

Teoria da Prova e da Programação

Departamento de Informática
Universidade da Beira Interior

Ano lectivo 2015/2016

Esta página no formato pdf

1 Novidades

- Primeira versão da página. Encontrará aqui as novidades associadas à disciplina de **Teoria da Prova e da Programação**. A sua consulta regular é necessária ao bom funcionamento da Unidade Curricular.
- Como colocar uma dúvida ao regente da Unidade Curricular?
 1. Comparecer nas aulas e colocá-la directamente ao regente
 2. Comparecer no horário de atendimento do regente e colocá-la directamente
 3. enviar um email ao regente (desousaUUU@UUUdi.ubi.pt, (retire os UUU)) com o assunto "TPP: XXXX" em que XXX é o título da dúvida em questão. Qualquer outro formato no assunto arrisca condenar o email ao esquecimento.
- Inscrição em turmas práticas: via site dos serviços académicos.
- Os alunos com estatuto de *trabalhador estudante* são convidados a dirigir-se ao regente para discutir dos critérios de avaliação.

Conteúdo

1 Novidades

1

2	Docentes	2
3	Objectivos	2
4	Programa	2
5	Critérios de Avaliação	3
5.1	Componente Ensino/Aprendizagem	3
5.2	Admissão e Avaliação por Exame	3
6	Datas Importantes	4
7	Material Pedagógico e Referências Bibliográficas	4

2 Docentes

Simão Melo de Sousa (regente) - Gabinete 3.17 - Laboratório (Rel)ease (6.25)
- Bloco VI

3 Objectivos

Esta disciplina visa abordar os tópicos fundamentais da ciência da computação e da programação. A disciplina tem assim como objectivo expor os alicerces conceptuais, as técnicas e as ferramentas que permitam explicar, justificar e conceber o raciocínio por computador como também as linguagens de programação e os seus próprios programas, baseando-se neste propósito num contexto teórico abrangente e uniforme.

4 Programa

- Lógica Proposicional e Predicativa, Sistemas de dedução, Problemas de decisão
- Procedimentos de Decisão clássicos
- Sistemas deductivos e Dedução Natural
- Lógica Intuicionista e Lógica de Ordem superior

- Lambda calculo puro e com tipos
- Isomorfismo de Curry Howard
- Tipos Dependentes
- Semantica operacional
- Teoria dos domínios
- Semântica Denotacional
- Aspectos fundamentais das linguagens de programação

5 Critérios de Avaliação

A avaliação será realizada por **avaliação contínua**.

Fraudes A equipa docente gostaria de realçar que qualquer tipo de fraude em qualquer dos itens desta disciplina implica a reprovação automática do aluno faltoso, podendo ainda vir a ser alvo de processo disciplinar. Listamos a seguir as diferentes componentes da avaliação.

5.1 Componente Ensino/Aprendizagem

- A avaliação contínua mede em termos práticos a aquisição dos conceitos expostos. Como tal é baseada na realização de **exercícios** entregue à equipa docente e que concluem a exposição de cada capítulo do programa previsto. Alguns exercícios poderão ter uma parte opcional que, se realizada, poderá valorizar a nota do exercício.
- A *Nota da Componente Avaliação Contínua* (NCAC, 20 valores) é a média das notas dos exercícios.

5.2 Admissão e Avaliação por Exame

- Mínimos: são seguidas as disposições aprovadas pelo Conselho Científico da Universidade aplicadas individualmente as duas componentes de avaliação. É admitido a exame quem tiver a nota *NCAC* acima dos mínimos (i.e. $NCAC \geq 6$).

- A Nota da Prova Escrita do exame substituirá a *NCAC*.

6 Datas Importantes

- Exame Época 1 : conferir nos SA.
- Exame Época 2 : conferir nos SA.
- Exame Época Especial : conferir nos SA.

7 Material Pedagógico e Referências Bibliográficas

- Apontamentos apresentados e disponibilizados nas aulas.
- Livro de Apoio: [1] (web-site : aqui)
- Referências Complementares:
 1. José Bacelar Almeida, Maria João Frade, Jorge Sousa Pinto, Simão Melo de Sousa. Rigorous Software Development, An Introduction to Program Verification, volume 103 of Undergraduate Topics in Computer Science. Springer-Verlag, first, 307 p. 52 illus. edition, 2011.
 2. John Harrison. Handbook of Practical Logic and Automated Reasoning. Cambridge University Press, 2009.
 3. B. A. Davey and H. A. Priestley. Introduction to Lattices and Order: 2nd Ed.. Cambridge University Press, 2002.
 4. Henk P. Barendregt. Lambda calculi with types. In S. Abramsky, D. Gabbay, and T. S. E. Maibaum, editors, Handbook of Logic in Computer Science, volume 2, chapter 2, pages 117?309. Oxford University Press, 1992.
 5. H. R. Nielson and F. Nielson. Semantics with Applications. John Wiley & Sons, Chichester, 1993
 6. Interactive Theorem Proving and Program Development: Coq'Art: The Calculus of Inductive Constructions, by Yves Bertot and Pierre Castéran. Springer-Verlag, 2004.

7. Certified Programming with Dependent Types, by Adam Chlipala. A draft textbook on practical proof engineering with Coq, available from his web page (<http://adam.chlipala.net/cpdt/>)
8. H. R. Nielson, F. Nielson, and C. L. Hankin. Principles of Program Analysis. Springer-Verlag, 1999.
9. B. C. Pierce. Types and Programming Languages. MIT Press, 2002.
10. B. C. Pierce, editor. Advanced Topics in Types and Programming Languages. MIT Press, 2005.
11. G. Winskel. The Formal Semantics of Programming Languages: An Introduction. Foundations of Computing series. MIT Press, Cambridge, Massachusetts, February 1993.
12. C. Hankin. Lambda-calculi. A guide for computer scientist. Oxford University Press, 1994.
13. A.S. Troelstra. Basic Proof Theory. Cambridge University Press, 1996.
14. D. Van Dalen. Logic and Structure. Springer Verlag, terceira edição, 1997.
15. Benjamin C. Pierce, Chris Casinghino, Michael Greenberg, Vilhelm Sjöberg, Brent Yorgey. Software Foundations. Available online at <http://www.cis.upenn.edu/~bcpierce/sf/>
16. Practical Foundations for Programming Languages, by Robert Harper. Manuscript, available from his web page (<http://www.cs.cmu.edu/~rwh/plbook/>)
17. Foundations for Programming Languages, by John C. Mitchell. MIT Press, 1996.

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