Ficha de Unidade Curricular

Disciplina: Programação Orientada por Objetos

Área CNAEF: 489 – Informática

Área Científica: 11.3 – Informatics, Computer Science

Precedências, caso existam: Não aplicável

Ano lectivo: 2012/13

Curso(s):

Engenharia Informática, 1º ciclo (LEI)
Engenharia Electrónica e Telecomunicações, Mestrado Integrado (MIEET)

Ano: 2 **Smestre:** 1º

ECTS: 6 **Tempo de trabalho total do aluno (em horas):** 168

Tempo de contacto do aluno (em horas): 30T; 30PL

Professor responsável: Helder Daniel

Professores: Helder Daniel

Resumo descritivo:
Esta unidade curricular (UC) tem como objetivo o estudo e aplicação dos princípios e técnicas de programação orientada por objetos; a introdução à modelação orientada por objetos e ao UML *(Unified Modeling Language)*; e o estudo e aplicação de Padrões de Projeto.
No final desta disciplina os alunos deverão ser capazes de gerar uma especificação UML recorrendo a Padrões de Projeto e implementar essa especificação na linguagem JAVA.

É recomendado que os alunos tenham domínio prévio dos fundamentos da programação e da programação imperativa.

Programa
Os resultados esperados de aprendizagem nesta unidade curricular são as seguintes:

- Enumerar, descrever e justificar os conceitos, princípios e técnicas da programação orientada por objetos.
- Modelar aplicações usando UML, designadamente no que se refere aos aspetos de arquitetura (diagramas de classes e objetos) e de colaboração (diagrama de sequência).
- Selecionar e utilizar padrões de projeto na resolução de problemas.
- Usar Java como linguagem de programação.

Competências a desenvolver

- Entender e aplicar os princípios e as técnicas de programação orientada por objetos.
- Gerar uma especificação UML recorrendo a padrões de projeto e implementar essa especificação na linguagem JAVA.

Conteúdos programáticos detalhados:

1. Conceitos básicos
2. Introdução à modelação orientada por objetos e ao UML (Unified Modeling Language)
   2.1. Diagrama de classes
   2.2. Diagrama de objetos
3. Princípios, conceitos e técnicas
   3.1. Estruturas de controlo
   3.2. Semântica de atribuição
   3.3. Binding
   3.4. Polimorfismo
   3.5. Classes abstratas
   3.6. Coleções e Iteradores
4. Programação Orientada por Objetos usando Java
   4.1. Contexto histórico
   4.2. Noção de referência
   4.3. Classes
      4.3.1. Membros privados, protegidos e públicos;
      4.3.2. Inicialização de objetos: Construtores e a Condição Invariante de Instância.
      4.3.3. Manipulação do recetor: a palavra reservada this
      4.3.4. Espaços de nomes
      4.3.5. Sobrecarga de nomes
      4.3.6. Operadores
      4.3.7. Conversões definidas pelo utilizador.
      4.3.8. Finalização de objetos
      4.3.9. Membros estáticos e membros constantes.
   4.4. Gestão de memória
   4.5. Herança e derivação.
      4.5.1. Especificação de derivação.
      4.5.2. Mecanismos de herança
   4.6. Interfaces
      4.6.1. Herança múltipla de interfaces
   4.7. Exceções e asserções
5. Padrões de projeto fundamentais
6. Estruturas de dados elementares
   6.1. Pilhas
   6.2. Filas

Métodos de ensino-aprendizagem:

A unidade curricular está dividida numa componente teórica e numa prática (laboratorial).

Nas aulas teóricas os conteúdos são apresentados e discutidos. A apresentação é efectuada com apoio de computador para projecção de transparências e execução de demonstrações.

Na componente prática são aplicados os conhecimentos apreendidos na teórica no desenvolvido de mini-projectos de programação, passando por atividades que vão desde a modelação à implementação e depuração.

O conteúdo teórico lecionado está publicado na tutoria electrónica da UAlg (moodle), bem como todos os enunciados das fichas práticas, e outros materiais de apoio.

A inscrição nos turnos práticos e submissão dos trabalhos práticos é efetuada com apoio de uma aplicação *web* desenvolvida especialmente para esse efeito.

Métodos de avaliação, incluindo forma de cálculo da classificação final:

A avaliação, em qualquer época (normal, recurso, especial para finalistas, melhoria de classificação e trabalhadores estudantes) é composta por duas componentes:

classificação final = 60% Exame + 40 % Avaliação prática

(Aprovação se classificação final >= 9,5 valores)

Exame

Os exames consistem numa prova escrita com consulta de até 25 páginas A4, com qualquer conteúdo que cada aluno considere conveniente.

Avaliação prática

1. Os trabalhos são realizados em grupos.
2. Os grupos são constituídos por, no máximo, três elementos.
3. A inscrição num grupo de trabalho é obrigatória (i.e., só são aceites trabalhos de alunos inscritos em grupos).
4. Cada grupo entrega uma única resolução dos trabalhos:
   - 4.1. A entrega de alguns trabalhos consiste apenas no código fonte que é avaliado imediatamente usando o Mooshak.
   - 4.2. A entrega de outros trabalhos consiste em dois itens: código fonte e relatório.
     - 4.2.1. O relatório deve ser completo, conciso e claro e inclui:
       - Descrição das opções tomadas, a estrutura e organização do programa;
       - Descrição das classes e métodos;
       - Não se consideram listagens de código
5. Critérios de Avaliação dos Trabalhos:
   - 5.1. Correção: Não se consideram características que excedam as especificações.
   - 5.2. Eficiência (utilização criteriosa de recursos)
   - 5.3. Modularidade. Será dada particular importância aos seguintes aspetos:
     - Organização do programa em componentes coerentes e independentes
     - Reutilização de código
   - 5.4. Legibilidade incluindo comentários adequados no código (não se consideram comentários sobre sintaxe ou semântica da linguagem de programação).
   - 5.5. Em caso de necessidade de qualquer tipo de esclarecimento sobre os trabalhos poderão ser agendadas discussões destes.
   - 5.6. A nota prática é individual, estando dependente do desempenho de cada elemento do grupo.

Critérios de admissão a exame, incluindo as consequências das faltas às componentes de avaliação previstas:

São admitidos a exame, na época normal e de recurso, os alunos cuja nota de nota de avaliação prática seja maior ou igual a 7,5 valores.

Bibliografia básica:

[Referência principal]

Bruce Eckel. “Thinking in Java, 4th edition”, Prentice Hall, New Jersey, 2006, cf. http://mindview.net/Books/TIJ4

[Referência complementar sobre padrões de projeto]
Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides. “Design Pattern – Elements of Reusable Object-Oriented Software”, Addison-Wesley, 1995

[Referência complementar sobre UML]
UML Quick Ref: http://www.uml.org

[Referência complementar sobre Estruturas de Dados]
Robert Sedgewick. “Algorithms in Java, 3rd edition – parts 1-4: Fundamentals, data structures, sorting, searching”, Addison-Wesley, 2003