

PV 168  
Seminar 07

# 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 **9069e722** (current head of master branch)
  - with name **week07-group0X-roomY** (0X is seminar group, Y is breakout room number)
  - If you from group PV168/**01** and breakout room **3**, branch name is week07-group**01**-room**3**
- 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

- Add these columns into the table as read only columns
  - **Birth Date**
  - **Age**
- Change the order of columns in table model to the following (as another commit):
  - **Age, Last Name, First Name, Birth Date**
- Replace conditions in Column.java on lines 42 and 50 with polymorphism
- Modify EmployeeDao to support also gender attribute and add editable **Gender** column to **EmployeeTableModel** as the first column
- Implement method **EmployeeDao.delete(...)**

# Some hints & Link to slides

How to calculate employee age (two possible ways):

- `ChronoUnit.YEARS.between(employee.getBirthDate(), LocalDate.now());`
- `Period.between(employee.getBirthDate(), LocalDate.now()).getYears();`

How to define subclass of class with generic parameters

```
class SubClass<T, E> extends SuperClass<T, E> { ... }
```

<https://is.muni.cz/auth/el/fi/podzim2020/PV168/um/seminare/PV168-seminar-07.pdf>

# Conclusion

Any questions?