



Universidade do Minho
Escola de Engenharia
Departamento de Informática

Programação Orientada aos Objectos

MiEI - 2º ano / 2º semestre

2016/2017

António Nestor Ribeiro
anr@di.uminho.pt

<http://www.di.uminho.pt>



1

Escolaridade

- 2 T + 2PL

Equipa Docente

- Aulas T
 - António Nestor Ribeiro (anr@di.uminho.pt)
- Aulas PL
 - António Luís Sousa (als@di.uminho.pt) (PL4, PL5)
 - João Saraiva (jas@di.uminho.pt) (PL1, PL6)
 - José Creissac Campos (jose.campos@di.uminho.pt) (PL2)
 - António Nestor Ribeiro (anr@di.uminho.pt) (PL3)



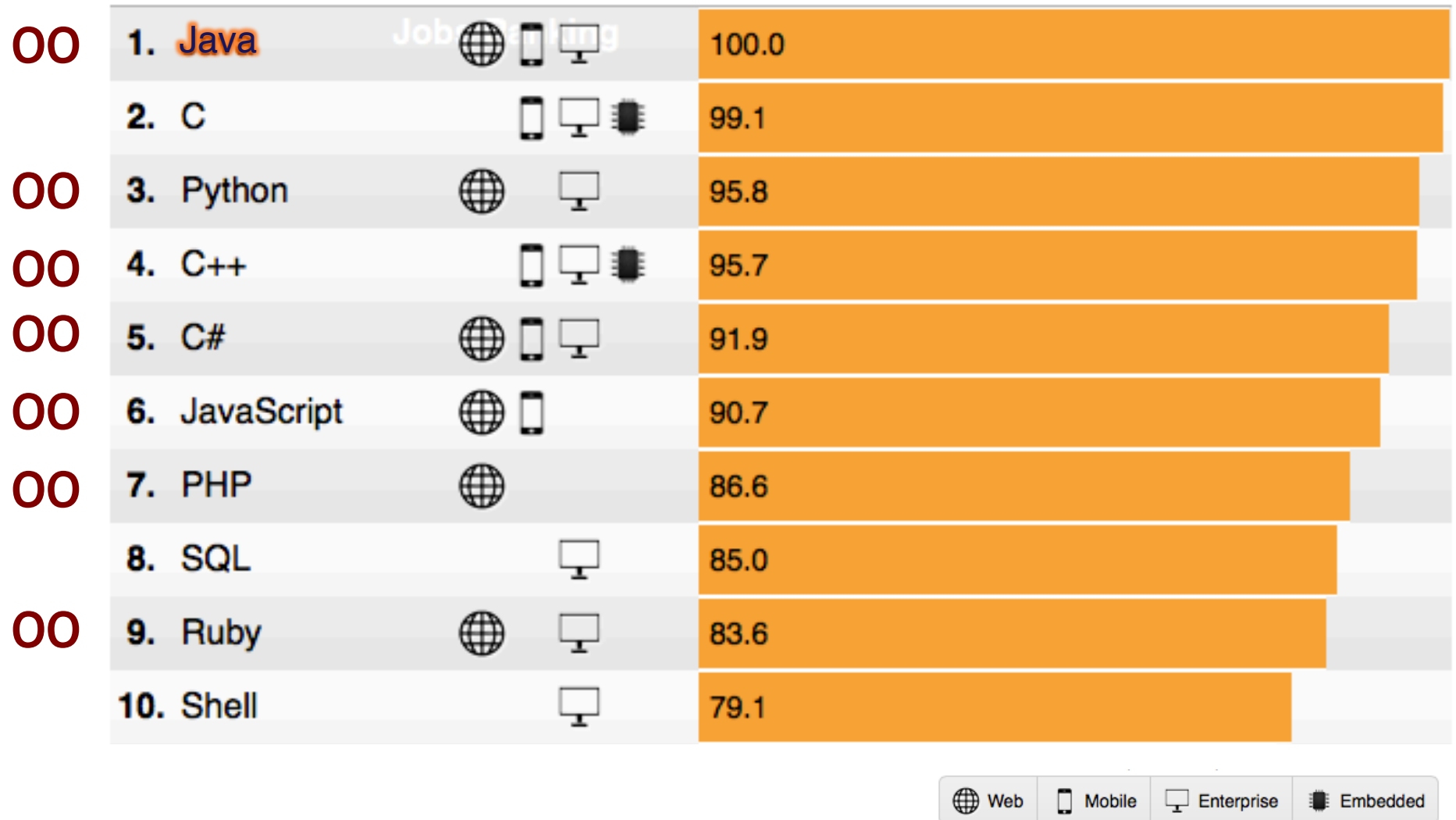
Apresentação de POO

- Programa
- Motivação
- Objectivos e resultados de aprendizagem
- Bibliografia
- Ferramentas
- Método de Avaliação



Programa

- Paradigma da programação Orientada aos Objectos:
 - Abstracção de Dados, Encapsulamento e Modularidade.
 - Objectos: estrutura e comportamento.
 - Mensagens.
 - Classes, hierarquia e herança.
 - Herança versus Composição.
 - Classes abstractas.
 - O princípio da substituição.
 - Dynamic binding.
 - Polimorfismo.
 - Interfaces parametrizadas.
 - Iteradores.
 - Tipos List, Map e Set.
 - Tipos Enumerados.
 - Streams: de caracteres, de bytes e de objectos.
 - Classes Genéricas.
- Tecnologia de programação JAVA:
 - Plataforma J2SE: JDK, JVM e byte-code.
 - Construções básicas: tipos primitivos e operadores.
 - Estruturas de controlo.
 - I/O básico.
 - Arrays.
 - Nível dos objectos: Classes e instâncias.
 - Construtores.
 - Métodos e variáveis de instância.
 - Modificadores de acesso.
 - Importação normal e estática.
 - Métodos e variáveis de classe.
 - Hierarquia de classes e herança.
 - Overloading e overriding de métodos.
 - Classes Abstractas.
 - Interfaces e tipos definidos pelo utilizador.
 - Tipo estático e dinâmico.
 - Procura dinâmica de métodos.
 - Polimorfismo e extensibilidade.
 - Colecções genéricas.



Linguagens mais pretendidas pelos empregadores em 2015
(IEEE Spectrum)



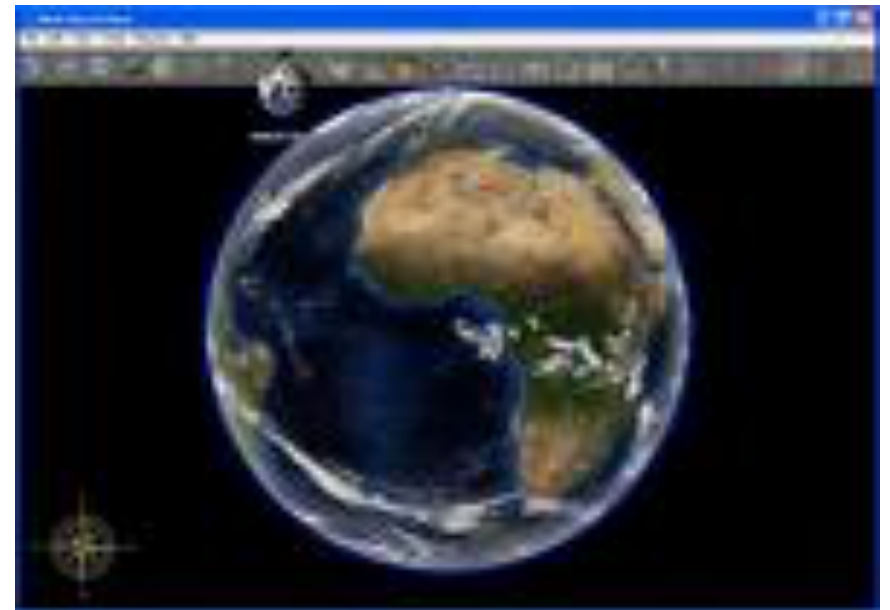
Exemplos de aplicação de Java



Project Looking
Glass: Desktop 3D



UltraMixer:
DJ mixing s/w



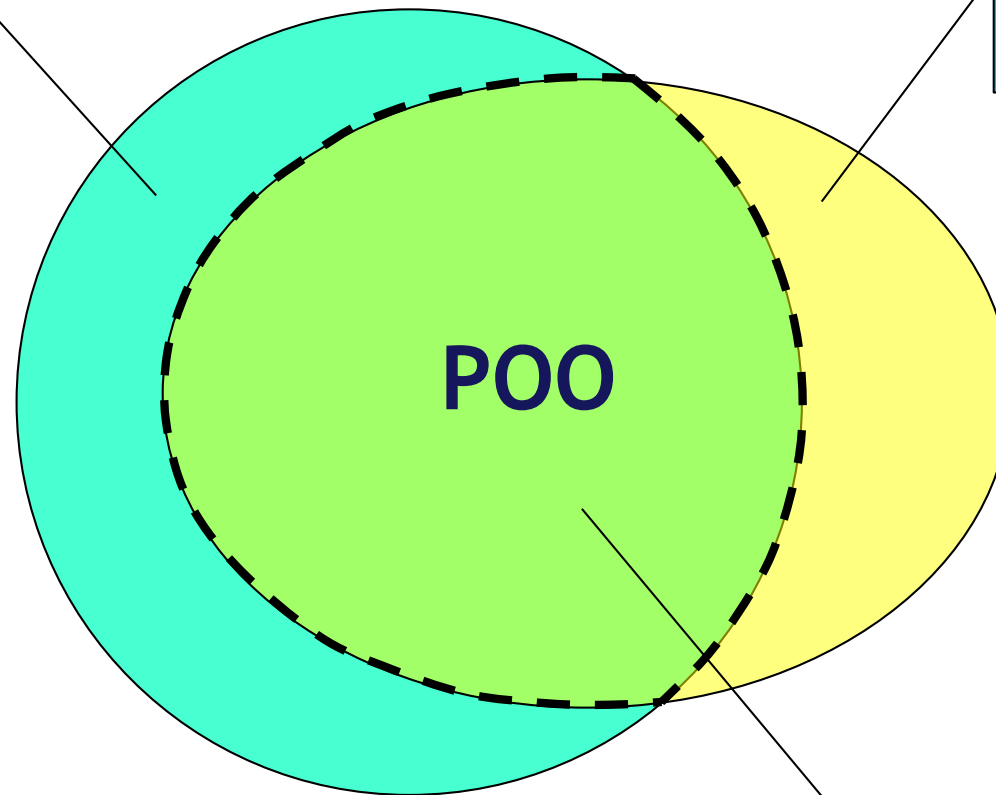
NASA World Wind:
Semelhante ao
Google Earth



Programar de acordo com o Paradigma!

Paradigma
Programação Orientada
aos Objectos

Tecnologia
Java



Programas Java que
seguem as boas práticas
do Paradigma OO



Resultados da aprendizagem

- a) Compreender os conceitos fundamentais da PPO (Objectos, Classes, Herança e Polimorfismo);
- b) Compreender como os conceitos básicos da PPO são implementados em construções JAVA;
- c) Compreender princípios e técnicas a empregar em programação de larga escala;
- d) Desenvolver o modelo de classes e interfaces para um dado problema de software (modelação);
- e) Desenvolver e implementar aplicações Java de média escala, seguras, robustas e extensíveis.



Programa Detalhado

Matéria Teórica

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. Construtores.
- Classes e sua Hierarquia. Superclassificação.
- Relações entre Classes. A herança.
- Herança como mecanismo de reutilização e de programação incremental.
- Herança simples e múltipla.
- Algoritmo de procura de métodos.
- Herança versus Agregação.



Programa Detalhado

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.



Programa Detalhado

Matéria Prática

- 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.
- Tipos básicos (não objectos) e operadores.
 - Numéricos. Boleanos. Declarações.
 - Arrays Java e suas vantagens e inconvenientes vs. ArrayList<E>.
- Estruturas de controlo.
 - Condicionais simples e compostas.
 - Estruturas Iterativas.
- 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.
- 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”.
 - Métodos equals(), clone() e toString().
- 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”.



Programa Detalhado

- O mecanismo de Excepções da linguagem JAVA.
Cláusulas try, catch, finally, throws e throw.
Regras de utilização.
- 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.
- Tipos parametrizados e Colecções de JAVA6.
Estudo da JCF (“Java Collections Framework”).
Os três tipos de colecções de JAVA: List, Set e Map.
Implementações de List<E>: ArrayList<E>: for(each) e iteradores;
O problema do clone() de colecções.
Colecções não-covariantes e “Wildcards”.
Implementações de Set<E>: HashSet<E> e TreeSet<E>;
Boxing e Unboxing automáticos.
Implementações de Map<K,V>: HashMap<K,V>, TreeSet<K, V>;
Implementação da interface Comparator<E>
Tipos Enumerados.
- 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. Interface Serializable.
Exemplos de eficiência no uso de streams. Exemplos de teste.
Medida de eficiência. Comparação da eficiência das diversas soluções.



Bibliografia

JAVA6 e Programação Orientada pelos Objectos

F. Mário Martins, Editora FCA, Série Tecnologias de Informação, Julho de 2009. ISBN 978-972-722-624-5

Objects First with Java - A Practical Introduction using BlueJ, Fifth edition

David J. Barnes & Michael Kölling, Prentice Hall / Pearson Education, 2012.
ISBN 978-013-283554-1

Object Oriented Design with Applications

G. Booch, The Benjamin Cummings Pub. Company, USA, 1991.

Projetos de POO em JAVA

F. Mário Martins, Editora FCA, Série Tecnologias de Informação, 2014.
ISBN 978-972-722-792-1



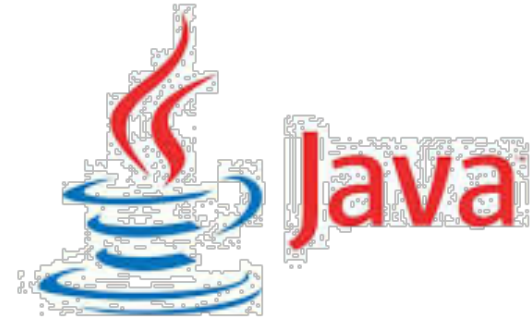
Bibliografia Online / Ferramentas

- Java:

<http://docs.oracle.com/javase/tutorial/>

documentação, SDK e máquina virtual

- Java SE - Standard Edition
- Java EE - Enterprise Edition
- Java ME - Micro Edition



- BlueJ

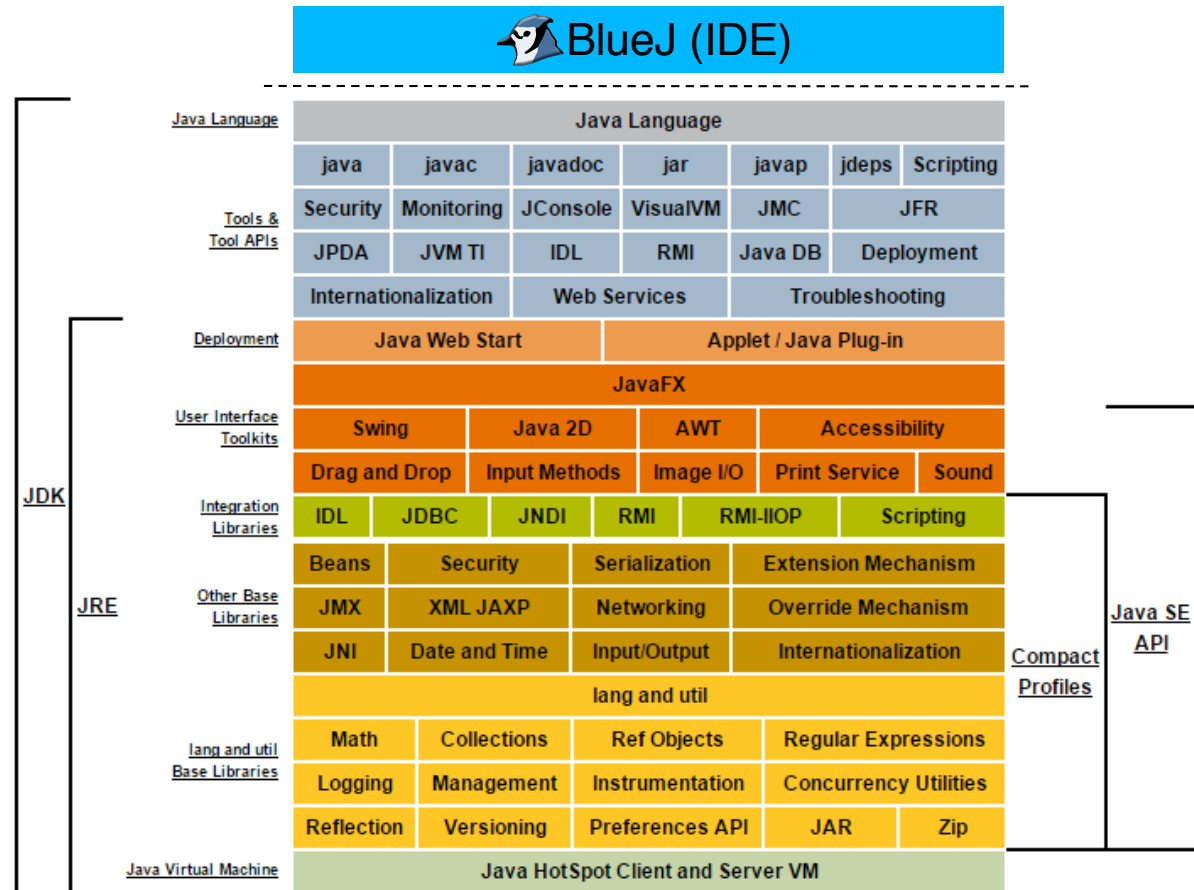
<http://bluej.org>

documentação e ambiente de edição, compilação e debug





JDK (Java Development Kit) / BlueJ





Avaliação

- . Teórica: 1 teste teórico com nota mínima de 9.0 valores
Data do teste: 30/05/2017
Data do exame: 28/06/2017
- . Prática: 1 projecto em grupo com nota mínima de 10.0 valores
Data de entrega electrónica: 03/06/2017
Data apresentação: semana de 5/6 a 9/6
- . **Nota Final** = $45\% \times \text{Prática} + 55\% \times \text{Teórica}$
- . **Nota Final** ≥ 10.0 , para obter sucesso à UC de POO



Métodos de Avaliação

- . A: obrigatório para alunos de primeira inscrição
 - . Avaliação teórica + projecto
 - . Número máximo de faltas permitido: 5 faltas
- . B: para alunos com mais de uma inscrição a POO
 - . Podem utilizar ('congelar') a nota prática de 2015/16
 - . A nota prática congelada ficará limitada a um máximo de 15 valores
 - . Não se efectuam congelamentos de notas 10.
 - . Alunos que queiram congelar a nota de 2015/16 devem comunicar o pedido até ao fim do mês de Fevereiro (vai ser criado um grupo no Blackboard).
 - . Sem reprovação por faltas.



Projecto

- . Tem duas fases de entrega:
 - . Uma intermédia (electrónica) em data a definir
 - . Uma final com relatório, a ser entregue electronicamente
- . O enunciado será disponibilizado na semana de 27/2 a 03/3.
- . Os trabalhos são efectuados por grupos de 3 alunos
- . Na altura da apresentação os elementos de cada grupo são avaliados individualmente(*)
- . Grupos de “melhoria” são compostos só por alunos em “melhoria”
- . Cada projecto deve ser acompanhado, na altura da entrega, de um relatório onde se expliquem as principais decisões tomadas pela equipa e o funcionamento do programa desenvolvido
- . Caso sejam detectadas cópias nos projectos, os grupos envolvidos serão automaticamente reprovados.
- . (*) A equipa docente reserva-se o direito de não passar elementos do grupo que estejam a funcionar em “modo remoto”.



Próximos passos

- Inscrição nos turnos práticos: já feitos pela Direcção de Curso e gerido pela mesma.
- Garantir que o perfil de cada aluno no elearning tem uma fotografia perceptível e actualizada.
- Aulas práticas começam na semana 2 (13/2 a 17/2)
 - Devem instalar, nas vossas máquinas, Java SE SDK + BlueJ
- Página da disciplina:
 - **No Blackboard (<http://elearnig.uminho.pt>)**