

Programação Funcional (cod.14348 & 14786)

Departamento de Informática
Universidade da Beira Interior

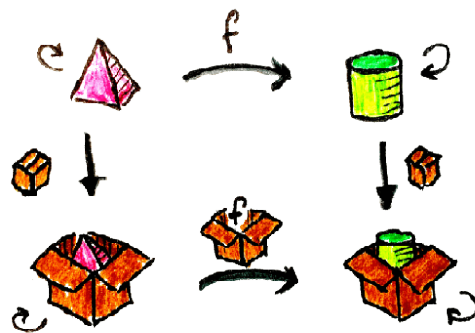


Figura 1: from the science behind functional programming.

To appreciate programming as an intellectual activity... you must read and write computer programs, many of them. It doesn't matter much what the programs are about or what applications they serve. What does matter is how well they perform and how smoothly they fit with other programs in the creation of still greater programs. The programmer must seek both perfection of part and adequacy of collection. [Alan J. Perlis]

Esta página no formato pdf

1 Programação funcional, programação OCaml

Para quê aprender à programação funcional? qual é o papel de uma unidade curricular como esta? Porque

o programador moderno, exímio e acutilante, domina a programação funcional (tipificada).

Uma reflexão interna mais detalhada poderá ser encontrada na aula 0 desta UC, ou na página do mini-curso “Introdução à Programação Funcional, em OCaml”.

Junta-se as referências externas seguintes

Why OCaml at JaneStreet?, Why functional programming matters? OCaml for the masses, OCaml at Cornell, OCaml for the Skeptical, OCaml, from its site, Real World OCaml - prologue, Why did Facebook pick OCaml?, OCaml at Princeton, OCaml at Stanford, OCaml at Cambridge, OCaml at Harvard, OCaml at MIT.

2 Novidades

- Encontrará aqui as informações iniciais associadas à disciplina de **Programação Funcional**. As notícias e informações em tempo de aulas serão dadas no **Canal da UC Programação Funcional no Microsoft Teams**.
- Avaliação Prática: [Mooshak](#) - processo de registo (link aqui).

O sistema mooshak encontra-se desde já configurado para a presente disciplina. Queira proceder ao seu registo.

Aceita-se a formação de grupos de, no máximo, duas pessoas. No processo de registo, escolhe o grupo UBI e defina o nome da equipa da seguinte forma.

Se for um grupo de um só elemento: numero de aluno + primeiro nome. Por exemplo, Luís com o numero 12345 tem por registo mooshak "12345Luis".

Se for um grupo de duas pessoas: numero de aluno do primeiro elemento (o de número mais baixo) + primeiro nome+numero de aluno do segundo elemento + primeiro nome. Assim se Luís forma grupo com o João (aluno 13245), então o grupo regista-se com o nome “12345Luis13245Joao”.

- Consultar a secção 9 para aceder aos material electrónico (teórico e prático) exposto nas aulas.
- Inscrição em turmas práticas: via site dos serviços académicos.
- As aulas práticas começam logo na primeira semana de aulas.
- Os alunos com estatuto de *trabalhador estudante* são convidados a dirigir-se ao regente para discutir eventuais alterações dos critérios de avaliação.

Conteúdo

1	Programação funcional, programação OCaml	2
2	Novidades	2
3	Docentes	4
4	Objectivos	4
5	Programa	5
6	Critérios de Avaliação	6
6.1	Actividades de Ensino-Aprendizagem e Metodologias Pedagógicas	6
6.2	Componente Prática	7
6.3	Componente Teórica	7
6.4	Concessão de Frequência e Avaliação Contínua	7
6.5	Avaliação por Exame	8
7	Datas Importantes	8

8	Atendimento	9
9	Material Pedagógico e Funcionamento da Disciplina	9
10	Resultados da avaliação	11
11	Bibliografia	11

3 Docentes

- Simão Melo de Sousa (regente) - Gabinete 3.17 - Laboratório Release/QuiVVer/NOVA-LINCS
- João Reis
- Vasco Ferrinho Lopes
- Tiago Roxo

Como colocar uma dúvida à equipa docente da Unidade Curricular?

1. Tirar proveito do Grupo Teams da UC e colocar a dúvida no canal adequado - método preferido!
2. Comparecer nas aulas e colocá-la directamente ao docente.
3. Comparecer no horário de atendimento do regente e colocá-la directamente.
4. enviar um email ao regente (desousaUUU@UUUdi.ubi.pt, (retire os UUU - medida anti-spam)) com o assunto "PF: XXXX" em que XXX é o título da dúvida em questão. Qualquer outro formato no assunto arrisca condenar o email ao esquecimento.

4 Objectivos

Os objectivos gerais de aprendizagem são os seguintes:

- Perceber os fundamentos de programação funcional para resolver problemas de natureza computacional.

- Compreender as diferenças entre os paradigmas de programação imperativa e funcional.
- Introduzir os conceitos básicos de programação funcional.
- Desenvolver capacidades de programação com recurso à uma linguagem funcional.
- Estudar algoritmos sobre estruturas de dados como listas e árvores.
- Estudar algoritmos de procura e de ordenação.
- Estudar técnicas algorítmicas como, e.g. a programação dinâmica ou ainda backtracking.

No final da UC, o aluno deverá ser capaz de (objectivos específicos de aprendizagem):

- Definir funções usando equações com padrões.
- Codificar algoritmos recursivos elementares sobre estruturas de dados fundamentais (e.g. listas e árvores, etc.).
- Definir novos tipos algébricos para representar dados.
- Decompor problemas de programação usando os mecanismos próprios da programação funcional.
- Saber desenhar uma solução programática que envolva estrutura de dados (sequências ou arborescentes) e algoritmos básicos (ordenação ou pesquisa) e para um problema computacional.
- Saber usar técnicas algorítmicas como dividir-e-conquistar, a programação dinâmica, algoritmia gulosa ou ainda programação com retrocesso.

5 Programa

- Tipos Básicos, Entrada/Saída, Estruturas de Controlo e Funções, Recursividade e Funções de Ordem Superior.
- Polimorfismo, Tipos de dados algébricos: tipos produtos, enumerados, soma e estruturados.

- Módulos e Estruturas de dados abstractas.
- Noções Gerais de análise de programas: complexidade computacional e correcção funcional.
- Estruturas sequenciais: vectores (redimensionáveis), tabelas de Hash, listas, pilhas e filas.
- Estruturas não sequenciais : conjuntos, dicionários, árvores, cordas, amontoados, árvores binária equilibradas.
- Ordenação, indexação e pesquisa.
- Algoritmia: dividir-e-conquistar, algoritmos gulosos, por retrocesso, programação dinâmica, memoização.

6 Critérios de Avaliação

6.1 Actividades de Ensino-Aprendizagem e Metodologias Pedagógicas

Por fim a avaliar as competências adquiridas, as actividades de Ensino-Aprendizagem avaliarão tanto a compreensão dos conceitos teóricos expostos como a capacidade em por estes em prática.

Assim, a avaliação será constituída por uma componente **teórica** e por uma componente **prática**.

Fraudes A equipa docente realça que qualquer tipo de fraude em qualquer dos itens desta disciplina implica a reprovação automática do aluno faltoso, podendo ainda vir a ser este alvo de processo disciplinar.

Consulte, para esse efeito o

código de integridade em vigor na UBI.
--

Listamos a seguir as diferentes componentes da avaliação.

6.2 Componente Prática

- Esta avaliação mede em termos práticos a aquisição dos conceitos expostos. Como tal é baseada na avaliação da resolução de exercícios práticos de programação. Esta avaliação tomará a forma da resolução de desafios de programação dos conceitos introduzidos nas aulas.
- Os exercícios avaliados são em número de 3 e resolvidos de uma forma sequencial e em grupo de um ou dois alunos. As datas exactas de entrega encontram-se na secção 7. A entrega é feita de forma electrónica no site mooshak da UC.
- Só serão consideradas para avaliação as submissões aceites/"Accepted" mais recentes devidamente comentadas conforme indicação do regente da UC. Qualquer outro tipo de submissão (não aceite, não conforme aos critérios de submissão) será considerada nula.
- Após o prazo de entrega, cada problema aceite será alvo de uma defesa pelo grupo que o entregou, na aula prática (do primeiro elemento do grupo) que segue a data de entrega.
- A *Nota da Componente Prática* (NCP, 20 valores) é resultante da avaliação atribuída ao trabalho prático.

6.3 Componente Teórica

A avaliação da componente teórica consiste numa frequência (ver secção 7 para conhecer a data prevista da frequência).

Da avaliação desta prova resulta a *Nota da Componente Teórica* (NCT, 20 valores).

6.4 Concessão de Frequência e Avaliação Contínua

O parâmetro de "Frequência" atribuído no final desta unidade curricular traduz, no contexto da avaliação contínua, a "aquisição de conhecimento mínima" do estudante ao longo do processo de ensino-aprendizagem no final das actividades de contacto.

Considera-se que o estudante demonstrou ter adquirido um grau de conhecimento mínimo (durante o processo de aprendizagem ao longo das actividades lectivas) quando este entregou a prova escrita da frequência e entregou

com sucesso pelo menos um dos exercícios da avaliação prática (i.e. submissão com “Accepted” validada pela equipa docente por meio de uma defesa).

É assim concedido **Frequência** ao aluno nesta situação.

Avaliação contínua qualitativa No caso de “Frequência”, a avaliação quantitativa, designada aqui de Nota da Avaliação Contínua (NAC), é determinada da seguinte forma:

$$NAC = \frac{\text{componente prática (NCP)} \times 0.8 + \text{componente teórica (NCT)} \times 1.2}{2}$$

Qualquer manifestação de fraude implica a não admissão imediata a exame!

6.5 Avaliação por Exame

A nota da prova escrita do exame substituirá a nota da componente teórica da Avaliação Contínua. Em consequência, a Nota da Avaliação por Exame segue o mesmo cálculo que a Nota da Avaliação Contínua.

Da mesma forma, as avaliações acima de 16 serão alvo de uma defesa de nota.

7 Datas Importantes

- Frequência: Terça-feira 30 de Maio de 2023 das 11h00 às 13h00.
- Componente prática:
 - Exercício 1 : Entrega semana do 28 de Março de 2023 (defesas, semana seguinte).
 - Exercício 2 : Entrega semana do 25 de Abril de 2023 (defesas, semana seguinte).
 - Exercício 3 : Entrega semana do 30 de Maio de 2023 (defesas, semana seguinte).
- Exame : (conferir no site dos académicos).

8 Atendimento

Preferencialmente *Teams*. Em alternativa, por email ou

Horário	Docente
segunda-feira das 16h00 às 18h00	S. Melo de Sousa

9 Material Pedagógico e Funcionamento da Disciplina

Os Apontamentos serão atempadamente disponibilizados nas aulas e por meios electrónicos. É esperado e assumido que o aluno tenha lido os acetatos referentes ao capítulo em curso antes das aulas teóricas.

Teóricas

Functional programming combines the flexibility and power of abstract mathematics with the intuitive clarity of abstract mathematics. (visto aqui)

(legenda : link com problemas = aula por disponibilizar)

Contextualização da UC

Aula inaugural : Aula 0.

Programação Funcional e OCaml : Aula 1.

Introdução à programação funcional em OCaml

OCaml : Aula 2.

OCaml : Aula 3.

OCaml : Aula 4.

Introdução à análise de algoritmos: complexidade algorítmica e correcção funcional

Complexidade algorítmica : Aula 5.

Pre/pos condições, invariantes e programação por contratos : Aula 6.

Base de algoritmos e de estruturas de dados

Estruturas de dados sequenciais : Aula 7.

Estruturas de dados não sequenciais (parte 1) : Aula 8.

Estruturas de dados não sequenciais (parte 2): Aula 9.

Ordenação, indexação e pesquisa : Aula 10.

Algoritmos gulosos, por retrocesso : Aula 11.

Programação dinâmica, memoização : Aula 12.

Práticas

Utilizaremos a plataforma Learn-OCaml configurada localmente para esta Unidade Curricular

[consultar o site \(link\)](#)

Para efeitos de consulta, deixamos aqui a versão html das fichas presentes na plataforma learn-ocaml, com a anotação dos exercícios por resolver):

- Ficha 0 (*)
- Ficha 1 (*)
- Ficha 2 (*)
- Ficha 3 (**)
- Ficha 4 (*)
- Ficha 5 (**)
- Ficha 6 (**)
- Ficha 7 (*)
- (mais fichas à caminho)

(*) - resolver todos os exercicios. (**) - resolver por nível:

- alunos do primeiro ano: 3 do nível básico, 2 do nível intermédio

- alunos do segundo ano: 1-2 do nível básico, 2-3 do nível intermédio, 1 do nível consolidado
- alunos do terceiros ano: 2-3 do nível intermédio, 2-3 do nível consolidado

Trabalhos Práticos

- o Problema A
 - o Problema B
 - o Problema C

10 Resultados da avaliação

Por definir

11 Bibliografia

As referencias principais são:

- Jean-Christophe Filiâtre, Sylvain Conchon, com tradução de Simão Melo de Sousa. “Aprender a programar com OCaml : algoritmos e estruturas de dados”. 1ra Edição, 2021. (Obra original : “Apprendre à programmer avec Ocaml : algorithmes et structures de données”. Eyrolle, 2014. ISBN-13: 978-2212136784. book website).

Utilizaremos ocasionalmente as referências:

- Emmanuel Chailloux, Pascal Manoury, and Bruno Pagano. Developing Applications With Objective Caml. English translation of “Développement d’applications avec Objective Caml”. O’Reilly, Paris, 2000, ISBN 2-84177-121-0. Online Version.
- Yaron Minsky, Anil Madhavapeddy, Jason Hickey. Real World OCaml, Functional Programming for the masses. 2nd Edition. O’Reilly. Versão online.

- Robert Sedgewick, Kevin Wayne. Algorithms. Addison-Wesley Professional; 4th edition (April 3, 2011). book site - with extra material, online courses and videos.
- Jon Kleinberg, Éva Tardos. Algorithm Design. Pearson; 1st edition (March 16, 2005).
- Thomas H. Cormen, Charles Eric Leiserson, Ronald Rivest, Ronald L. Rivest E Clifford Stein. Introduction to Algorithms. MIT Press; 3rd edition (September 1, 2009).
- Steven S. Skiena. The Algorithm Design Manual. Texts in Computer Science. Springer; 3rd edition (October 6, 2020).
- Tim Roughgarden. Algorithms Illuminated.
 - Part 1, The Basics. Soundlikeyourself Publishing; Illustrated edition (September 27, 2017)
 - Part 2, Graph Algorithms and Data Structures. Soundlikeyourself Publishing, LLC; First edition (August 5, 2018)
 - Part 3, Greedy Algorithms and Dynamic Programming. Soundlikeyourself Publishing, LLC; Illustrated edition (May 1, 2019).
 - Material adicional - site web dos livros (contém recursos suplementares e videos das aulas, curso online coursera, curso EDX1, curso EDX2)
- Jeff Erickson. Algorithms. Independently published; 1st edition (June 2019).
- Guy Cousineau and Michel Mauny. The Functional Approach to Programming with Caml. Cambridge University Press, 1998, ISBN 0-521-57183-9 (hardcover), 0-521-57681-4 (paperback)
- Andrei de Araújo Formiga. OCaml Programação funcional na prática. Casa do Código. ISBN: 978-85-5519-070-4 site.
- John Whittington. OCaml from the very beginning. Coherent Press 2013.

- Steven Halim, Felix Halim and Suhendry Effendy. Competitive Programming 4 - The Lower Bound of Programming Contests in the 2020s. (with code and solutions written in OCaml)
 - Book 1. Ed. Lulu.com, 2018. (Site)
 - Book 2. Ed. Lulu.com, 2020. (Site)

Enviar comentários e dúvidas para (retire os UUU) : desousaUUU@UUUdi.ubi.pt