PV 168
Seminar 09

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 **63d0c48f** (current head of master branch)
  - with name **week09-group0X-roomY** (0X is seminar group, Y is breakout room number)
  - If you from group PV168/01 and breakout room 3, branch name is week09-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

1. Run the tests with Coverage
    - Analyze which parts of DAO classes are not covered with tests.
    - Create a proposal on how to implement tests to increase line/statement coverage of these classes (you don’t need to implement them).
2. Implement these methods using TDD (write a test, then production code)
    - DepartmentDao.finaAll()
    - DepartmentDao.delete(...)
3. Uncomment line marked as **TASK #3** in DepartmentDao and run the tests
    - This will uncover bug which is somewhere in DAO classes. Find and fix the bug.
    - What are the reasons why this bug happened?
    - What could be done to avoid this kind of bugs?
4. How many SQL statements is sent to DB by method EmployeeDAO.findAll()?
5. Find and resolve duplicates introduced by adding Department entity

Link to slides

https://is.muni.cz/auth/el/fi/podzim2020/PV168/um/seminare/PV168-seminar-09.pdf

Conclusion

Any questions?