

Course Unit	Algorithms and Data Structures		Field of study	Computer Science	
Bachelor in	Informatics Engineering		School	School of Technology and Management	
Academic Year	2023/2024	Year of study	2	Level	1-2
Type	Semestral	Semester	1	ECTS credits	6.0
Code	9119-706-2101-00-23				
Workload (hours)	162	Contact hours	T -	TP 60	PL -
			TC -	S -	E -
			OT -	O -	

T - Lectures; TP - Lectures and problem-solving; PL - Problem-solving, project or laboratory; TC - Fieldwork; S - Seminar; E - Placement; OT - Tutorial; O - Other

Name(s) of lecturer(s) Paulo Duarte Ferreira Gouveia, Adília Isabel Domingues Cruz Alves, Helder Francisco Silva Vieira, Jose Paulo Machado Da Costa

### Learning outcomes and competences

At the end of the course unit the learner is expected to be able to:

1. demonstrate a good domain of the Java programming language;
2. understand and use the Java collections standard library;
3. evaluate the efficiency of the algorithms;
4. design the data structures of computer applications;
5. implement data structures and algorithms for linked lists, stacks, queues, binary search trees and heaps;
6. develop and implement abstract data types, using an object oriented language;
7. define and implement iterators for different types of data structures;

### Prerequisites

Before the course unit the learner is expected to be able to:  
program in an object-oriented language.

### Course contents

The Java programming language; Java collections framework; analysis of algorithms; elementary rules of data structures design; the abstract data types (ADTs) stack, queue, deque, priority queue, map and decision tree; implementation of linear and nonlinear data structures that realize the ADTs (linked lists, binary search trees, heaps).

### Course contents (extended version)

1. The Java language
  - java compiler and technologies;
  - java versus C++;
  - value and reference types;
  - creating and using objects;
  - strings, wrappers, and arrays types;
  - simple input and output;
  - packages and imports;
  - javadoc documentation;
  - creating, testing and debugging.
2. POO in Java
  - abstract data types;
  - interfaces and abstract classes;
  - Object superclass and its main methods;
  - inheritance and constructors;
  - generics;
  - exceptions.
3. Java collections framework (JCF)
  - iterators;
  - Iterable and Iterator interfaces;
  - the classes and interfaces architecture of JCF;
  - Set, List, Queue, Deque and Map interfaces;
  - the ordination of the JCF concrete collections;
  - Comparable and Comparator interfaces.
4. Analysis of algorithms
  - algorithmic efficiency;
  - standard reference functions for algorithm analysis;
  - asymptotic algorithm analysis;
  - big-Oh notation.
5. Defining and implementing linear data structures
  - singly and doubly linked lists;
  - array-based stack;
  - linked list-based stack;
  - circular array-based queue;
  - linked list-based queue;
  - circular array-based deque;
  - doubly linked list-based deque;
  - sorted array-based map;
  - sorted doubly linked list-based map;
  - implementation of iterators.
6. Trees
  - defining and implementing binary trees;
  - tree traversal algorithms;
  - decision trees;
  - binary search trees;
  - AVL trees;
  - heap-based priority queue.

### Recommended reading

1. "Data Structures and Algorithms in Java - 6th edition", M. T. Goodrich, R. Tamassia, and M. H. Goldwasser, Wiley, 2014.
2. "Estruturas de Dados e Algoritmos em Java", António Adrego da Rocha, FCA, 2011.
3. "Projetos de POO em Java", F. Mário Martins, FCA, 2014.

**Recommended reading**

4. "Java6 e Programação Orientada pelos Objectos", F. Mário Martins, FCA, 2009.
5. "Thinking in Java - 4th Edition", Bruce Eckel, Prentice-Hall, 2006.

**Teaching and learning methods**

This course is composed by theoretical-practical lectures, divided into two kinds of periods: - expository periods during which the theoretical contents are presented and explained based on practical examples; - implementation periods during which the students put in practice the knowledge acquired in the expository periods. Non-presence periods are aimed to study and implement practical works.

**Assessment methods**

1. Alternative 1 - (Regular, Student Worker) (Final, Supplementary)
  - Practical Work - 60% (two practical works)
  - Final Written Exam - 40% (minimum grade of 5 points)
2. Alternative 2 - (Regular, Student Worker) (Special)
  - Final Written Exam - 100%

**Language of instruction**

1. Portuguese
2. English

**Electronic validation**

Paulo Duarte Ferreira Gouveia	Tiago Miguel Ferreira Guimaraes Pedrosa	Luisa Maria Garcia Jorge	José Carlos Rufino Amaro
04-10-2023	07-10-2023	16-10-2023	31-10-2023