

## ^ Objectivos e Competências a Desenvolver

Editar

- Português:**
- Adquirir competências de programação em linguagem Java, consolidando os conhecimentos fundamentais de programação orientada a objetos adquiridos anteriormente.
  - Adquirir conhecimentos sobre *Software Design Patterns* no desenvolvimento de aplicações.
  - Adquirir a capacidade de desenvolver aplicações para ambiente consola e também para ambiente gráfico, baseado em eventos.
  - Desenvolver capacidades de organização das estruturas de dados, conseguindo aplicá-las adequadamente em aplicações de média dimensão, estruturando-as de uma forma extensível e racional, separando os dados da sua forma de representação na interação com o utilizador.
- Inglês:**
- Acquire programming skills in Java language, consolidating the fundamental knowledge of object-oriented programming previously obtained.
  - Acquire knowledge about Software Design Patterns in the development of applications.
  - Acquire the ability to develop applications for console and also for graphical environments, based on events.
  - Develop abilities in data structures organization, their suitable use in medium-sized applications, structuring them in an extensible and rational way, separating the data from its visual representation and user interaction.

## ^ Programa

Editar

- Português:**
- Principais características da linguagem Java
  - Transição de C/C++ para Java
    - Tipos primitivos
    - Arrays de tipos primitivos
    - Operadores
    - Instruções de controlo de execução
  - Classes
    - Objetos e referências
    - Garbage collector
    - Construtores
    - A palavra chave this
    - Construtores que chamam construtores
    - Membros static
    - Métodos overloaded
    - O método finalize()
    - Arrays de objetos
    - Classes interiores
  - Packages
    - Organização de componentes
    - Especificadores de acesso
  - Composição e herança
    - Composição
    - Herança
    - Escolher entre composição e derivação
    - A palavra-chave super
    - A palavra-chave final
  - Polimorfismo
    - Upcasting
    - A classe Object
    - Classes abstratas
    - Interfaces
  - Coleções
    - Principais Interfaces derivados de Collection
    - Algumas implementações concretas
  - Excepções

- Ficheiros
- Interface gráfica
  - Introdução ao JavaFX
  - Aplicações com interface gráfico
  - Programação baseada em eventos
  - Componentes, eventos e listeners
- Padrões de programação
  - MVC, MVP e MVVM
  - Máquinas de estados
  - Singleton
  - Fábrica de objetos
  - Command
  - Memento
  - Outros padrões

### **Inglês:**

- Main features of the Java language
- Transition from C/C++ to Java
  - Primitive types
  - Arrays of primitive types
  - Operators
  - Execution control instructions
- Classes
  - Objects and references
  - The Garbage Collector
  - Constructors
  - The keyword this
  - Constructors that call constructors
  - Static members
  - Overloaded methods
  - The finalize() method
  - Arrays of objects
  - Inner and nested classes
- Packages
  - Components organization
  - Access modifiers
- Composition and inheritance
  - Composition
  - Inheritance
  - Choose between composition and derivation
  - The super keyword
  - The final keyword
- Polymorphism
  - Upcasting
  - Object class
  - Abstract classes
  - Interfaces
- Collections
  - Interfaces derived from Collection
  - Some concrete implementations
- Exceptions
- Files
- Graphic interface
  - Introduction to JavaFX
  - Applications with graphical interface
  - Event-based programming
  - Components, events and listeners
- Software design patterns
  - MVC, MVP and MVVM
  - State machine
  - Singleton
  - Object Factory
  - Command
  - Memento
  - Other design patterns

## ^ Conhecimentos Base Recomendados

Editar

- Português:**
- Esta disciplina pressupõe conhecimentos lecionados na disciplina de Programação Orientada a Objetos.
- Inglês:**
- This course assumes that the students have the knowledge related to the Object Oriented Programming course.

## ^ Métodos de Ensino

Editar

- Português:**
- A unidade curricular possui aulas teóricas e práticas
    - Nas aulas teóricas é utilizada uma metodologia expositiva para a apresentação dos conceitos teóricos da matéria com exemplos de aplicação
    - Nas aulas práticas os alunos aplicam os conhecimentos adquiridos nas aulas teóricas e/ou aulas práticas e exercitam as competências desenvolvendo aplicações com a supervisão dos docentes.
  - Fora das aulas, os alunos desenvolvem, em grupos de dois, um projeto de média dimensão que é avaliado numa fase intermédia (segundo uma vertente mais qualitativa) e numa fase final (avaliação quantitativa). Os docentes acompanham o desenvolvimento do projeto, esclarecendo dúvidas e orientando quanto às opções tomadas.
- Inglês:**
- The curricular unit has theoretical and practical classes
    - In the theoretical classes, an expository methodology is used for the presentation of theoretical concepts of the subject with examples of application
    - In practical classes, students apply the knowledge acquired in theoretical classes and/or practical classes and exercise their skills by developing applications under teacher supervision.
  - Outside the classes, students develop a medium-sized project, in pairs, that is evaluated in an intermediate phase (according to a more qualitative evaluation) and in a final phase (quantitative evaluation). The teachers supervise the project development, clarifying doubts and providing guidance on the options taken.

## ^ Métodos de Avaliação

Editar

- Português:**
- A classificação final de um aluno resulta da média ponderada de uma componente escrita e de uma componente prática.
  - A componente escrita resulta da classificação obtida nas provas escritas realizadas nas épocas de exame disponíveis.
    - O peso desta componente na classificação final é de 40%.
  - A componente prática resulta da classificação obtida num trabalho prático.
    - A componente prática, no seu todo, tem um peso de 60% na classificação final.
    - O trabalho prático consiste no desenvolvimento de um problema proposto e é realizado em grupos de 2 elementos.
    - Este trabalho prático semestral é entregue em duas fases, durante o período letivo:
      - 1ª fase
        - Data limite de entrega: 2022.05.02 08h00
        - Através da avaliação desta fase será atribuído um fator multiplicativo sobre nota final do trabalho, após a defesa da 2ª fase. O fator multiplicativo apenas possui dois valores possíveis: 0.8 (caso não entregue ou os elementos entregues não satisfazem minimamente os objetivos definidos) e 1 (restantes casos).
        - 1/3 dos trabalhos serão selecionados para serem sujeitos a defesa.
      - 2ª fase
        - Data limite de entrega: 2022.06.20 08h00

- Penalização de 5% por cada hora de atraso na entrega
- Todos os trabalhos serão sujeitos a defesa.
- A nota final do trabalho será multiplicada pelo fator obtido na primeira fase.
  - A nota obtida no trabalho entregue é válida para todas as épocas do corrente ano letivo.
- Todas as componentes de avaliação têm mínimos de 35%, que serão reduzidos para 25% caso o aluno tenha mais de 2/3 de presenças, na turma onde está oficialmente inscrito.

#### **Inglês:**

- The final mark of a student results from the weighted average of a written component and a practical component.
- The written component results from the mark obtained in the written tests carried out in the available exam periods.
  - The weight of this component in the final mark is 40%.
- The practical component results from the mark obtained in a practical work.
  - The practical component has a weight of 60% in the final mark.
  - The practical work consists of the development of a proposed problem and is carried out in groups of 2 elements.
  - The practical work is delivered in two phases, during the academic semestre:
    - 1st phase
      - Deadline for delivery: 2022.05.02 08:00
      - A multiplicative factor will be assigned and applied in the final grade of the work, after the defense of the 2nd phase. This multiplicative factor only has two possible values: 0.8 (if not delivered or the delivered elements do not minimally satisfy the defined objectives) and 1.0 (remaining cases).
      - 1/3 of the works will be selected to be subject of defense.
    - 2nd phase
      - Deadline for delivery: 2022.06.20 08:00
      - Penalty of 5% for each hour of delay after the deadline.
      - All works will be subject to defense.
      - The final grade of the work will be multiplied by the factor obtained in the first phase.
  - The grade obtained in the work is valid for all official exams dates of the current academic year.
- All assessment components have a minimum of 35%, which will be reduced to 25% if the student has more than 2/3 of attendance, in the class where he/she is officially enrolled.

## ^ Bibliografia

Editar

#### **Português:**

- Bibliografia Principal:
  - Eck, D. (2020). Introduction to Programming Using Java (8th Edition). Disponível em [math.hws.edu/javanotes/](http://math.hws.edu/javanotes/)
  - Eckel, B. (2006). Thinking in Java (4th Edition). Prentice Hall. (Cota: 1A-1-448 (ISEC) – 14372).
  - Deitel, P., Deitel, H. (2007). Java How to Program (7th Edition). Prentice Hall. (Cota: 1A-1-146 (ISEC) - 14445).
  - Gamma, E., Helm, R., Johnson, R., Vlissides, J. (1995). Design Patterns: Elements of Reusable Object Oriented Software. Addison-Wesley. (Cota: 1A-7-97 (ISEC) - 15112).
  - Freeman, E., Freeman, E. (2004). Head First Design Patterns. O'Reilly Media. (Cota: 1A-7-75 (ISEC) - 14706).
  - Grand, M. (1998). Patterns in Java Volume 1 : a catalog of reusable design patterns illustrated with UML. John Wiley. (Cota: 1A-1-305 (ISEC) V.1o v. – 09760).
- Bibliografia Complementar:
  - Savitch, W. (2018). Java: An Introduction to Problem Solving and Programming (8th Edition). Pearson.
  - Horstmann C., Cornell, G. (2015). Core Java, Volume I – Fundamentals (11th Edition). Prentice-Hall.

**Inglês:**

- Main Bibliography:
  - Eck, D. (2020). Introduction to Programming Using Java (8th Edition). Available in math.hws.edu/javanotes/
  - Eckel, B. (2006). Thinking in Java (4th Edition). Prentice Hall. (Cota: 1A-1-448 (ISEC) – 14372).
  - Deitel, P., Deitel, H. (2007). Java How to Program (7th Edition). Prentice Hall. (Cota: 1A-1-146 (ISEC) - 14445).
  - Gamma, E., Helm, R., Johnson, R., Vlissides, J. (1995). Design Patterns: Elements of Reusable Object Oriented Software. Addison-Wesley. (Cota: 1A-7-97 (ISEC) - 15112).
  - Freeman, E., Freeman, E. (2004). Head First Design Patterns. O'Reilly Media. (Cota: 1A-7-75 (ISEC) - 14706).
  - Grand, M. (1998). Patterns in Java Volume 1 : a catalog of reusable design patterns illustrated with UML. John Wiley. (Cota: 1A-1-305 (ISEC) V.1o v. – 09760).
- Complementary Bibliography:
  - Savitch, W. (2018). Java: An Introduction to Problem Solving and Programming (8th Edition). Pearson.
  - Horstmann C., Cornell, G. (2015). Core Java, Volume I – Fundamentals (11th Edition). Prentice-Hall.

^ Atividades de promoção da ligação do estudante a  
atividades de desenvolvimento da investigação, transferência  
de tecnologia e conhecimento e inovação

[Editar](#)**Português:**

- Motivar a exploração de aspetos não lecionados que se integrem no desenvolvimento de trabalhos práticos.

**Inglês:**

- Motivate the exploration of aspects not taught that are integrated in the development of practical work.