Java from scratch

Technical Introduction

www.sdacademy.pl

Agenda

1. Why Java?
2. Components of Java Program
3. Am I prepared for the course?
4. FAQ
5. Q&A

Why Java?

Java

1. Great entry point
2. Easy to learn, hard to master
3. Object oriented
4. Gives all of the tools required to build big and efficient applications
5. **Java Program is not only about Java**

Components of Java Program

Common assumptions

1. **20% theory and 80% practice**
2. There are no less or more important modules
3. We open the doors, give required tools and show how to use them but everyone is responsible to practice individually
4. IT industry requires constant commitment and learning

Java Fundamentals

1. Basics of Java language, Java core.
2. Simple, small applications, „proof of concept”.
3. Many of the covered topics are applicable to other languages.
4. It’s not only about the theory but we write only simple, independent applications.

- **Must:**

  - None

- **Should:**

  - Introduction to Computer Science

System Git - video

1. Independent of the language.
2. The most important tool to control versioning of the application.
3. Used worldwide by mostly every software developer.

- **Must:**

  - None

- **Should:**

  - Java Fundamentals

Java Fundamentals: Coding

1. Uses theory from the Java Fundamentals block.
2. No new theory - but there is some time to refresh the theory or to fill the gaps.
3. We write more useful and bigger applications.

- **Must:**
  - Java Fundamentals
- **Should:**
  - Git

Software Testing - Fundamentals

1. How to write automated tests for our applications?
2. How to handle quality of the application and why is it so important?
3. Common practices of testing.

- **Must:**
  - Java Fundamentals
- **Should:**
  - Git

Java Advanced Features

1. Advanced features of Java language.
2. Must have if we would like to write bigger applications and to get our first job. Not required for writing basic applications.
3. Some of the topics are also applicable for other languages.

- **Must:**
  - Java Fundamentals
- **Should**:
  - None

Design Patterns & Good Practices

1. One of the hardest part of the course.
2. Not limited to the Java language.
3. Shows best approaches to common development problems.

- **Must:**
  - Java Advanced Features
- **Should:**
  - None

Java Advanced Features - Coding

1. Similar to the Java Fundamentals – Coding.

- **Must:**
  - Java Advanced Features
- **Should:**
  - Git
  - Software Testing - Fundamentals

Databases -SQL

1. Introduction of new language.
2. Not always required but good to know.
3. Shows how to store data properly, without excel files, using our applications.

- **Must:**
  - None
- **Should:**
  - None

JDBC & Hibernate

1. Two most important drivers to connect to the database from Java application.
2. Hibernate is top of the top for that, as we don't have to know SQL to communicate with the database.
3. Still JDBC is easier to handle for smaller applications.

- **Must:**
  - Java Advanced Features
  - Databases
- **Should:**
  - Software Testing - Fundamentals

Practical Project

1. Checks the knowledge of every part of the course.
2. Somehow it's a next step of Java Advanced Features – Coding but with databases and so on.

- **Must:**
  - All topics mentioned before
- **Should:** -

Introduction to HTTP - video

1. Shows how to communicate with web browser, web applications and between servers using common tools.
2. Show how does the request and response looks like, how to extract it and get useful part.
3. Additional block, nice to know, not so required.

- **Must:**
  - None
- **Should:**
  - None

HTML, CSS, JS

1. Basic information how to create our own webpage.
2. It won't look fancy at this moment.
3. Covers topics like colours, fonts, buttons and interaction with them.
4. HTML is about what do we see.
5. CSS is about how does is look.
6. JS is about how does it work.

- **Must:**
  - None
- **Should:**
  - Git
  - Introduction to HTTP

Frontend Technologies - Angular

1. Modern framework for frontend development.
2. The other ones with similar majority are: ReactJS and vue.js
3. Huge frontend applications are very hard to handle without something like this.

- **Must:**
  - Java Advanced Features
  - HTML, CSS, JS
- **Should:**
  - Git

Spring

1. The most important framework for backend development.
2. Gives the possibility to create interaction between our web and desktop part of application.
3. Absolutely must-have for Java Developers.

- **Must:**
  - Java Advanced Features
  - JDBC & Hibernate
  - Angular
- **Should:**
  - Git

Software Testing – Advanced Features

1. How to fake objects?
2. How to create parameterized tests?

- **Must:**
  - Software Testing - Fundamentals
  - Java Advanced Features
- **Should:**
  - Git

Agile & Scrum - video

1. Shows common methodologies and approaches to software development (and not only).

- **Must:**
  - None
- **Should:**
  - None

Final Project

1. Summary of everything that we have learned during the course.
2. Consists of every topic from the Java course.
3. During this block we create a complex application, like movie rent service, with database, web application and whole backend.

- **Must:**
  - All of the topics
- **Should:** -

Am I prepared for the course?

Do I have...?

1. Installed IntelliJ Idea?
2. Installed GIT?
3. Setup Java in PATH?
4. Prepared dedicated place (on my laptop) to store everything related to the course?
5. Verified access to the LMS?
6. Verified access to the SPOJ?
7. Everything that I need, to start the course?

FAQ

FAQ

1. Am I not too old to join the course?
2. Am I limited to Java after the course?
3. What job position should I consider after finishing the course?
4. What skillset is required to join the course?

Q&A

Thank you for your attention!