

PV 168  
Seminar 12

# Agenda

- Project: Next Steps
- Project: Third Milestone
- Seminar task
- Seminar task reflection

# Project: Next Steps

- Fix all issues found during the Second Milestone review
  - Problems in the GUI (usability)
  - Code quality
- Cover your code with tests (to enable further refactoring)
  - Everything but GUI
- Make sure that all DB operations are performed out of *Event Dispatcher Thread*
- Translate your application into two languages
  - English (as default)
  - Czech/Slovak/German/French/Russian/Filipino/Cebuano

# Project: Third Milestone

- Implement the fully functional application
  - In the scope agreed with the customer
- Deadline: **8. 1. 2021 23:59**
  - The **master** branch of your repository at <https://gitlab.fi.muni.cz> contains complete application
- 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 **71865b6f** (current head of master branch)
  - with name **week12-group0X-roomY** (0X is seminar group, Y is breakout room number)
  - If you from group PV168/01 and breakout room 3, branch name is week12-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. Add columns **ID**, **Number** and **Name** to `DepartmentTableModel`.
2. Implement class `DepartmentTab` and use it in `MainWindow`.
  - Look at `EmployeeTab` for inspiration.
  - Add / Edit operations will not work yet, because you don't have `DepartmentDialog`
  - Do not try to implement/extend `TestGenerator` for generating test `Department`. Just create new department instance with empty name and number.
  - Run the application and test the result.
3. Implement class `DepartmentDialog` and use it in `DepartmentTab`.
  - Look at `EmployeeDialog` for inspiration.
  - Add / Edit operations should work now.
  - Run the application and test the result.

# Seminar task

4. Run the application and try to delete department which has some employees.

Then discuss and answer these questions:

- What happened and why?
- Is this behaviour OK?
- Is there any way how to improve it?

5. Look at the commit with message “*Department ComboBox added to EmployeeDialog*”. Then discuss and answer these questions:

- What functionality was introduced by this commit?
- Which classes were added to bring this functionality?
- What is the purpose of these classes?
- Do you understand their code?



# Link to slides

<https://is.muni.cz/auth/el/fi/podzim2020/PV168/um/seminare/PV168-seminar-12.pdf>

# Conclusion

Any questions?