

Programação Orientada a Objectos

22/23

Programação Orientada a Objectos 21/22

- Conteúdos programáticos
- Pressupostos
- Avaliação
- Bibliografia
- Ferramentas

Conteúdo programático

- Linguagem C++
- Programação Orientada a Objetos
 - Encapsulamento e abstração de dados
 - Herança
 - Polimorfismo
- Trata-se de um paradigma de programação
 - Poderá ser necessário “desaprender” algumas más práticas de programação imperativa (exemplo, C)
 - O uso de bibliotecas e de lógica “programação em C” é muito penalizado nesta disciplina

Pressupostos

- A programação orientada a objetos inclui todos os conceitos da programação imperativa estudada em
 - Introdução à Programação
 - Programação
- Esta disciplina não vai debater em explicar “os ciclos *for/while*” (IP/P) nem se vão explicar conceitos básicos de algoritmia (IP)
- *Introdução à Programação* e *Programação* são dois pressupostos absolutamente essenciais

Avaliação

- Exame – 12 valores com mínimos de 35%. Presencial
- Trabalho prático – 8 valores
 - Grupos de 2 alunos
 - Duas metas com defesa
 - Meta 1 tem uma avaliação entre 0.8 e 1.0
 - Nota final do TP = nota da meta 1 x avaliação da meta 2
 - A meta 1 sem a meta 2 não serve para nada
 - É possível faltar à meta 1 e fazer a meta 2 (mas perdem-se 20%)
- O exame e as defesas são presenciais

Trabalho prático

- O trabalho é feito durante o semestre, não havendo outras alternativas em outras épocas.
 - Se o trabalho ficar por fazer, a sua cotação fica definitivamente perdida.
 - Depois de feito, a nota do trabalho é válida (aproveitada) para todo o ano letivo.
- TP ano passado:
 - Se realmente realizado em 21/22, e se tiver 50% ou mais.
 - O aproveitamento do trabalho de 21/22 é pedido através do nónio ou moodle (a definir) numa sondagem para o efeito que só será disponibilizada perto do final do semestre.
 - Pedidos de aproveitamento feitos por outras vias são ignorados

Bibliografia

- C++ Annotations, Version 12.2.0, Frank B. Brokken
 - Disponibilizado pelo autor em <http://www.icce.rug.nl/documents/cplusplus/>
 - Este livro e as aulas deverão ser suficientes
- The C++ Programming Language (4th Edition), Bjarne Stroustrup
Addison-Wesley ISBN 978-0321563842
 - Trata-se do “inventor” original da linguagem, mas o é de leitura menos fácil
- Effective C++, Scott Meyers
- Effective Modern C++, Scott Meyers
 - Dois livros bons, mas que exigem que já se saiba C++. Leitura aconselhada para aperfeiçoamento, eventualmente já com a disciplina feita.

Ferramentas necessárias - Compiladores e IDE

-> Qualquer IDE e compilador que não seja antiquado servirá.

Exemplos:

- Visual Studio (a edição Community Edition é gratuita)
- Eclipse ou Netbeans com gcc por baixo (gratuito, mas em desuso)
- CLion (licença gratuita para estudantes)
- Visual Studio Code com um *container* docker gcc (gratuito)
- Evitar coisas obviamente desatualizadas ou menos poderosas (codeblocks e afins)

-> Para o trabalho prático: **CLion**