

Quick recap

The meeting focused on build systems in software development, with a particular emphasis on Bazel, a tool developed by Google and used by major tech companies. The instructor demonstrated how to set up and use Bazel for Java projects, explaining its advantages over other build tools like Maven and Gradle, especially for large-scale development environments. The session also covered class expectations, package structures, and the importance of understanding build systems for software engineering, with future sessions planned to explore Gradle and other related topics.

Next steps

- [Students to revise the content of the session and not make notes during class.](#)
- [Students to rewatch the recording of the session.](#)
- [Students to attempt any assignments or follow-up tasks after the class.](#)
- [Students to explore why the package signature "sample Java project dot example" was not working in the demo.](#)
- [Students to try setting up a project using Bazel themselves.](#)
- [Instructor to upload the code from the session to the shared repository.](#)
- [Instructor to send Discord links to students who registered after 4 PM.](#)
- [Students to check their spam folders for Discord links if not received.](#)
- [Instructor to cover Gradle in the next session.](#)

Summary

Class Expectations and Build Systems

AlgoCamp explained the class expectations, emphasizing attentive listening without note-taking during lectures and encouraging students to revise content later. He outlined the availability of class recordings and the use of tools like IntelliJ, while discussing the importance of build systems in software development. AlgoCamp conducted a poll to assess students' familiarity with Java and multi-file projects, and he demonstrated a terminal exercise to illustrate the concept of build systems.

Java Multi-File Project Structure

AlgoCamp demonstrated how to create and run a simple Java project using multiple files, highlighting the importance of package structure and compilation paths. He explained that when working with multi-file projects, it's crucial to ensure all relevant files are visible to each other,

either by compiling from the correct folder or specifying the full path to the package. AlgoCamp also emphasized the significance of understanding package visibility and how it relates to folder structure in Java projects.

Efficient Build Tools for Developers

AlgoCamp discussed the importance of proper compilation and dependency management in large software projects. They explained how build tools help manage dependencies and compile projects efficiently, even when unused files exist. The discussion touched on various build tools across different programming ecosystems, such as Maven for Java, NPM for Node.js, and Pip for Python. AlgoCamp emphasized that developers should focus on writing logic rather than managing compilation details, and highlighted the concept of raising small pull requests or diffs in companies like Meta and Uber to facilitate easier code review and integration.

Bazel: A Fast and Correct Build Tool

AlgoCamp introduced Bazel, a build tool developed by Google and used by major tech companies like Uber and Meta, emphasizing its speed and multilingual capabilities. He explained that Bazel is not only fast but also correct, making it an ideal tool for large-scale development environments. AlgoCamp highlighted the importance of understanding build tools at a fundamental level and mentioned that while Gradle will be the primary build tool in the course, exploring Bazel will provide valuable insights into emerging technologies used in big tech. He also mentioned that there will be a homework assignment related to package signatures and paths.

Bazel Project Setup Overview

AlgoCamp led a session on creating a basic Bazel project, emphasizing its simplicity and ease of onboarding. They demonstrated setting up a Java project with a nested folder structure, including a main class and a utility class, and explained the purpose of package definitions. AlgoCamp highlighted the importance of not relying on IDE capabilities for development to better understand the underlying processes. They also mentioned that they would discuss the reasoning behind specific folder structures and naming conventions in future sessions.

Bazel Setup and Basics on Mac

The instructor demonstrated how to set up Bazel, a build tool, on a Mac using Homebrew. They explained the basics of creating a workspace file and a build file to define projects as Java libraries or executables. The instructor showed how to configure Bazel to manage dependencies and compile projects, highlighting the differences between libraries and binaries.

They also touched on Bazel's support for multiple languages and its simplicity compared to other build systems.

Build Systems Comparison and Best Practices

The team discussed build systems, focusing on Bazel, Maven, and Gradle. They explored how Bazel provides explicit configuration for dependencies and executables, making it more readable than Maven or Gradle. The group also covered the use of workspace files for managing external libraries and discussed the advantages of using a single build system across projects, with examples from Google and Uber. They addressed questions about package naming conventions, build file requirements, and the potential for migrating legacy projects to new build tools.

Bazel Build Tool Demonstration

The instructor demonstrated Bazel, a build tool, by showing how to compile and run Java projects. They explained Bazel's syntax, including how to specify project dependencies and visibility settings. The instructor addressed questions about migrating from other build tools and the tool's efficiency. They also discussed Bazel's adoption by major tech companies and its use in various programming ecosystems.

Bazel Build System Overview

The session focused on Bazel, a build system that simplifies managing complex projects with features like handling circular dependencies and supporting multi-file projects. AlgoCamp explained that Bazel compiles everything at build time, unlike Java class loaders, and discussed its advantages over Gradle, particularly for large monorepos like Google's. The instructor emphasized the importance of understanding build systems for software engineering and shared a repository containing code examples from the session. The next session will cover Gradle, and participants were encouraged to explore Bazel further and engage in discussions on Discord.