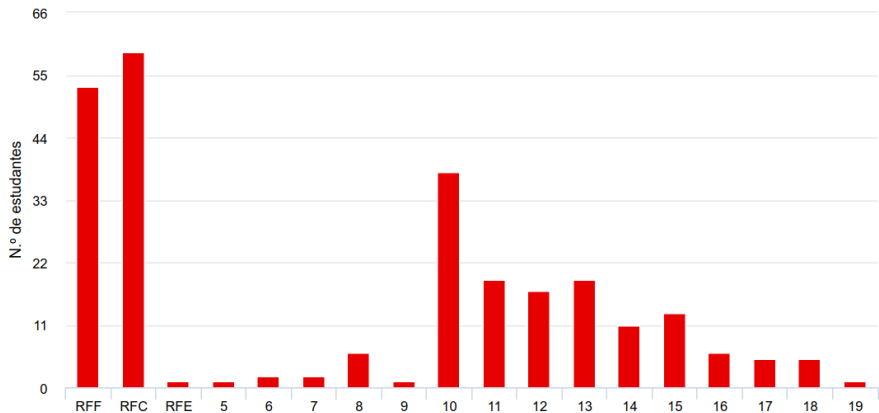
Estruturas de Dados nos últimos 5 anos:

- Total de Alunos:
255 (14/15), **254** (15/16), **272** (16/18), **285** (17/18), **259** (18/19)
- Alunos aprovados:
75 (14/15), **74** (15/16), **76** (16/17), **133** (17/18), **135** (18/19)

Distribuição de notas no ano passado:

Histograma dos resultados finais

Distribuição em 2018/19



Estruturas de Dados 2019/2020

- Total de Alunos inscritos: **241**
- Nº de inscrições: **157** (1ª), **43** (2ª), **22** (3ª), **10** (4ª), **6** (5ª), **3** (6ª)
- Total por curso: **133** (MI:ERS), **90** (L:CC)
10 (L:M), **5** (L:F), **1** (L:B), **1** (L:BQ)
1 (*sem curso definido*)

Programação Imperativa 2019/2020 (*dados apenas da época normal?*)

- Total de Alunos: **200**
- Total de Avaliados: **142**
- Total de Aprovados: **68**

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.**

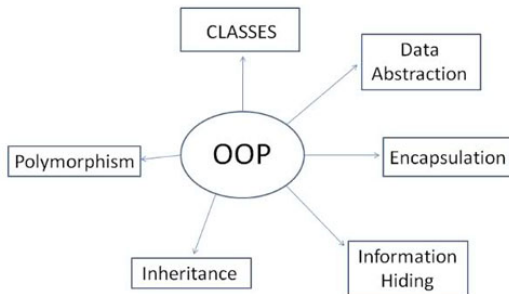
Conceitos fundamentais de Java:

- 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



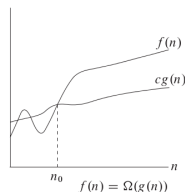
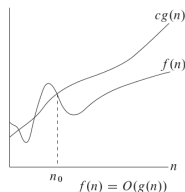
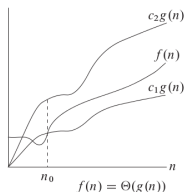
Programação Orientada a Objectos

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



Conceitos de Análise Assintótica

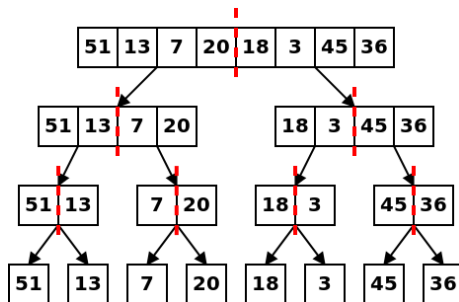
- Noções de análise assintótica
- Classes de complexidades típicas e sua comparação
- Exemplos de análise de algoritmos



Visão Geral do Programa

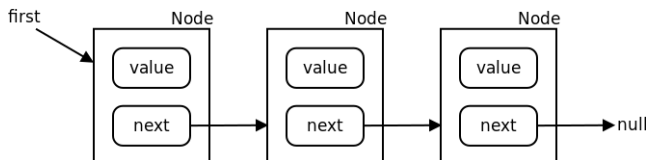
Técnicas de Desenho de Algoritmos

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



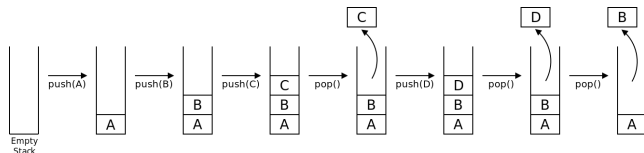
Estruturas de Dados Fundamentais

- 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

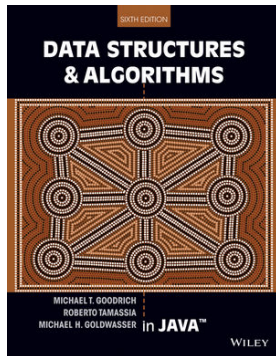
- **Teóricas:** slides +
uso do computador do docente +
exposição no quadro
(+ quizzes no Mooshak)
- **Práticas:** guião com exercícios (html) +
implementação em código (Java)
(+ submissões no Mooshak)
- **Material auxiliar:** divulgado no site, em português e/ou inglês
slides, apontamentos, animações, applets, ...

Importante trabalhar fora das aulas!

As aulas são "apenas" as **horas de contacto**. É importante vir 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.