

**Universidade do Minho**  
Escola de Engenharia  
Departamento de Informática

# Programação Orientada aos Objectos

LEI + LCC - 2º ano / 2º semestre

2023/2024

António Nestor Ribeiro  
anr@di.uminho.pt

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



# Escolaridade

- 2 T + 2PL

## Equipa Docente

- Aulas T
  - António Nestor Ribeiro ([anr@di.uminho.pt](mailto:anr@di.uminho.pt))
- Aulas PL
  - António Luís Sousa ([als@di.uminho.pt](mailto:als@di.uminho.pt)) (LEI: PL1 , LCC: PL1)
  - José Creissac Campos ([jose.campos@di.uminho.pt](mailto:jose.campos@di.uminho.pt)) (LCC: PL2)
  - Fábio Coelho ([fabio.a.coelho@inesctec.pt](mailto:fabio.a.coelho@inesctec.pt)) (LEI: PL6, PL7)
  - Cláudia Brito ([claudia.v.brito@inesctec.pt](mailto:claudia.v.brito@inesctec.pt)) (LEI: PL8, PL9 )
  - Susana Marques ([susana.v.marques@inesctec.pt](mailto:susana.v.marques@inesctec.pt)) (LEI: PL4, PL5 )
  - Francisco Maia ([fmaia@di.uminho.pt](mailto:fmaia@di.uminho.pt)) (LEI: PL2, LCC: PL3)
  - António Nestor Ribeiro ([anr@di.uminho.pt](mailto:anr@di.uminho.pt)) (LEI: 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.
  - Model-View-Controller
  - Padrões arquitecturais mais comuns
- Tecnologia de programação JAVA:
  - Plataforma Java: JDK, JVM e byte-code.
  - Construções básicas: tipos primitivos e operadores. Arrays.
  - Estruturas de controlo.
  - I/O básico.
  - Categorias dos objectos: Classes e instâncias.
  - Construtores.
  - Métodos e variáveis de instância.
  - Modificadores de acesso.
  - 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.

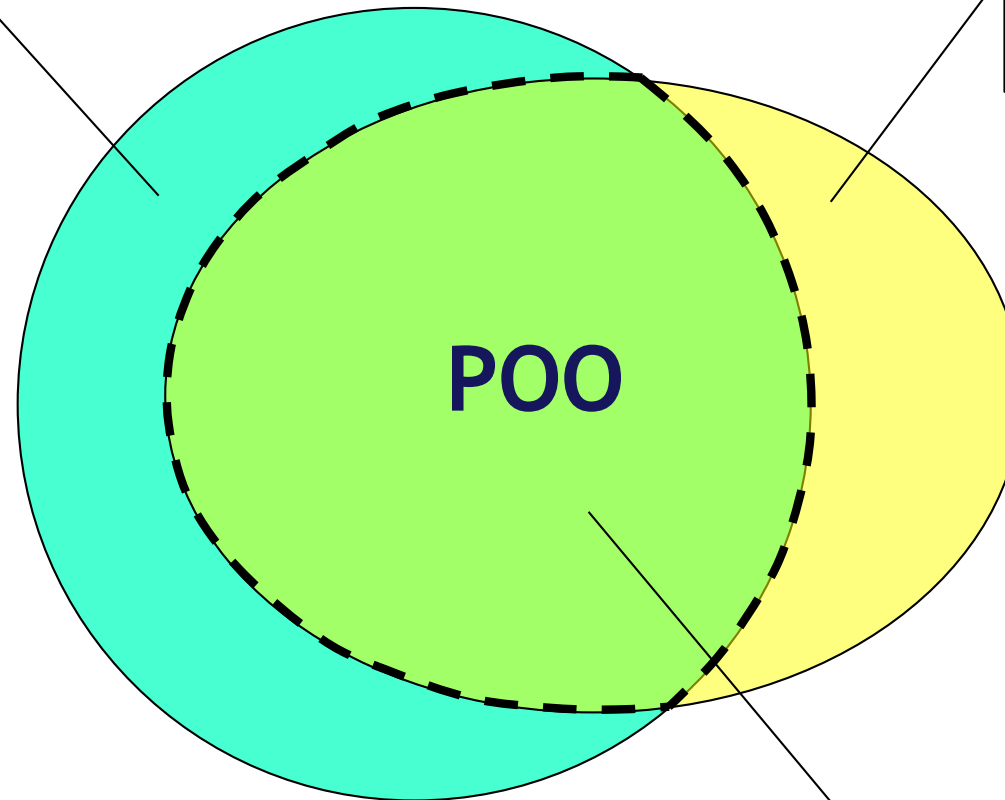


5

# 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 (em Java) de média escala, seguras, robustas e extensíveis.



## Bibliografia

### **Java 8 POO + Construções Funcionais**

F. Mário Martins, Editora FCA, Série Tecnologias de Informação, ISBN-978-972-722-838-6, 2017

### **Object Oriented Design with Applications**

Grady Booch, Third Edition, Addison-Wesley Professional, ISBN 978-020-189-551-3, 2007

### **Objects First with Java - A Practical Introduction using BlueJ,**

David J. Barnes & Michael Kölling, Sixth Edition, Pearson Education, ISBN 978-1-292-15904-1, 2016

### **Java Program Design - Principles, Polymorphism, and Patterns**

Edward Sciore, Apress Media, ISBN 978-1-4842-4142-4, 2019

### **Fundamentals of Java Programming**

Mitsunori Ogihara, Springer, ISBN 978-3-319-89490-4, 2018

### **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>

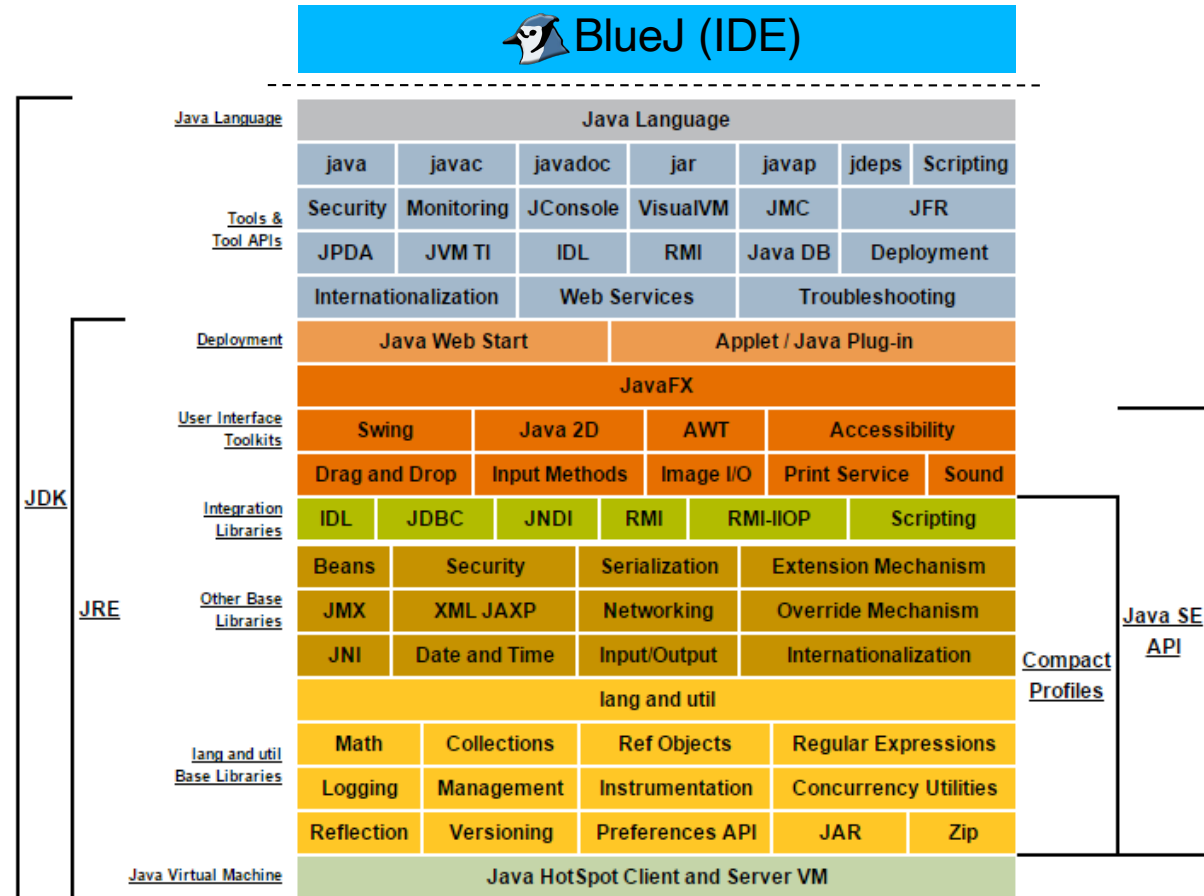
documentação e ambiente de edição, compilação e debug







# JDK (Java Development Kit) / BlueJ





## Avaliação

- . Teórica: 1 teste teórico com nota mínima de 9.0 valores  
Data do teste: 17.maio.2024  
Data do exame: 14.junho.2024
- . Prática: 1 projecto em grupo com nota mínima de 10.0 valores  
Data de entrega electrónica: 11.maio.2024  
Data da apresentação: 27 a 29.maio.2024
- . **Nota Final = 45%\*Prática + 55%\*Teórica**
- . **Nota Final  $\geq$  10.0**, para obter sucesso à UC de POO



## Métodos de Avaliação

- **alunos de primeira inscrição a POO**
  - Avaliação teórica + projecto
- **alunos com mais de uma inscrição a POO**
  - Podem utilizar a nota prática obtida em 2022/23
- A nota prática “congelada” ficará limitada a um máximo de 15 valores
- Alunos que queiram manter a nota de 2022/23 devem comunicar o pedido até ao fim do mês de Fevereiro (será criado um grupo no Blackboard). **Caso não o façam assume-se que estão novamente a realizar a componente prática.**



# Projecto

- . O enunciado será disponibilizado no dia 16 de março.
- . Os trabalhos são efectuados por grupos de **3 alunos**
- . Na altura da apresentação os elementos de cada grupo são avaliados individualmente.
- . Cada projecto deve ser acompanhado, na altura da entrega, de um pequeno 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.



## Próximos passos

- Inscrição nos turnos práticos: em LEI e LCC já operacionalizada pela Direcção de Curso e gerido pela mesma. A informação dada pelas direcções de curso serão carregadas para o elearning antes da primeira aula PL.
- Aulas práticas: começam na semana de 12/2 a 17/2
  - Devem instalar, nas vossas máquinas, Java SE SDK + IDE(BlueJ/IntelliJ/Netbeans/etc.)
- Página da disciplina:
  - **No Elearning (<http://elearnig.uminho.pt>)**
  - Canal privilegiado para acesso a recursos sobre a UC e meio privilegiado de comunicação com os alunos.
  - Nos contactos por email colocar no assunto (no início) **[POO]**