

PV 168
Seminar 08

Agenda

- Project: Second Milestone
- Seminar task
- Seminar task reflection

Project: Second Milestone

- Implement the application in the scope of MVP
 - Including data persistence and business logic
 - Use Apache Derby in embedded mode for storing data
 - Fix all issues from the First Milestone
- **Deadline: 4. 12. 2020 23:59**
 - The **master** branch of your repository at <https://gitlab.fi.muni.cz> contains complete MVP
- If you have any questions, ask assigned seminar tutor (the one who has customer role for your project)

Seminar task

- You will be now split to teams consisting of four students
- Create and push the branch (see the next slides)
- Work on the tasks in the specified order (see the next slides)
- You have **60 minutes** to solve all the tasks
- If you need any help, **Ask for Help in Zoom**

Working with branches in Git

- Clone the project <https://gitlab.fi.muni.cz/pv168/employee-evidence>
- Create new local branch
 - based on commit **a9ff78cb** (current head of master branch)
 - with name **week08-group0X-roomY** (0X is seminar group, Y is breakout room number)
 - If you from group PV168/01 and breakout room 3, branch name is week08-group01-room3
- Pavel Hrdina or Petr Adámek will give you write access to employee-evidence repository
 - Only one person per breakout room - the one who will use the computer for coding
 - Petr or Pavel will visit your breakout room to ask who is this person
 - In the meantime work on other tasks
- Push the branch to origin repository

Seminar task

- Fix **EmployeeDao** so that all tests are passing (including the disabled ones)
 - #1 EmployeeDaoTest.createEmployeeWithExistingId()
 - #2 EmployeeDaoTest.delete()
 - #3 EmployeeDaoTest.deleteWithNullId()
 - #4 EmployeeDaoTest.deleteNonExisting()
 - #5 EmployeeDaoTest.update()
 - #6 EmployeeDaoTest.updateWithNullId()
 - #7 EmployeeDaoTest.updateNonExisting()
- Add Non-binary gender (https://en.wikipedia.org/wiki/Non-binary_gender)
 - Add enum constant
 - Modify **EmployeeDaoTest** to help verify that the new constant is handled correctly
 - Modify **EmployeeDao** to handle the new enum constant correctly

Seminar task

- Implement the **Department** class
 - Each department has a name (e.g. IT) and a number (e.g. 007)
 - Just fields are sufficient, no need to implement get/set/toString methods
- Update **Employee** class to include **Department**
 - Each employee is a member of a single department
- Prepare DDL for creating tables
 - **Department** (create a new one)
 - **Employee** (update the existing one)
- If you still have time, try to
 - Update **EmployeeDao** to handle **department** attribute
 - Create **DepartmentDao**

Link to slides

<https://is.muni.cz/auth/el/fi/podzim2020/PV168/um/seminare/PV168-seminar-08.pdf>

Conclusion

Any questions?