* Controller
  + sessionスコープ
    - private HttpSession session;
  + applicationスコープ
    - private ServletContext application;
  + formからdomainへ
    - BeanUtils.copyProperties(form, domain);
  + 登録などの処理はリダイレクト
* Form
  + ＠NotBlank(message=”名前は必須です”)
  + ＠Size(min=1, max=127, message=”Eメールは...です”)
  + ＠NotEmpty(message=”趣味は..さい”)(チェックボックス等に使用)
* Domain
* Service
* Repository
  + templateのDI
  + @Autowired
  + private NamedParameterJdbcTemplate template;
  + RowMapperの実装
  + private static final RowMapper<Employee> *EMPLOYEE\_ROW\_MAPPER* = (rs, i) -> {
  + Employee employee = new Employee();
  + employee.setId(rs.getInt("id"));
  + employee.setName(rs.getString("name"));
  + employee.setAge(rs.getInt("age"));
  + employee.setGender(rs.getString("gender"));
  + employee.setDepartmentId(rs.getInt("department\_id"));
  + return employee;
  + };
* HTML(Thymeleaf)
  + <span th:if=”${car.speed <= 60}”>法定速度です</span>
  + 入力した内容が残った状態で画面に戻る処理
    - <form th:object=”${recieveForm}”>
      * <input type=”text” name=”age” th:field=”\*{age}”> (th:text=”\*{age}”とかもある)
    - </form>
  + aタグにパラメータを埋め込む
    - <a th:href=”@{/controller-path/method-path?id=} + ${employee.id}”>リンク</a>
  + LocalDateクラスのformat
    - <span th:text=”${#temporals.format(localDate, ‘yyyy年MM月dd日’)}”></span>
  + Integerオブジェクトのフォーマット
    - <span th:text=”${#numbers.formatInteger(num(変数),1(最低桁数), ‘COMMA’)}”></span> (doubleオブジェクトは「#numbers.formatDecimal」)
  + 入力値エラー
    - <form th:object=”${userForm}”>
      * <div th:errors=”\*{name}” style=”color:red”>モック</div>
      * <input type=”text” name=”age” th:field=”\*{age}”>
    - </form>