

ALGORITMOS E ESTRUTURAS DE DADOS

2018/2019

APRESENTAÇÃO

Armanda Rodrigues

11 de setembro 2018

Até aqui...

- Introdução à Programação
 - Primeira disciplina de programação
 - Introdução ao Java como primeira linguagem, com base no conceito de Objecto
- Programação Orientada aos Objectos
 - Desenvolvimento dos conceitos subjacentes à Programação orientada a Objectos
 - Herança, Polimorfismo, Interfaces e Classes, etc...
 - Desenvolvimento dos vossos conhecimentos de Java
 - Bibliotecas Java

Neste momento são capazes de desenvolver programas que funcionam!

Algoritmos e Estruturas de Dados

- Pôr algo a funcionar não chega!
 - O objectivo fundamental desta disciplina é que se tornem capazes de melhorar o vosso código melhorando a sua qualidade
- Como vamos conseguir isto ?
 - Tipos Abstratos de Dados
 - Estruturas de Dados
 - Algoritmos
 - Estudo da complexidade de algoritmos
- No final do semestre deverão ser capazes de escolher (e implementar) as melhores opções para resolver problemas que vos são apresentados

Programa (1)

- Tipos Abstratos de Dados
 - Pilha (disciplina LIFO)
 - Fila (disciplina FIFO)
 - Lista (acesso por posição)
 - Dicionário (acesso por chave): não ordenado e ordenado
 - Fila com prioridade
- Estruturas de Dados
 - Vetores circulares
 - Listas simplesmente e duplamente ligadas
 - Tabelas de dispersão
 - Árvores: genéricas; binárias; binárias de pesquisa (sem restrições; AVL)
 - Heaps

Programa (2)

- Introdução à Recursividade
 - Técnicas associadas à recursividade:
 - Divisão e conquista.
 - Função-memória.
- Introdução à Análise de Algoritmos
 - Complexidade temporal
 - Complexidade espacial
- Algoritmos de Ordenação
 - Ingénuos: (e.g. insertion sort; selection sort; bubble sort)
 - Eficientes: (e.g. heapsort; mergesort; quicksort)
 - Por indexação: (e.g. bucket sort; radix sort)

Bibliografia

- Referências Principais
 - Michael T. Goodrich, Roberto Tamassia and Michael H. Goldwasser. Data Structures and Algorithms in Java (6th edition). John Wiley & Sons, 2015.
 - Mark Allen Weiss. Data Structures and Algorithm Analysis in Java (3rd edition). Addison-Wesley, 2012.
- Outras referências
 - Mark Allen Weiss. Data Structures and Problem Solving Using Java (4th edition). Addison-Wesley, 2013.
 - Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. Introduction to Algorithms (3rd edition). The MIT Press, 2009.

Regras de Avaliação

- Qualquer aluno envolvido numa fraude (detetada imediatamente ou a posteriori, no trabalho ou no exame) reprova na disciplina.
- A avaliação é composta por:
 - Um trabalho, entregue em três fases obrigatórias;
 - Dois testes.
- O trabalho é obrigatório para quem não tem frequência e dá frequência
- Avaliação
 - Nota do trabalho prático (NP) – 25%
 - Fase 1 – 5%; Fase 2 – 5%; Fase 3 – 15%
 - Nota dos Testes (NT) – 75%
 - Teste 1 – 20%; Teste 2 – 55%.

Frequências de edições anteriores

- Tal como definido no regulamento de avaliação de FCT-NOVA, **a frequência tem validade por um ano** (regras de 2017/18 aplicam-se)
- Alunos com frequência anterior a 2017/18 podem ser dispensados de obtenção da mesma, **embora a nota do trabalho não seja considerada na nota final.**
- Para estes alunos:
 - Em caso de aprovação por testes, o primeiro teste vale 27% da nota final e o segundo teste, 73%;
 - Em caso de aprovação por exame, a nota final será a nota de exame.
- Os alunos que pretendam abdicar da dispensa de frequência, devem comunicá-lo ao docente do turno prático e **realizar todas as fases do trabalho prático**
- **O processo torna-se irreversível a partir da entrega de uma das fases do trabalho**

Trabalho

- O trabalho consiste:
 - no desenho, análise e implementação de uma aplicação de pequena dimensão;
 - na elaboração de um relatório; e
 - na realização de uma discussão.
- É realizado individualmente ou em grupo de dois alunos
- O trabalho é realizado em 3 fases obrigatórias, entregues em datas marcadas
- A nota global do trabalho (NP) terá de ser maior ou igual a 10 para a obtenção da frequência
- A autoria do trabalho deverá ser comprovada em discussão presencial.

Trabalho - Versões

- Cada estrutura de dados necessária para o desenvolvimento do trabalho, será aceita numa de duas versões:
 - Versão I, implementada de raiz
 - Versão J, utilizada a biblioteca `java.util`
- No limite, se os alunos não usarem a biblioteca `java.util`, entregarão uma **Versão I completa**, avaliada entre 10 e 20 valores, onde não farão uso do *package* `java.util`, à exceção da classe `Scanner`
- A **Versão J completa**, avaliada entre 10 e 15 valores, poderá fazer uso de todas as interfaces e classes do Java
- **Possível uma versão mista:** A avaliação do trabalho terá em conta o número de estruturas de dados que fazem importação de classes contidas na biblioteca `java.util`

Trabalho - Importante

- Não é permitido usar qualquer tipo de código não desenvolvido expressamente pelos alunos do grupo, com exceção do código disponibilizado na página Moodle da disciplina
 - **Dica:** Para iniciar o desenvolvimento do trabalho deve criar-se um projeto vazio no Eclipse, inserindo, conforme as necessidades, os TADs e classes disponíveis na página da disciplina
- Quaisquer classes utilizadas no trabalho que não tenham sido desenvolvidas de raiz pelos alunos, deverão ser devidamente referidas no código e no relatório final e poderão condicionar a nota final.
- A cópia de código será verificada utilizando o software MOSS

Faseamento do Trabalho

- O trabalho será desenvolvido incrementalmente, em 3 fases obrigatórias com entregas em datas marcadas
- Em cada fase, todas as funcionalidades deverão ser implementadas, mas poderão crescer de complexidade
- Uma vez que o valor do trabalho é de 25% da nota final, esta nota será dividida da seguinte forma, nas fases do trabalho
 - 1ª fase – 5%
 - 2ª fase – 5%
 - 3ª fase – 15%
- Se uma das fases não for entregue no prazo definido, a nota associada à mesma fase será 0 (zero)

Testes do trabalho e discussão

- O trabalho terá de satisfazer todos os requisitos especificados no enunciado
- A verificação será efetuada pelo sistema Mooshak, através de testes de avaliação
- Em cada fase:
 - **ATENÇÃO:** Se o programa não passar **todos** os Testes de Avaliação, os seus autores obterão a nota de 0 (zero) na fase em questão;
 - Os programas que passarem os Testes de Avaliação da fase terão uma avaliação preliminar, que deverá ser confirmada no final do semestre, numa discussão final.
- No final do semestre, todos os grupos em posição de obter frequência serão submetidos a uma discussão do trabalho, para verificação de autoria

Discussão final

- **Objetivo**
 - Verificar autoria do trabalho por parte dos alunos do grupo
 - Verificar que o trabalho foi desenvolvido pelos alunos
 - Verificar que todos os alunos do grupo contribuíram **de forma equilibrada**
- **Obrigatoriamente:**
 - Os alunos são responsáveis por **todo o código** submetido ao Mooshak, têm se saber como este funciona
 - Os alunos têm de conhecer o código submetido em qualquer das fases do trabalho (têm de preparar a discussão nesse contexto)
- **NÃO VALE DIZER QUE JÁ NÃO SE LEMBRAM DA 1ª FASE**

Datas importantes para o trabalho

- **Publicação do Enunciado:** 24 de Setembro de 2018
- **Submissão da primeira fase do trabalho, no Mooshak:** entre 8 e 12 de Outubro de 2018 (até às 17h).
- **Submissão da segunda fase do trabalho, no Mooshak:** entre 5 e 9 de Novembro de 2018 (até às 17h).
- **Submissão da terceira e última fase do trabalho e do relatório final, no Mooshak:** entre 26 e 30 de Novembro de 2018 (até às 17h).
- **Publicação das marcações das discussões finais:** 3 de Dezembro de 2018.

- **Realização das discussões:** entre 4 e 7 de Dezembro, em princípio, dentro do horário da disciplina (teórico e prático).

Testes

- Os testes são sem consulta, escritos e individuais. Os alunos devem levar um caderno de teste (de AED) e um documento de identificação com fotografia
- A nota **combinada** dos testes e arredondada à unidade deverá ser superior ou igual a 10 em 20, independentemente do peso dos testes na nota final.
 - Quem não cumprir os requisitos dos testes irá a exame de recurso, desde que já tenha obtido frequência ou tenha sido dispensado da mesma
- Primeiro teste – dia 29/10/2018 18h
 - Inscrição obrigatória no CLIP entre 30/09 e 14/10
- Segundo teste – dia 13/12/2018 18h
 - Inscrição obrigatória no CLIP entre 14/11 e 28/11

**DATAS PROPOSTAS
PELA CCMIEI!**

Exame

- O exame é escrito e individual.
- O exame é obrigatório para todos os alunos que não tenham obtido o requisito de testes (mas só pode ser realizado por alunos que tenham obtido frequência)
- O exame é avaliado de 0 a 20 valores, com arredondamento à unidade. Para obter aprovação, a nota mínima é 10 valores.
- Para o exame, os alunos devem levar um caderno de exame (de AED) e um documento de identificação com fotografia

Resumo - Frequências de 2018/19 ou 2017/18 (I)

- O aluno obtém aprovação se (e só se):
 - Tiver obtido frequência em 2017/18 ou 2018/19;
 - $NP = (Fase1*0,05 + Fase2*0,05 + Fase3*0,15) / 0,25$
 - NP arredondada à unidade tem de ser ≥ 10
 - Tiver nota positiva na componente teórico-prática (NT)
 - $NT = (Teste1*0,2 + Teste2*0,55) / 0,75$
 - NT arredondada à unidade tem de ser ≥ 10
 - A nota final (NF) for positiva:
 - $NF = NP*0,25 + NT*0,75$
 - NF arredondada à unidade tem de ser ≥ 10
- Em época de recurso, a nota dos dois testes é substituída pela nota do exame, que vale 75% da nota final.

Resumo - Frequências anteriores a 2017/18 (II)

- O aluno obtém aprovação se :
 - Tiver obtido frequência antes de 2017/18;
 - Não tiver entregue qualquer fase do trabalho em 2018/19
 - Tiver nota positiva na componente teórico-prática (NT)
 - $NT = (\text{Teste1} * 0,27 + \text{Teste2} * 0,73)$
 - NT arredondada à unidade tem de ser ≥ 10
 - A nota final (NF):
 - $NF = NT$
- Em época de recurso, a nota dos dois testes é substituída pela nota do exame, que vale 100% da nota final.

8 Turnos práticos

Sofia Cavaco (scavaco@fct.unl.pt)
Monitores (a definir)

P3, P4, P5, P6 e P7
P1, P2, P8

Página de AED no Moodle

<https://moodle.fct.unl.pt/course/view.php?name=AED1819>
(password ALG1819)