MÉTODOS DE PROGRAMAÇÃO II

1º ANO DA LEC

PROGRAMAÇÃO ORIENTADA AOS OBJECTOS USANDO A LINGUAGEM JAVA 2

NOTAS TEÓRICAS

Prof. F. Mário Martins

fmm@di.uminho.pt

Departamento de Informática
Universidade do Minho

2004/2005

MÉTODOS DE PROGRAMAÇÃO II

Departamento de Informática / Escola de Engenharia
Universidade do Minho

Ano Lectivo 2004/2005 (2º semestre)

Escolaridade

2 TP

Cursos a que é leccionada:

Curso	Ano	Código
Engenharia de Comunicações	*1º*	

Responsável:

Prof. F. Mário Martins

Equipa Docente:

Docente	T	TP
Prof. F. Mário Martins		*2 (1 x 2h) LEC*

ESTRUTURA DE FUNCIONAMENTO:

Aulas teóricas onde são introduzidos os conceitos fundamentais deste paradigma tão importante e tão particular, em articulação com as aulas teórico-práticas que têm por objectivo consolidar tais noções à custa da exemplificação da sua utilização (ou até das consequências da sua falta de utilização) em programas escritos utilizando **JAVA** como linguagem de programação.

Estrutura e Objectivos Pedagógicos:

As aulas teóricas terão por objectivo a exposição da matéria fundamental que permita caracterizar o *paradigma da programação orientada aos objectos*.

Esta disciplina pretende realizar a transição entre a *programação em pequena escala* até aqui realizada nas disciplinas anteriores, onde se estudaram estruturas de dados mas onde as mesmas raramente foram aplicadas em aplicações de envergadura real, introduzindo-se agora todos os problemas inerentes ao desenvolvimento de aplicações de carácter e envergadura reais. Surgem assim, conceptual e pragmaticamente questões tais como a modularidade conceptual e de implementação, questões relacionadas com a correcção de erros e eficiência, e, ainda, com a própria facilidade de manter e estender as aplicações desenvolvidas. Procura mostrar-se como as metodologias orientadas aos objectos permitem satisfazer um elevado e importante número de princípios fundamentais da Engenharia de Software.

Pretende-se atingir uma boa compreensão da evolução histórica do paradigma, surgido como muitos outros nos anos 70, mas que só mais recentemente atingiu uma enorme projecção fora das universidades, das suas principais características identificadoras e dos conceitos formais que o consubstanciam e distinguem de outros.

A título de exemplo, e procurando introduzir certas novas noções a partir de outras noções introduzidas em anteriores disciplinas, o conceito de *objecto*, unidade modular fundamental ao paradigma, é associada às noções de módulo, abstracção de dados, protecção e encapsulamento tão importantes para a independência do software.

Não se pretende a este nível a apresentação formal de conceitos tais como objecto, classe, polimorfismo, hierarquia, herança, agregação, etc. Porém, sendo estes os conceitos fundamentais à compreensão não só da tecnologia mas também dos métodos de análise e concepção de software baseada no paradigma, em termos pragmáticos, os mesmos deverão ser profundamente apresentados (segundo diferentes

perspectivas da sua implementação até) e no final da disciplina perfeitamente assimilados.

Apresentar-se-ão igualmente alguns dos principais problemas típicos que se colocam na concepção de aplicações orientadas aos objectos, em particular os que têm a ver com decisões de classificação de novas classes desenvolvidas. Aqui procura-se uma clara distinção entre os dois principais tipos de herança (lógica ou física, simples ou múltipla) e ainda a distinção clara entre os mecanismos de herança e de agregação.

Ao longo da disciplina procurar-se-á sempre, ao nível das aulas teóricas, apresentar comparações entre as mais diversas maneiras como a actual tecnologia baseada no paradigma dos objectos implementa tais conceitos fundamentais. Apresentam-se assim as implementações realizadas em algumas linguagens de objectos mais usadas, em particular comparando a linguagem usada na disciplina, JAVA, com C++.

Numa primeira fase do curso, a preocupação centrar-se-á na construção da camada computacional de aplicações orientadas aos objectos em isolamento da camada inter-activa. Na parte final do curso, se possível, abordar-se-á o modelo de interacção do JAVA.

As aulas teórico-práticas, todas realizadas em laboratórios, consistirão, desde o seu início, da apresentação sintética e estudo, sob a forma de resolução de pequenos problemas adequados, da linguagem JAVA, em particular das regras a cumprir para realizar PPO em JAVA, suas Classes e respectivos métodos.

As aulas teórico-práticas são realizadas em laboratórios e têm sempre componente prática em computador. Será usado ambiente de desenvolvimento integrado (IDE) BlueJ como sistema de apoio ao desenvolvimento e prototipagem das pequenas aplicações que forem sendo desenvolvidas, com a vantagem adicional da visualização de Diagramas de Classes, etc.

Numa fase posterior, e admitindo um razoável conhecimento por parte dos alunos das principais classes e métodos, bem como de alguns conceitos introduzidos nas aulas teóricas, as aulas teórico-práticas procurarão exercitar questões relacionadas com a concepção de maiores aplicações, nas quais o domínio dos mecanismos de herança e de agregação, de classificação, e de metodologia de concepção se torna importante.

As aulas teórico-práticas terminarão com a resolução de um ou dois exercícios onde o correcto uso de certas classes existentes em JAVA permitirá a construção final da desejada aplicação interactiva. A extensiva utilização do *mecanismo de excepções* de JAVA permitirá chamar a atenção dos alunos para os tão importantes aspectos da segurança e robustez do software, e, ao usar JAVA, tirar todo o potencial do muito claro e limpo mecanismo de tratamento de excepções.

São palavras e ideias chave da engenharia do software que importa reforçar como bem abordadas por este paradigma: a *modularidade* e o *encapsulamento* (pelo uso de objectos, classes e mensagens), a *flexibilidade* (via polimorfismo), a *classificação* (via hierarquia), a *reutilização* (via mecanismo de herança e agregação), a *extensibilidade* e a *generalidade* (via polimorfismo natural), entre outras.

Deve ainda ser realçada a *verticalidade* da utilização destes conceitos no projecto de software, dado que os mesmos podem ser aplicados desde a análise à implementação de sistemas seguindo o paradigma OO.

Síntese de Objectivos:

A disciplina de Paradigmas da Programação IV tem por objectivo completar a formação dos alunos na área da programação, pela introdução de outros modelos de programação existentes com grande capacidade na resolução de classes particulares de problemas. É objectivo da disciplina a apresentação do *Paradigma da Programação Orientada aos Objectos*, das suas bases formais, das suas capacidades específicas e das áreas da sua particular aplicação.

É particularmente importante que esta disciplina e este paradigma estabeleçam, ao nível do curso, a distinção entre a *programação em pequena escala* e os problemas da *programação em grande escala*, designadamente a adopção de técnicas de concepção e desenvolvimento modulares e escaláveis, em particular explorando todas as que o paradigma da programação por objectos oferece, neste caso particular usando as características da linguagem JAVA.

Compreendidas as potencialidades do paradigma para a resolução de certas categorias de problemas, pretende-se, do ponto de vista prático, que os alunos se tornem auto-suficientes na escrita de aplicações em JAVA, usem em seu favor o IDE BlueJ, até para documentação rápida de projecto, e adquiram o conhecimento genérico

suficiente para que, posteriormente, e por si, possam, se tal for necessário, desenvolver as suas capacidades de utilização da linguagem e dos conceitos em disciplinas mais avançadas do curso, tais como Desenvolvimento de Sistemas de Informação, Técnicas Avançadas de Orientação aos Objectos, Bases de Dados Orientadas aos Objectos, Sistemas Multimédia, Sistemas Operativos, Criptografia, e, de forma geral, em quase todos os projectos de investigação.

Sistema de Avaliação:

A avaliação tem uma componente teórica e uma componente prática, ambas obrigatórias. Tal significa que um aluno que não obtenha a classificação mínima fixada para cada componente não será aprovado.

A **componente prática** consistirá da realização de **1 trabalho prático**, sob a forma de trabalho de grupo de no máximo 3 elementos. A não realização do(s) trabalho(s) implica de imediato a reprovação do aluno à disciplina, passando a ser **não admitido** a exame, e considerado para efeitos estatísticos como **não avaliado**.
Para a componente prática a nota mínima deverá ser de **10 valores**, caso contrário o aluno é igualmente **não admitido**, ainda que seja estatisticamente considerado como **avaliado**.

Nas aulas teórico-práticas da disciplina são propostos e acompanhados pequenos exercícios, agrupados em fichas semanais teórico-práticas. Tais trabalhos pretendem servir de guião à componente teórico-prática da cadeira, e fios condutores do estudo dos alunos, tendo ainda como objectivo serem auxiliares à resolução do trabalho final. Tais trabalhos servirão também de base para muitas das questões que serão colocadas aos alunos nos exames que dizem respeito à avaliação da sua formação teórica.

A **nota teórica** será obtida através da realização de **1 teste individual escrito**, sendo a nota mínima necessária para a realização da componente teórica **9,5 valores**. Caso o aluno obtenha uma classificação entre 9 e 9,4, esta nota teórica, devidamente pesada, é considerada para efeitos de média com a nota prática, sendo-lhe no entanto descontado 1 valor à sua média final, devendo esta ser obviamente positiva após tal ajuste (>= 9,5).

A nota teórica deverá ser obtida em exame, desde que o aluno possua já nota prática. O exame poderá ser realizado numa das duas chamadas da época normal de

Junho/Julho, ou na respectiva época de recurso (ou ainda na de Novembro para os alunos para tal habilitados).

A **nota final** da disciplina será encontrada, após satisfação das regras anteriores, e salvo o caso especial de nota teórica entre 9 e 9,4 anteriormente referido, pela aplicação da seguinte fórmula simples:

Nota Final = (Nota Teórica x 0, 55) + (Nota Prática x 0, 45)

O trabalho prático entregue pelos alunos de um dado grupo terá uma classificação que poderá, quando tal se justifique, ser individualizada. A não presença, injustificada, de um dos elementos de um dado grupo na apresentação e discussão do respectivo trabalho implica a sua não avaliação e consequente reprovação.

A classificação final do trabalho prático entregue, deverá ser calculada em função da seguinte escala de critérios e valores:

Escalão	Nota
Sem qualidade	6
Pouca qualidade	8/9
Qualidade mínima	10
Qualidade média	13
Bom trabalho	15
Muito bom trabalho	17
Trabalho excelente	18-20

A avaliação dos trabalhos terá em consideração diversas componentes, tais como:

- *pontualidade na entrega do trabalho;*
- *qualidade e complexidade das decisões de projecto;*
- *qualidade, em apresentação e síntese, do relatório apresentado;*
- *qualidade do código fonte apresentado;*
- *qualidade da execução (com ou sem erros, satisfaz ou não requisitos, etc.);*
- *qualidade da apresentação ao utilizador (interface), caso tal se aplique;*
- *facilidade de utilização do programa sem ler manuais;*
- *nível da prestação oral dos elementos do grupo;*

PARADIGMAS DE PROGRAMAÇÃO IV

CONTEÚDO PROGRAMÁTICO DETALHADO

PROGRAMAÇÃO POR OBJECTOS EM JAVA4

1. MATÉRIA TEÓRICA DE PPO:

1.1.- Introdução à Programação por Objectos.

- Origem do paradigma. Via Simulação. Via Computação.
- Conceitos básicos fundamentais.
- Modelos: de processos versus de objectos.
- A procura da modularidade no software.
- Independência do contexto como condição fundamental.
- Encapsulamento versus independência e modularidade.
- Modularização pelos dados: a solução em PPO.

1.2.- Noção de "Objecto" em PPO.

- Noção de "objecto" em PPO. Estrutura e Comportamento.
- Encapsulamento e protecção nos objectos.
- Interacção entre objectos. Mensagens vs. Métodos.
- Introdução ao Polimorfismo.
- Tipos de objectos: instâncias e classes.

1.3.- Classes, Hierarquia de Classes e Herança.

- Definição de Classe em PPO.
- Relação Classe-Instâncias. Introdução.
- Mecanismo de instanciação.
- Classes e sua Hierarquia. Superclassificação.
- Relações entre Classes. A herança.
- Herança lógica versus herança de implementação.
- Herança como mecanismo de reutilização e de programação incremental.
- Herança simples e múltipla.
- Algoritmo de procura de métodos.
- Herança versus Agregação.

1.4.- Classes e Herança.

- Criação de Classes.
- Classes "run-time" versus Classes para "compile-time".
- Tipos estáticos e dinâmicos das variáveis.
- Polimorfismo, “static” e “dynamic binding”.
- Classes não instanciáveis.

1.5.- Classes Abstractas.

- Definição de Classe Abstracta. Importância das Classes Abstractas.
- Classes Abstractas vistas como Tipos Abstractos de Dados.
- Classes Abstractas como mecanismo de abstracção.
- Classes Abstractas como mecanismo de reutilização e de extensibilidade.
- Polimorfismo. Estudo dos diferentes tipos.

1.6.- Concepção de aplicações em PPO.

- Subclassificação e herança versus agregação.
- Subclasses como especializações.
- Subclasses para implementação.
- Algumas regras de concepção em PPO.

1.9.- Actuais principais aplicações do paradigma dos objectos.

- Linguagens e Tecnologia de interacção.
- Tecnologia de Sistemas Operativos e "plataformas".
- Metodologias OO de Análise e Concepção de Software (UML, Rational Rose, Together, etc.

2. PROGRAMAÇÃO POR OBJECTOS EM JAVA: ESTUDO DA LINGUAGEM JAVA4 (JAVA 1.41)

2.1.- Programação por Objectos em JAVA.
Características do ambiente de desenvolvimento JDK.
A JVM ("Java Virtual Machine"). Byte-code.
Estrutura dos programas.
Bibliotecas. Packages.
Introdução ao IDE Bluej. Características e funcionalidade.

2.2.- Tipos básicos (não objectos) e operadores.
Numéricos. Booleanos. Declarações.
Arrays Java e suas inconveniências vs. a classe Vector ou ArrayList.

2.3.- Estruturas de controlo.
Condicionais simples e compostas.
Estruturas Iterativas.

2.4.- Definição de Classes e Instâncias em JAVA.
Construtores. Métodos e variáveis de instância e de classe.
Tipos de qualificadores de visibilidade e acesso das variáveis e constantes.

2.5.- Hierarquia de Classes em JAVA.
Classe Object. Classes versus Packages.
Herança simples.
Redefinição e sobreposição de métodos e variáveis.
Classes e subclasses. Exemplos clássicos.
Compatibilidades entre instâncias de classes e subclasses.
O mecanismo de "dynamic type checking".

2.6.- Classes Abstractas em JAVA.
Declaração.
Polimorfismo e sua utilização. Regras da linguagem.
"Static-checking" vs. "Run-time checking" em JAVA.
Exemplos com classes abstractas.
"Casting".

2.7.- O mecanismo de Excepções da linguagem JAVA.
Cláusulas try, catch, finally, throws e throw.
Regras de utilização.

2.8.- *Interfaces* JAVA como especificações de Tipos de Dados.
Classes como subclasses e classes como subtipos. Análise aprofundada.
Herança múltipla de Interfaces em JAVA.
Regras para a implementação de Interfaces em Classes.

2.9.- Estudo das *Streams* de JAVA.
Streams de caracteres versus streams de bytes.
Streams de input e streams de output. As classes abstractas Writer e Reader.

Subclasses de Writer e Reader. Mecanismo de "aninhamento" de streams.
As ObjectStreams como mecanismo de persistência de dados. Serializable.
Exemplos de eficiência no uso de streams na gravação e leitura de 300.000 fichas com a informação típica de um Aluno. Medida de eficiência.
Comparação da eficiência das diversas soluções.

2.10.- Os Packages JAVA como mecanismos de Meta-Modularidade.

Estudo dos packages principais de JAVA.

Revisão das classes e interfaces fundamentais de JAVA, tais como Object, Number, Math, Array, Class, StringBuffer, Collections (cf. Vector, Stack, ArrayList, Hashtable, HashMap, LinkedList), Number, StringTokenizer, Stream, Exception e outras, e das interfaces List, Map, Iterator, Enumeration, Cloneable e Serializable.

Importância das "inner classes" e de outras construções de JAVA como suporte efectivo a implementações genéricas de estruturas de dados.

BIBLIOGRAFIA

SOBRE O PARADIGMA DA PROGRAMAÇÃO POR OBJECTOS USANDO A LINGUAGEM JAVA2

Programação Orientada aos Objectos em JAVA2
F. Mário Martins, Editora FCA, Série Tecnologias de Informação,
ISBN-972-722-196-3, 1ª edição, Setembro de 2000, 4ª Edição, Fevereiro de 2004;

% Um livro que apresenta as características fundamentais do paradigma da PPO, e como tais características podem e devem ser implementadas usando, por exemplo, JAVA, para a criação metodológica de aplicações %

Paradigmas da Programação IV
Programação Orientada aos Objectos em JAVA
F. Mário Martins, Notas Pedagógicas, revisão de 2005.

% O conjunto dos apontamentos teóricos da disciplina, tal como apresentados nas aulas teóricas, aos quais se anexam diversos exemplos concretos de pequenos projectos de preparação. %

Object Oriented Design with Applications
G. Booch, The Benjamim Cummings Pub. Company, USA, 1991.

% Embora seja o livro onde o autor apresenta a sua metodologia para concepção de aplicações orientadas aos objectos, possui definições profundas dos principais conceitos existentes em PPO. Muito bom livro. No mínimo merece consulta. %

An Introduction to Object Oriented Programming
T. Budd, Addison-Wesley, 2nd Edition, 1997

% Livro de síntese dos principais conceitos do paradigma da PPO, e que apresenta um estudo comparativo sobre como tais conceitos são implementados nas mais relevantes linguagens de PPO, designadamente, JAVA, C++, ObjectPascal, Objective C, Eiffel, etc. %

SOBRE A LINGUAGEM JAVA
(não necessariamente bons para PPO)

Dominando o JAVA
P. Naughton, McGraw-Hill, 1996

% Livro em português do Brasil, que tem algum interesse apenas na fase inicial da aprendizagem de JAVA dado ser muito simples e muito limitado. Apresenta alguns erros graves e código fonte não correspondente aos resultados depois apresentados. Encontrar estes erros poderá ser um exercício interessante. Para os primeiros passos em JAVA, em particular para a sintaxe básica e familiarização com algumas classes vale a leitura dado não existirem muitas alternativas em simplicidade. %

Core Java
G. Cornell, C. Horstmann
Prentice-Hall, 1996.

% Livro bastante rigoroso e completo. Não acessível numa primeira fase do curso. Os autores usam muitas vezes classes não existentes no JDK1.1. É no entanto um verdadeiro livro de programação em JAVA, apresentando aplicações muito interessantes, podendo-se aprender bastante com o código apresentado.
Por exemplo, a programação com Streams e Applets, ainda que com problemas de compatibilidade com o JDK1.4, é ilustrada com base em aplicações com interesse %

Java in a Nutshell
D. Flanagan
O´Reilly & Associates, 3rd Edition, 1999

% A 3ª edição do livro anterior. Compatível com JDK1.4. Apresenta ainda algumas novidades (cf. classes anónimas, etc.) %

Data Structures in JAVA
T. Standish
Addison Wesley, 1998

% Livro bastante bom que ensina a implementar em Java as principais estruturas de dados usadas em programação, introduzindo ainda medidas simples de eficiência e complexidade .%

Data Structures & Problem Solving using JAVA
M. Weiss
Addison Wesley, 1998

% Livro interessante e complementar ao anterior. %

SOBRE BLUEJ

BlueJ Manual On-Line, M. Kolling, em www.bluej.org
"Objects First with Java", D. Barnes e M. Kolling, Prentice Hall-Person Education, 2003.

Notas das Aulas Teórico Práticas de PPIV (on-line na página de PPIV)
Mário Martins, UM/DI, 2002

% Todos os exercícios resolvidos nas aulas teórico-práticas por mim leccionadas %

Manuais ON-LINE, APIs, Fontes, etc. de JAVA j2sdk1.4.2_04
Página da disciplina: http://www.di.uminho.pt
Ver ainda: http://www.sun.com e
http://www.javasoft.com e
http://www.bluej.org

NOTAS DE FUNCIONAMENTO IMPORTANTES:

- **CADA AULA TP TERÁ UM ASSUNTO TEÓRICO DE EXPOSIÇÃO E, SE POSSÍVEL, ALGUMA EXPERIMENTAÇÃO SOBRE O MESMO;**

- **SERÁ REALIZADO APENAS 1 TRABALHO PRÁTICO, CUJO ENUNCIADO SERÁ TORNADO PÚBLICO EM ABRIL;**

- **A ENTREGA DO TRABALHO REALIZAR-SE-Á EM DATA A COMBINAR (FINAL DE MAIO);**

- **NA PÁGINA FOI ABERTO UM FÓRUM PARA DÚVIDAS VIA O ENDEREÇO DE E-MAIL: duvidas_MPIIEC@di.uminho.pt**