



Introduction to PV168

Course Information

1. Seminar groups
2. Project
3. Workshops for team leads
4. Lectures (optional)

Seminar Groups

- Methods of experiential learning
 - Try to do stuff by yourself
 - Then learn how to do it properly
- Pair programming

Project

- Teams of four with one team lead
- Assigned by project manager (PM)
- Teacher in the role of a tech lead
 - Guarantees the quality
 - **Solves unpleasant situations**
- Details at courses [pages](#).

Project Schedule

1. Initial GUI
2. The basics of the business logic
3. The persistence
 - Business logic completion
4. Non-blocking GUI
 - Finish & retrospection
5. Final presentation

Team Lead

- Communicates with both PM and Tech lead.
- Organizes his/her team and their.
- **Does less programming**
 - Organization takes time as well
- Attends extra workshops

Workshops

- After the first milestone
 - Lead's responsibilities
 - Options & Decisions
 - Ways of leading
- After the second milestone
 - Sharing with other team leads

Lectures

- Optional but recommended
- Explanation of work at seminars
- Place for discussions



Git Culture

Git Commits

- Commits should be descriptive
- Commits should deal with single change.
- Commit should always pass tests.

Git Commits

DBZ-5606 Include storage modules into the build for DB2 workflow

 vjuraneq authored and jpechane committed yesterday ✓

DBZ-5043 Add topic prefix to Mongo config definition

 vjuraneq authored and jpechane committed yesterday ✓

DBZ-5043 Make topic prefix field mandatory and with proper group

 vjuraneq authored and jpechane committed yesterday

○ Commits on Sep 13, 2022

DBZ-5592 Include COMPUTE_CLAUSE in PHYSICAL_ATTRIBUTE_CLAUSE

 ani-sha authored and jpechane committed 2 days ago ✓

DBZ-5592 Allow LIST_VALUE_CLAUSE to use TIMESTAMP LITERAL

 ani-sha authored and jpechane committed 2 days ago

Feature Branches

- Work according to [Gitlab Flow](#)
- Main branch is ready for release
- Work in feature branches
- Utilize Pull/Merge request
- Test every commit

Merge Requests & Code Reviews

- Nothing goes directly to master
- Don't review more that ~500 LOC
- Have at least 2 team member approve each MR
- Have a discussion if something is not crystal clear



Apache Maven

Distribution

Intermezzo

What does it take to build,
distribute & run java application?

What are the distribution options?



What is Maven

- From Yiddish word meaning "Expert"
- Originally developed to simplify build of Jakarta Turbine
- A tool used to build and manage Java-based projects
- Currently in version 3.9.x
- [Apache Maven](#)

Maven's Goals

1. Making the build process easier
2. Providing a uniform build system
3. Providing quality project information
4. Encouraging better development practices

Maven What?

1. Describes how to build your project via POM file descriptor
2. Allows you to define useful metadata about your project
3. Provides dependency management
4. There are reasonable defaults

Project Object Model

- [Project descriptor](#) named `pom.xml`
- Each project identified by GAV

```
<project>
  <modelVersion>4.0.0</modelVersion>

  <groupId>cz.muni.fi.pv168</groupId>
  <artifactId>zinger</artifactId>
  <version>1.0-SNAPSHOT</version>
</project>
```

Project structure

```
project
|- src
    |- main
        |- java
        |- ressources
    |- test
        |- java
        |- ressources
|- pom.xml
```

Maven Dependencies

- Maven projects can be published to repositories
- Default remote repository is [Maven Central](#)
- Ability to [use multiple repositories](#)
- Local repository in `${M2_HOME}/repository` typically `~/.m2/repository` on Unix systems

POM Example

Let's look at full [pom.xml](#)

- What dependencies does this project have?
- What is the build output?
- What metadata does the project provides?

Maven Alternatives

"I don't like it, what are the other options?"

- Though luck, you need it!
- There is one, [Gradle](#)
- You still need to know Maven!