

Java

TechGuide - Alura, FIAP e PM3

Java

Nível 1

☐ Java - Fundamentals:

- Java is a widely-used programming language for coding web applications. Java is a multi-platform, object-oriented, and network-centric language that can be used as a platform in itself. It is a fast, secure, reliable programming language for coding everything from mobile apps and enterprise software to big data applications and server-side technologies.
- Knowing the primitive types
- Declaring variables, considering the different types
- Using conditional structures ('if', 'else')
- Knowing the assignment and comparison operators
- Using repetition structures and loops ('while', 'for')
- Using functions, passing parameters and arguments
- Manipulating methods
- Manipulating arrays and lists
- Getting data from an API
- Making asynchronous 'Future' calls, etc
- Creating constructors

☐ Object-oriented Programming Concepts:

- Object-oriented programming (OOP) is a programming paradigm based on the concept of 'objects', which can contain data and code: data in the form of fields (often known as attributes or properties), and code, in the form of procedures (often known as methods). A common feature of objects is that procedures (or methods) are attached to them and can access and modify the object's data fields. Some of the main concepts are classes and instances, inheritance, and encapsulation.
- How objects work
- Creating and using constructors
- What classes are
- Creating and using Methods
- How encapsulation works
- What inheritance is
- What polymorphism is
- How interfaces work
- What abstractions are

☐ **Java - Error handling:**

- Error handling refers to the response and recovery procedures from error conditions present in a software application. In other words, it is the process comprised of anticipation, detection and resolution of application, programming or communication errors.
- Knowing and handling the most common exceptions
- Knowing the types of errors and in which situations they can occur
- Using 'try' and 'catch' for error handling
- Learning on what occasions and how to use `throw`
- Creating specific exceptions according to your application's needs

☐ **Java - Collections:**

- A collection represents a group of objects, known as its elements. They are like containers that group multiple items in a single unit. Some collections

allow duplicate elements and others do not. Some are ordered and others unordered.

- Learn the uses and differences between List, Set and Map
- Learn the uses and differences between Equals and hashCode
- Learn to work with ArrayList, LinkedList or Vector
- Wrapper classes

☐ **Java - Testing:**

- Software testing is the process of evaluating and verifying that a software product or application does what it is supposed to do. The benefits of testing include preventing bugs, reducing development costs and improving performance.
- Using unit tests
- Using integration testing
- Using behavioral testing
- Using mocks

☐ **Java - Packages:**

- A package in Java is used to group related classes, similarly to a folder in a file directory. Packages are used to avoid name conflicts and to write a better maintainable code.
- Using imports and organizing your code through packages
- Getting to know java.lang
- Understanding immutability and the String class
- Understanding the java.lang.Object class
- Getting to know java.io

☐ **Data Structures:**

- In the context of computers, the data structure is a specific way of storing and organizing data in the computer's memory so that these data can be easily retrieved and efficiently used when needed later.

- Knowing the main data structures (linked list, stack, queue, tree, etc)
- Implementing the main data structures

Nivel 2

☐ JVM:

- The Java Virtual Machine (JVM) is a program that loads and runs Java applications, converting the bytecodes into machine-executable code. The JVM is responsible for managing the applications as they run. Thanks to the Java Virtual Machine, programs written in Java can run on any hardware and software platform that has a version of the JVM, thus making these applications independent of the platform they run on.
- Entender como funciona a máquina virtual do Java

☐ Java - Memory management:

- In Java, memory management is the process of allocation and de-allocation of objects, called Memory management. Java does memory management automatically. Java uses an automatic memory management system called a garbage collector. Thus, we are not required to implement memory management logic in our application.
- Understanding how memory and its management work in Java
- Understanding how the Garbage Collector works

☐ Spring Framework:

- Spring is an open source framework for the Java platform. It is a non-intrusive framework, based on the Inversion of Control (IoC) and Dependency Injection design patterns. In Spring, the container is responsible for "instantiating" classes from a Java application and defining the dependencies between them through a configuration file in XML format, inferences from the framework, what is called auto-wiring, or even annotations on classes, methods and properties. This way, Spring enables low coupling between classes in an object-oriented application.
- Understanding the concept of Dependency Injection

- Understanding the MVC pattern
- Using Spring Data to manipulate data

☐ **Spring Boot:**

- Spring Boot is an open source Java-based framework used to create microservices with Spring Framework. It is used to build stand-alone and production ready Spring applications.
- Creating standalone Spring applications
- Using the built-in HTTP servers

☐ **Java build tools:**

- A build tool is a system that allows you to automate all the routine tasks of a project in an organized way that avoids the developer from having to waste time. In other words, adding a new library, performing tests, packaging and deploying, or even compatibility between different IDEs are tasks easily solved with a build tool.
- Getting to know the most popular Java build tools, such as Maven, Jenkins, Apache Ant, Gradle, etc., and how to use them

☐ **Java - Persistence:**

- The concept of "data persistence" refers to ensuring that the information inputted into an application will be stored in a medium where it can be retrieved consistently. In other words, they are permanent records that are not lost when the session is closed.
- Understanding JDBC and JPA
- Using frameworks such as Spring Data and Hibernate
- Communicating with a relational database
- Understanding the difference between EAGER and LAZY relationships
- Planning queries with join fetch
- Encapsulating access in a DAO
- Understanding how memory works in this situation

Nivel 3

☐ **Microservices architecture:**

- Microservices are an architectural approach in which software consists of small independent services that communicate with each other and are organized according to their business domains.
- Learning the concept of planned architecture for microservices
- Performing communication using APIs
- Improving the scalability of a system

☐ **Java - Concurrency:**

- Concurrency is a programming paradigm for building programs that make use of the simultaneous execution of several interactive computational tasks, which can be implemented as separate programs or as a set of threads created by a single program.
- Running tasks simultaneously
- Making tasks wait until a certain event occurs
- Understanding how memory works in this situation

☐ **Containers:**

- Containers are software packages that contain all the elements needed to run in any environment.
- Kubernetes (also known as k8s or "kube") is an open source container orchestration platform that automates many of the manual processes involved in deploying, managing, and scaling containerized applications.
- Isolating your software to run independently
- Deploying software in clusters
- Modularizing your system into smaller packages
- Getting to know the Docker platform
- Getting to know Kubernetes

☐ **Kafka:**

- Apache Kafka is a distributed data transmission platform that is capable of publishing, subscribing, storing, and processing real-time log streams. This platform is designed to process data streams from multiple sources and deliver them to multiple clients.
- Using Kafka for asynchronous communication
- Creating microservices with Kafka
- Creating producers and consumers
- Understanding how to use Kafka for parallelism and serialized execution
- Obtaining guarantees regarding the sending or delivery of messages

Habilidade Auxiliar: Infrastructure

☐ Git & GitHub - Fundamentals:

- Git is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency.
- GitHub is a hosting service for software development and version control using Git.
- Creating a repository
- Cloning a repository
- Committing, pushing and pulling to and from the repository
- Reversing a commit
- Creating branches and pull requests
- Handling merge and conflicts

☐ HTTP - Fundamentals:

- HTTP stands for Hyper Text Transfer Protocol. Communication between client computers and web servers is done by sending HTTP Requests and receiving HTTP Responses.
- Understanding the difference between HTTP verbs
- Testing requests and checking the status codes in the browser

- Learning how to make a HTTP request on the command line with WGET
- Downloading an image with WGET
- Performing a POST

☐ **JSON:**

- JSON stands for JavaScript Object Notation. It is a text format for storing and transporting data.
- Creating an object
- Transforming an object into a string
- Transforming a string into an object
- Manipulating an object

☐ **Command Line - Fundamentals:**

- CLI is a command line program that accepts text input to execute operating system functions.
- Knowing the most important commands

☐ **Cloud - Fundamentals:**

- Cloud, or cloud computing, is the distribution of computing services over the Internet using a pay-as-you-go pricing model. A cloud is composed of various computing resources, ranging from the computers themselves (or instances, in cloud terminology) to networks, storage, databases, and everything around them. In other words, everything that is normally needed to set up the equivalent of a server room, or even a complete data center, will be ready to use, configured, and run.
- Knowing the difference between IaaS, PaaS and SaaS
- Knowing the largest cloud providers
- Specializing in a specific provider of your choice

☐ **SQL - Fundamentals:**

- Structured Query Language (SQL) is a standardized programming language that is used to manage relational databases and perform various operations

on the data in them.

- Knowing the most common SQL commands
- Using SELECT to query a table
- Using INSERT to insert data into a table
- Using UPDATE to update a table
- Using DELETE to remove data from a table
- Using JOIN to connect data from multiple tables
- Knowing the clauses (FROM, ORDER BY, etc.)

Habilidade Auxiliar: Good practices

☐ **SOLID:**

- SOLID has five principles that are considered best practices in software development that help programmers write cleaner code by separating responsibilities, reducing docking, easing refactoring, and encouraging code reuse.

☐ **Clean Architecture:**

- Clean architecture is a way of developing software, such that just by looking at the source code of a program, you should be able to tell what the program does.

☐ **Design Patterns:**

- In software engineering, a Design Pattern is a general, reusable solution to a commonly occurring problem within a given context in software design. It is a description or template for how to solve a problem that can be used in many different situations. Design Patterns are formalized best practices that the programmer can use to solve common problems when designing an application or system.
- Getting familiarized with and applying the main Design Patterns

☐ **Clean Code:**

- Applying simple techniques that aim to make a code easier to write and read
- Refactoring your code to make it clearer

Domain-Driven Design (DDD) Concepts:

- Domain-Driven Design (DDD) is an approach to software design and development that is first informed by business requirements. The program components (objects, classes, arrays, etc.) indicate the industry, sector, or business domain in which the business operates.
- Modeling domains effectively
- Basing complex projects on domain models
- Getting to know the building blocks of DDD