

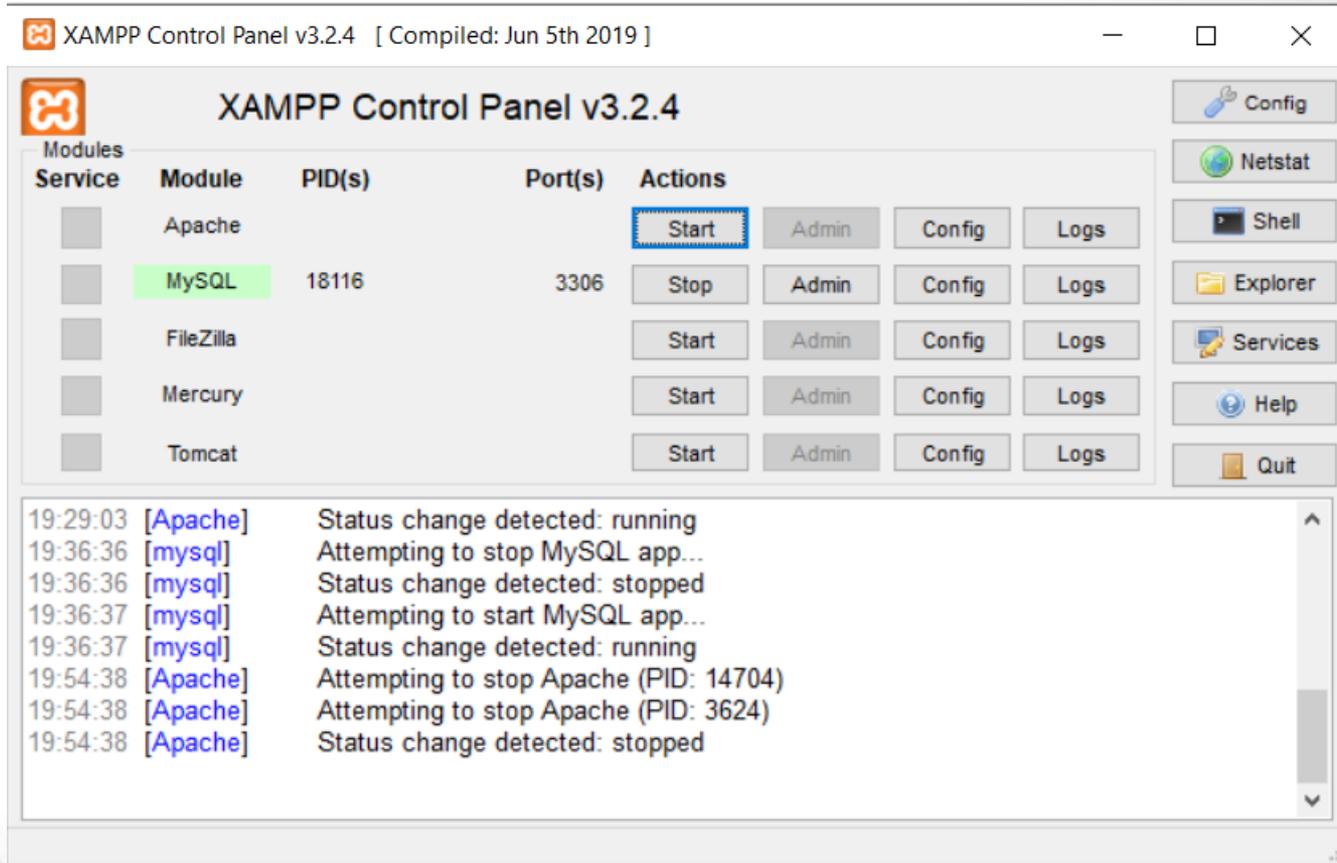
JavaFX + JPA + MySQL (MariaDB)

Использование Java Persistence для работы со связанными таблицами в БД

Евгений Беркунский, НУК
eugeny.berkunsky@gmail.com
<http://berkut.mk.ua>



Сначала запустить MariaDB



XAMPP Control Panel v3.2.4 [Compiled: Jun 5th 2019]

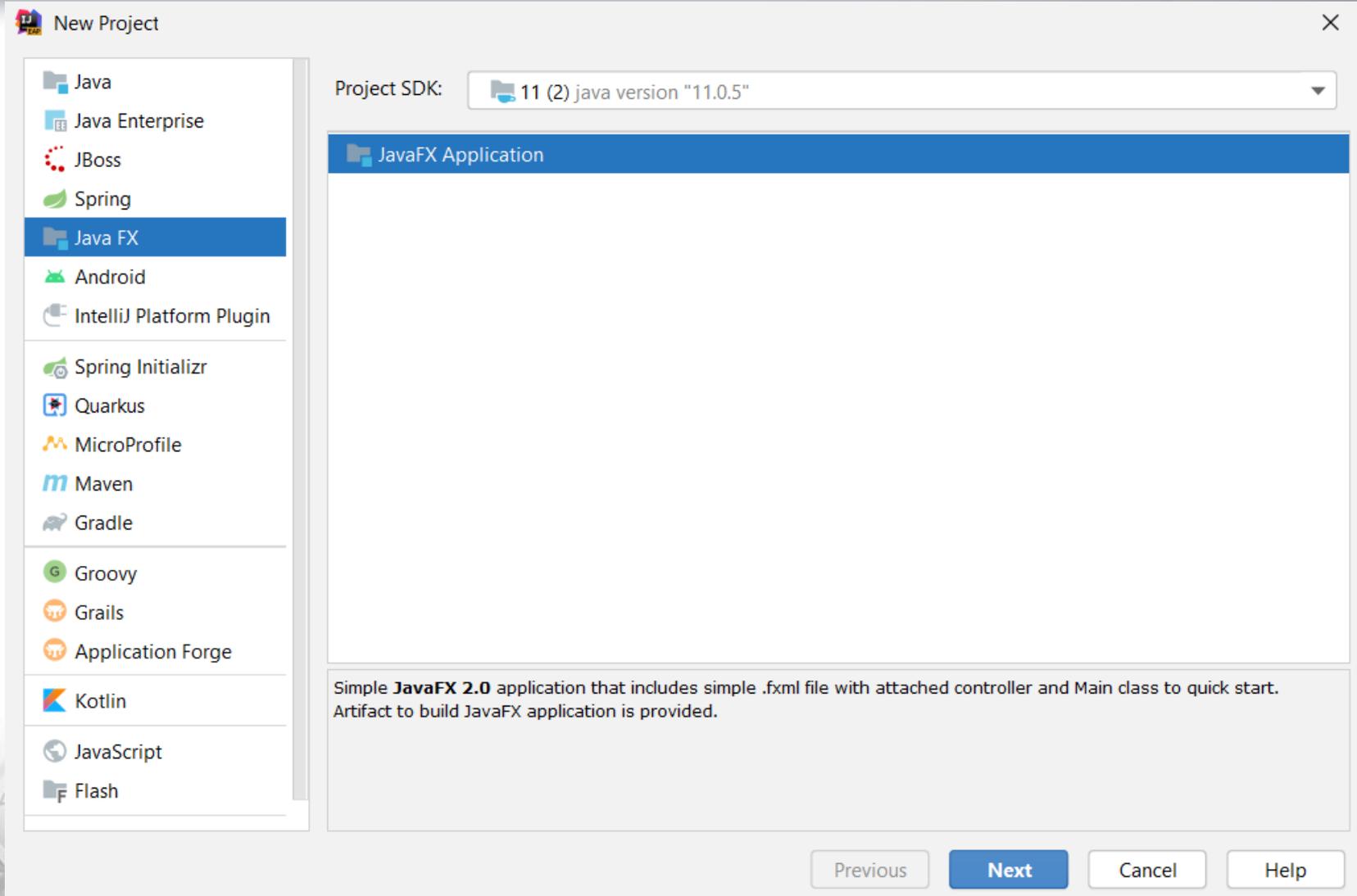
XAMPP Control Panel v3.2.4

Service	Module	PID(s)	Port(s)	Actions
<input type="checkbox"/>	Apache			<input type="button" value="Start"/> Admin Config Logs
<input checked="" type="checkbox"/>	MySQL	18116	3306	Stop Admin Config Logs
<input type="checkbox"/>	FileZilla			Start Admin Config Logs
<input type="checkbox"/>	Mercury			Start Admin Config Logs
<input type="checkbox"/>	Tomcat			Start Admin Config Logs

19:29:03 [Apache] Status change detected: running
19:36:36 [mysql] Attempting to stop MySQL app...
19:36:36 [mysql] Status change detected: stopped
19:36:37 [mysql] Attempting to start MySQL app...
19:36:37 [mysql] Status change detected: running
19:54:38 [Apache] Attempting to stop Apache (PID: 14704)
19:54:38 [Apache] Attempting to stop Apache (PID: 3624)
19:54:38 [Apache] Status change detected: stopped

На сервере должен быть зарегистрирован *пользователь student с паролем 123*

Создаем приложение JavaFX



The screenshot shows the 'New Project' dialog in IntelliJ IDEA. On the left, a list of project templates is shown, with 'Java FX' selected and highlighted in blue. The 'Project SDK' dropdown is set to '11 (2) java version "11.0.5"'. The 'JavaFX Application' template is selected in the main list. Below the list, a description reads: 'Simple **JavaFX 2.0** application that includes simple .fxml file with attached controller and Main class to quick start. Artifact to build JavaFX application is provided.' At the bottom, there are four buttons: 'Previous', 'Next' (highlighted in blue), 'Cancel', and 'Help'.

New Project

Project SDK: 11 (2) java version "11.0.5"

JavaFX Application

Simple **JavaFX 2.0** application that includes simple .fxml file with attached controller and Main class to quick start. Artifact to build JavaFX application is provided.

Previous Next Cancel Help

Реорганізуємо структуру пакетів

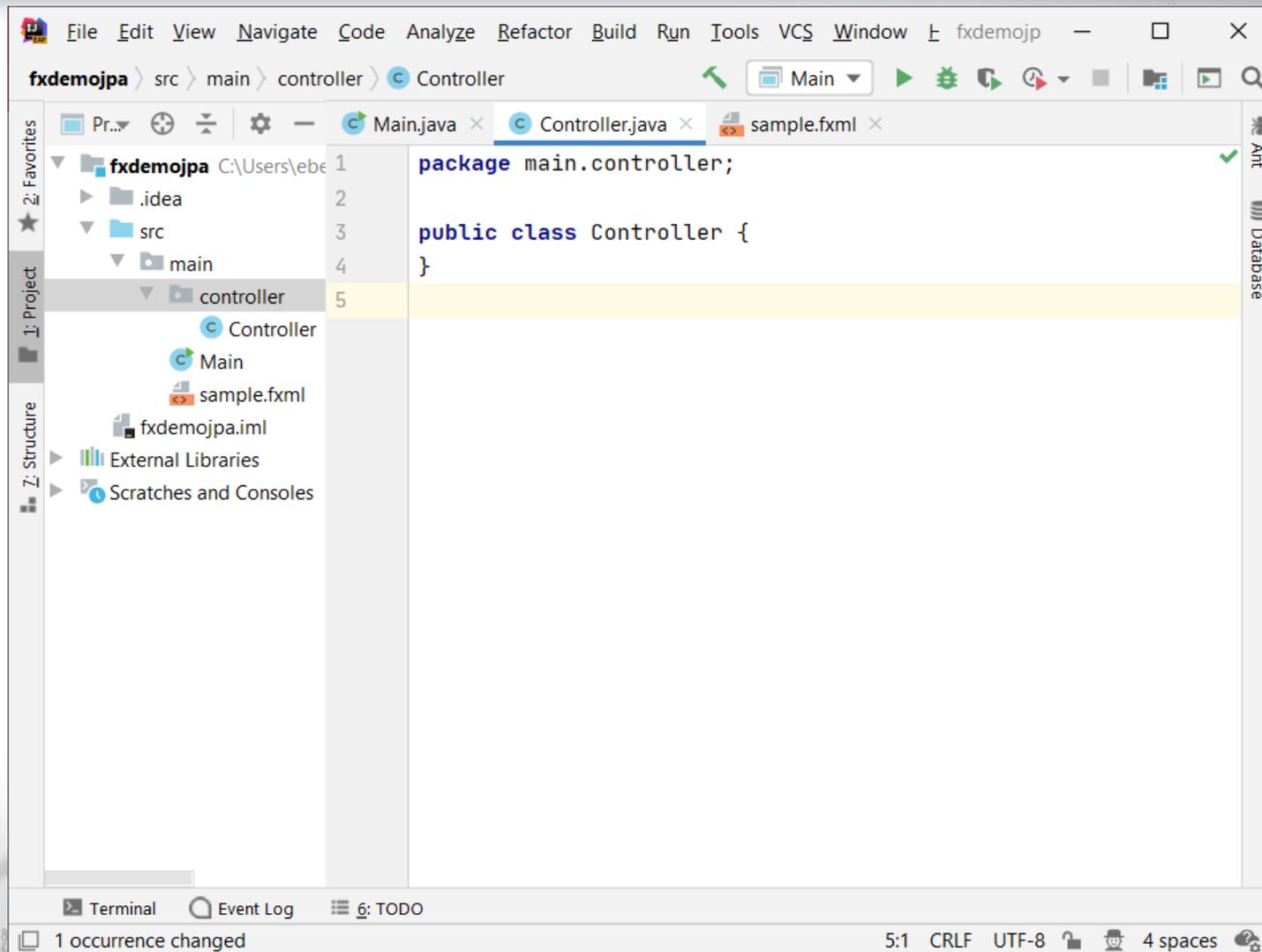
The screenshot shows the IntelliJ IDEA IDE interface. The main editor displays the following code in `Controller.java`:

```
1 package sample;  
2  
3 public class Controller {  
4 }  
5
```

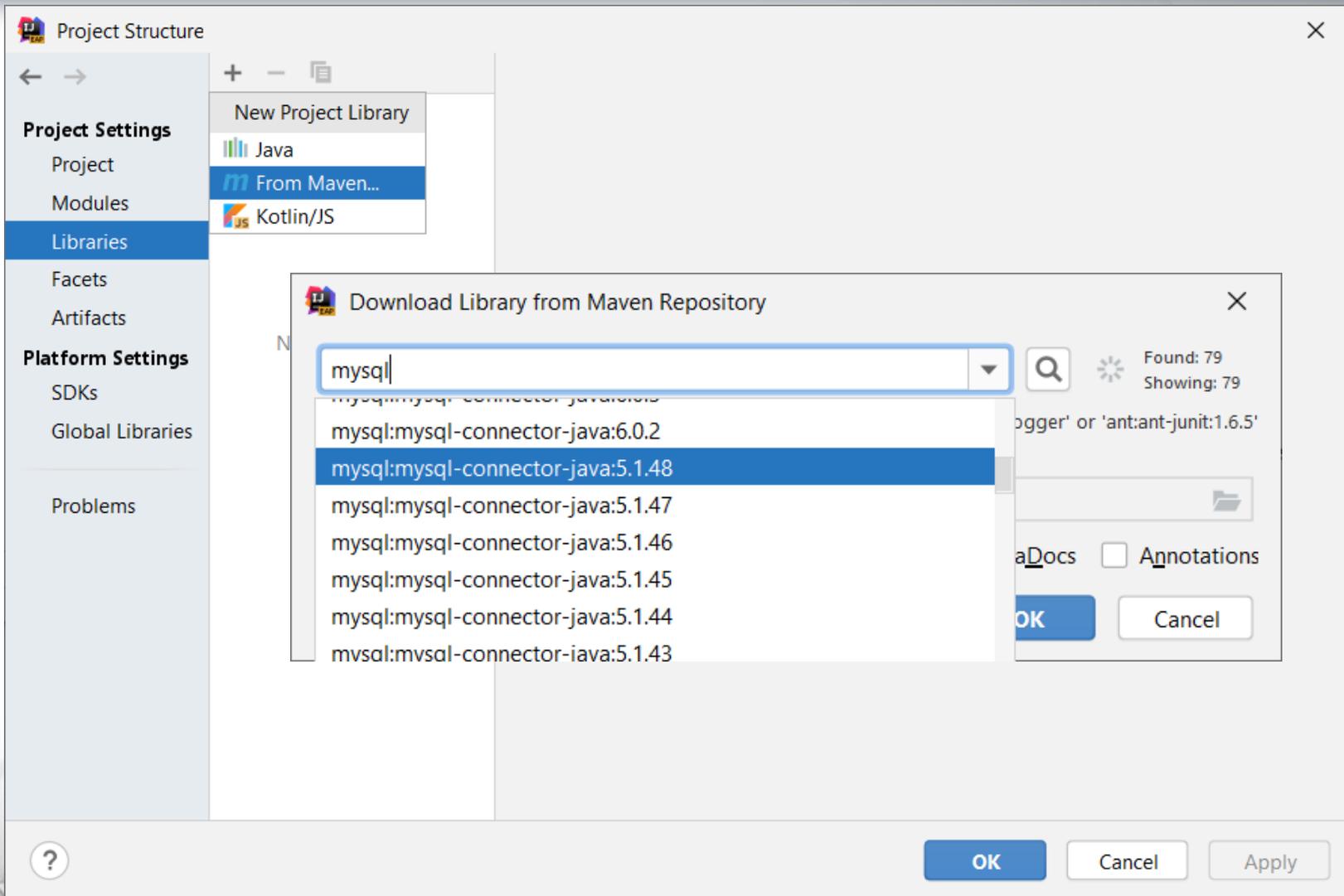
The `sample` package is highlighted in the Project Structure view on the left. A **Rename** dialog box is open in the foreground, with the following details:

- Title:** Rename
- Message:** Rename package 'sample' and its usages to:
- Input field:** Contains the text `main`.
- Options:**
 - Search in comments and strings
 - Search for text occurrences
- Scope:** A dropdown menu showing `Project Files`.
- Buttons:** **Refactor** (highlighted in blue), **Preview**, and **Cancel**.

Переместим контроллер в отдельный пакет



Подключаем библиотеки: 1 - MySQL



The image shows the 'Project Structure' dialog in IntelliJ IDEA, with the 'Libraries' tab selected. A 'New Project Library' menu is open, showing options for 'Java', 'From Maven...', and 'Kotlin/JS'. The 'From Maven...' option is selected, which has opened the 'Download Library from Maven Repository' dialog. In this dialog, the search term 'mysql' is entered in the search bar. The search results list several versions of the 'mysql:mysql-connector-java' artifact, with 'mysql:mysql-connector-java:5.1.48' highlighted. The dialog also shows 'Found: 79' and 'Showing: 79' results, and buttons for 'OK', 'Cancel', and 'Apply'.

Project Structure

- Project Settings
 - Project
 - Modules
 - Libraries**
 - Facets
 - Artifacts
- Platform Settings
 - SDKs
 - Global Libraries
- Problems

New Project Library

- Java
- From Maven...**
- Kotlin/JS

Download Library from Maven Repository

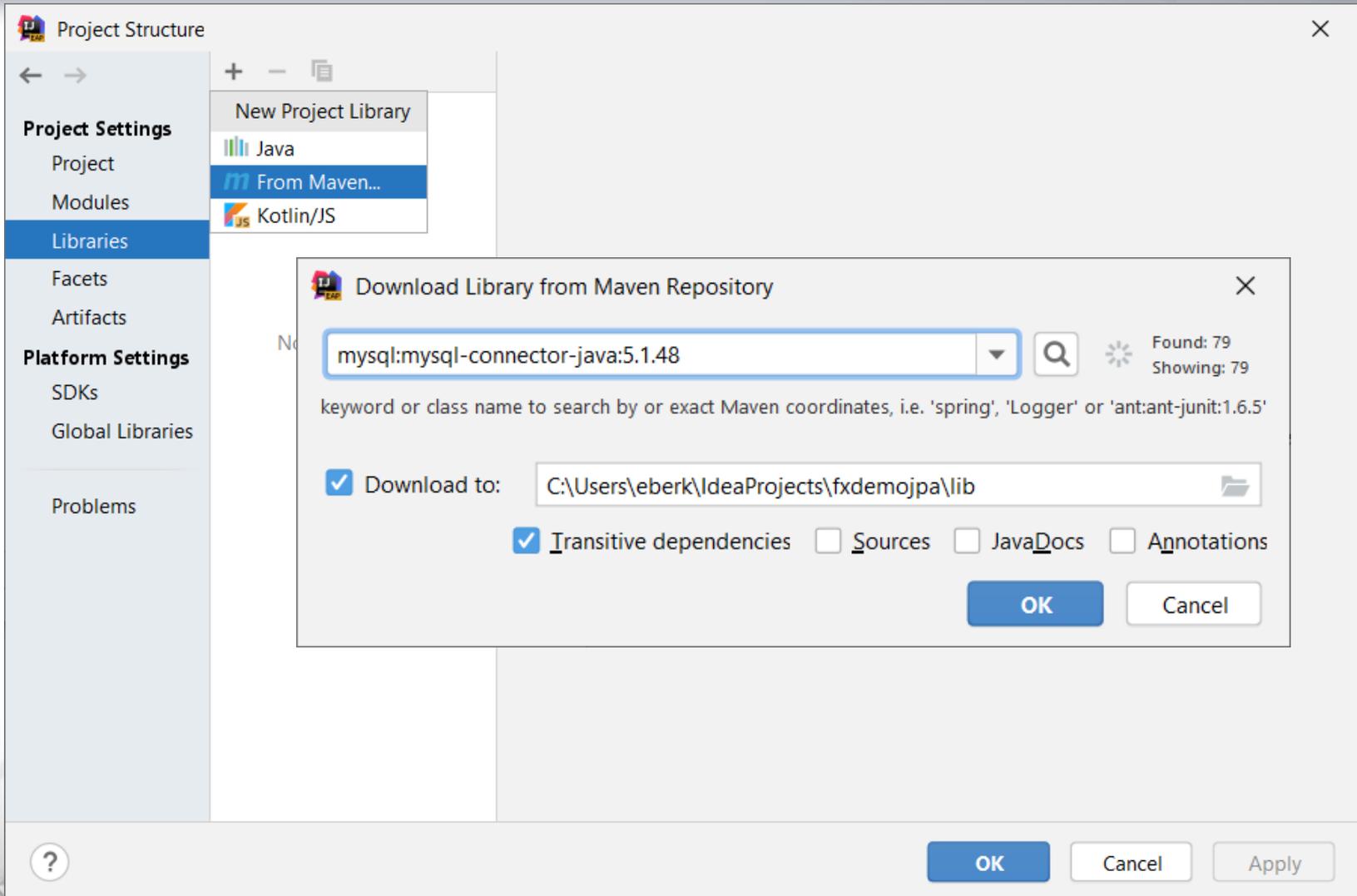
mysql

Found: 79
Showing: 79

- mysql:mysql-connector-java:6.0.2
- mysql:mysql-connector-java:5.1.48**
- mysql:mysql-connector-java:5.1.47
- mysql:mysql-connector-java:5.1.46
- mysql:mysql-connector-java:5.1.45
- mysql:mysql-connector-java:5.1.44
- mysql:mysql-connector-java:5.1.43

OK Cancel Apply

Подключаем библиотеки: 1 - MySQL



The screenshot shows the 'Project Structure' dialog in IntelliJ IDEA. The 'Libraries' tab is selected in the left sidebar. A context menu is open over the 'Libraries' tab, with 'From Maven...' selected. A secondary dialog, 'Download Library from Maven Repository', is overlaid on top. It features a search input field containing 'mysql:mysql-connector-java:5.1.48'. Below the search field, there are checkboxes for 'Download to:', 'Transitive dependencies', 'Sources', 'JavaDocs', and 'Annotations'. The 'Download to:' field is set to 'C:\Users\eberk\IdeaProjects\fxdemojpa\lib'. The 'Transitive dependencies' checkbox is checked. At the bottom of the dialog, there are 'OK', 'Cancel', and 'Apply' buttons.

Project Structure

Project Settings

- Project
- Modules
- Libraries
- Facets
- Artifacts

Platform Settings

- SDKs
- Global Libraries

Problems

New Project Library

- Java
- From Maven...
- Kotlin/JS

Download Library from Maven Repository

mysql:mysql-connector-java:5.1.48

Found: 79
Showing: 79

keyword or class name to search by or exact Maven coordinates, i.e. 'spring', 'Logger' or 'ant:ant-junit:1.6.5'

Download to: C:\Users\eberk\IdeaProjects\fxdemojpa\lib

Transitive dependencies Sources JavaDocs Annotations

OK Cancel

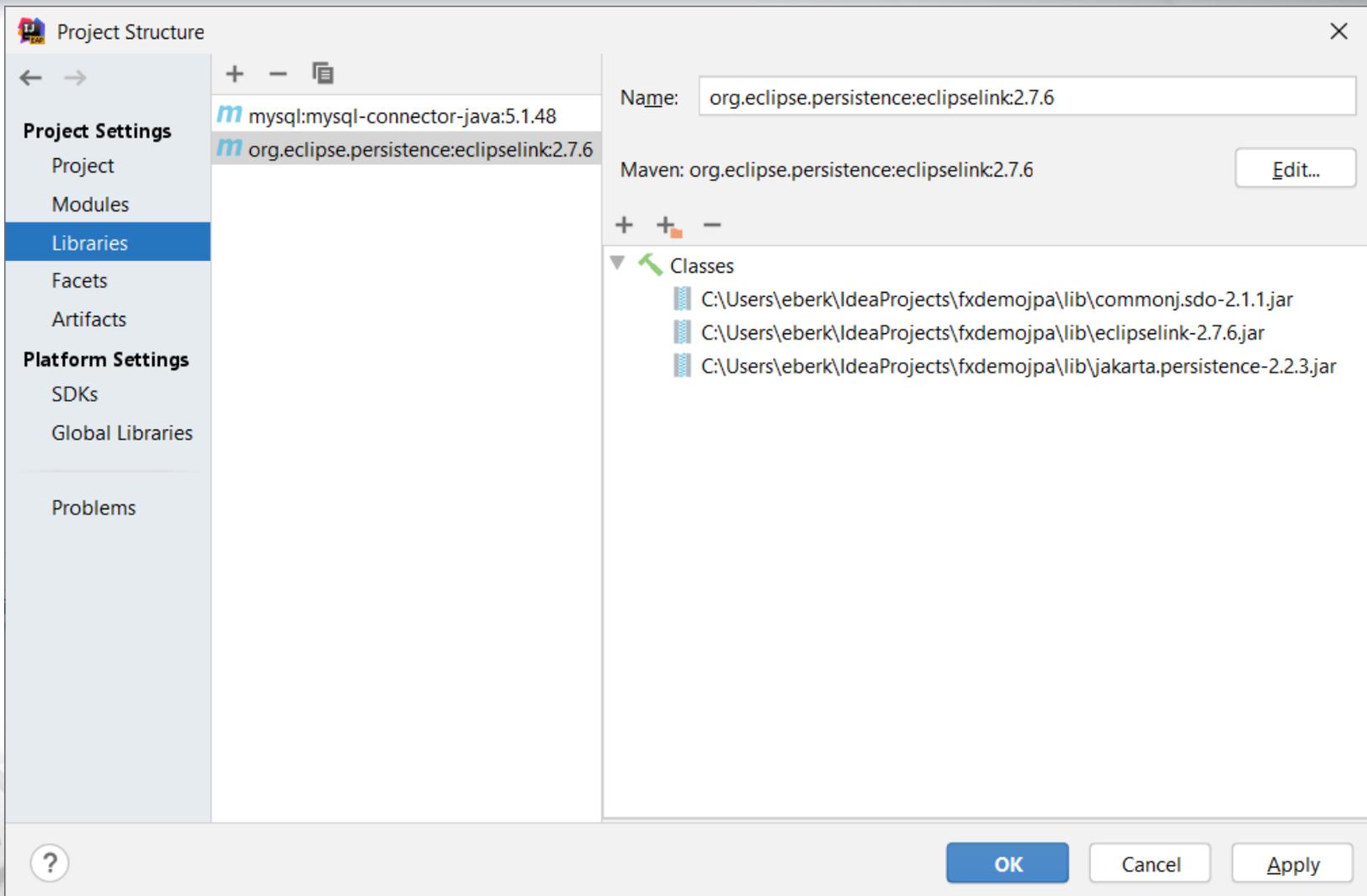
OK Cancel Apply

Подключаем библиотеки: 2 – EclipseLink (JPA)

The screenshot shows the Eclipse IDE interface. On the left, the 'Project Structure' window is open, with the 'Libraries' tab selected. The 'Project Settings' sidebar is visible, showing 'Project', 'Modules', 'Libraries', 'Facets', 'Artifacts', 'Platform Settings', 'SDKs', 'Global Libraries', and 'Problems'. The main workspace displays the 'Project Structure' tree with 'mysql:mysql-connector-java' selected. The 'Name' field shows 'mysql:mysql-connector-java:5.1.48' and the 'Maven' field shows 'mysql:mysql-connector-java:5.1.48'. An 'Edit...' button is visible next to the Maven field.

In the foreground, a 'Download Library from Maven Repository' dialog is open. The search input field contains 'org.eclipse.persistence'. The search results list several artifacts, with 'org.eclipse.persistence:eclipselink:2.7.6' selected. The dialog also shows 'Found: 373' and 'Showing: 373'. There are 'OK' and 'Cancel' buttons at the bottom of the dialog.

Подключаем библиотеки



Project Structure

← → + - 📄

Project Settings

- Project
- Modules
- Libraries**
- Facets
- Artifacts

Platform Settings

- SDKs
- Global Libraries

Problems

Name:

Maven:

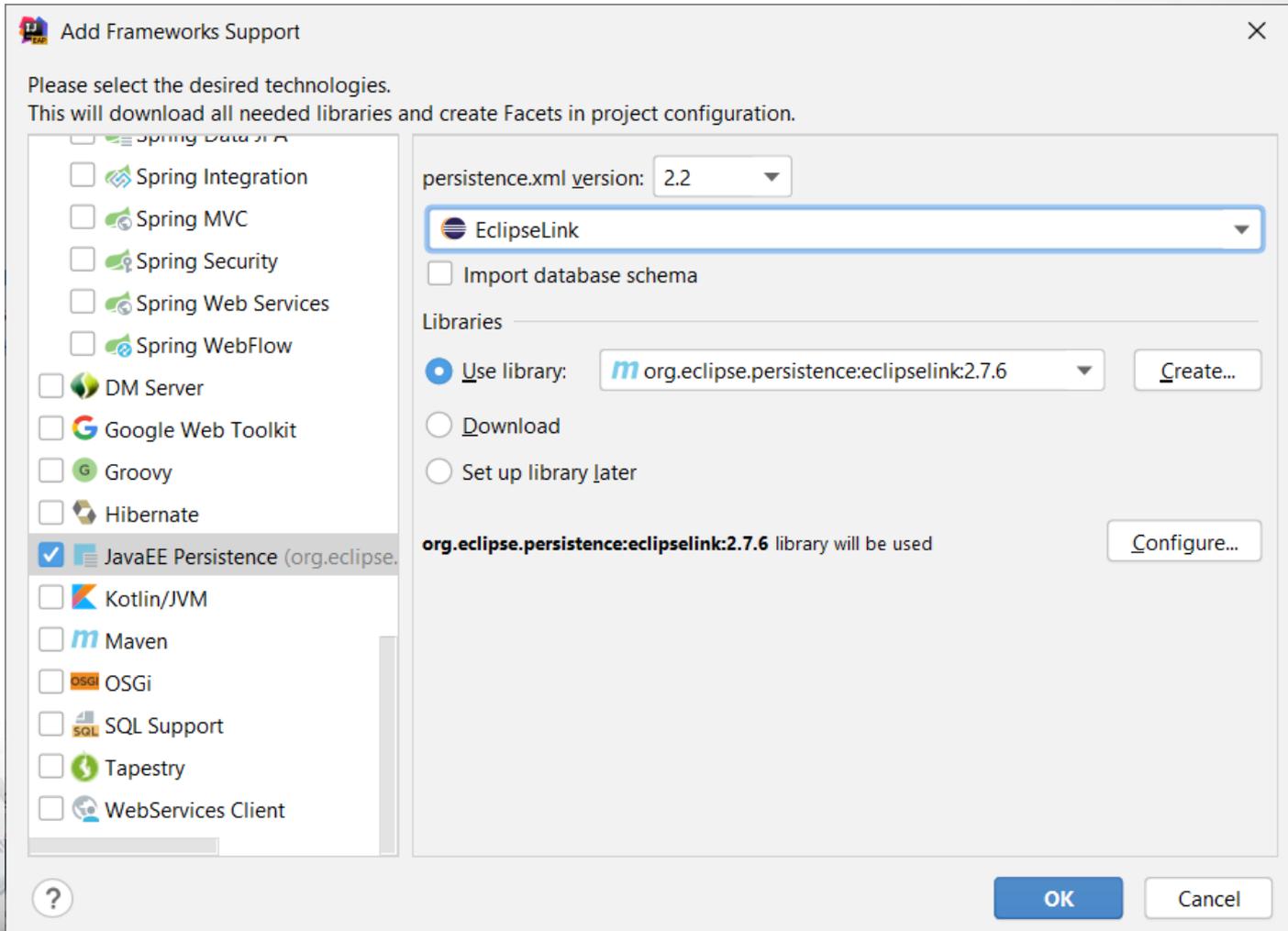
+ + -

Classes

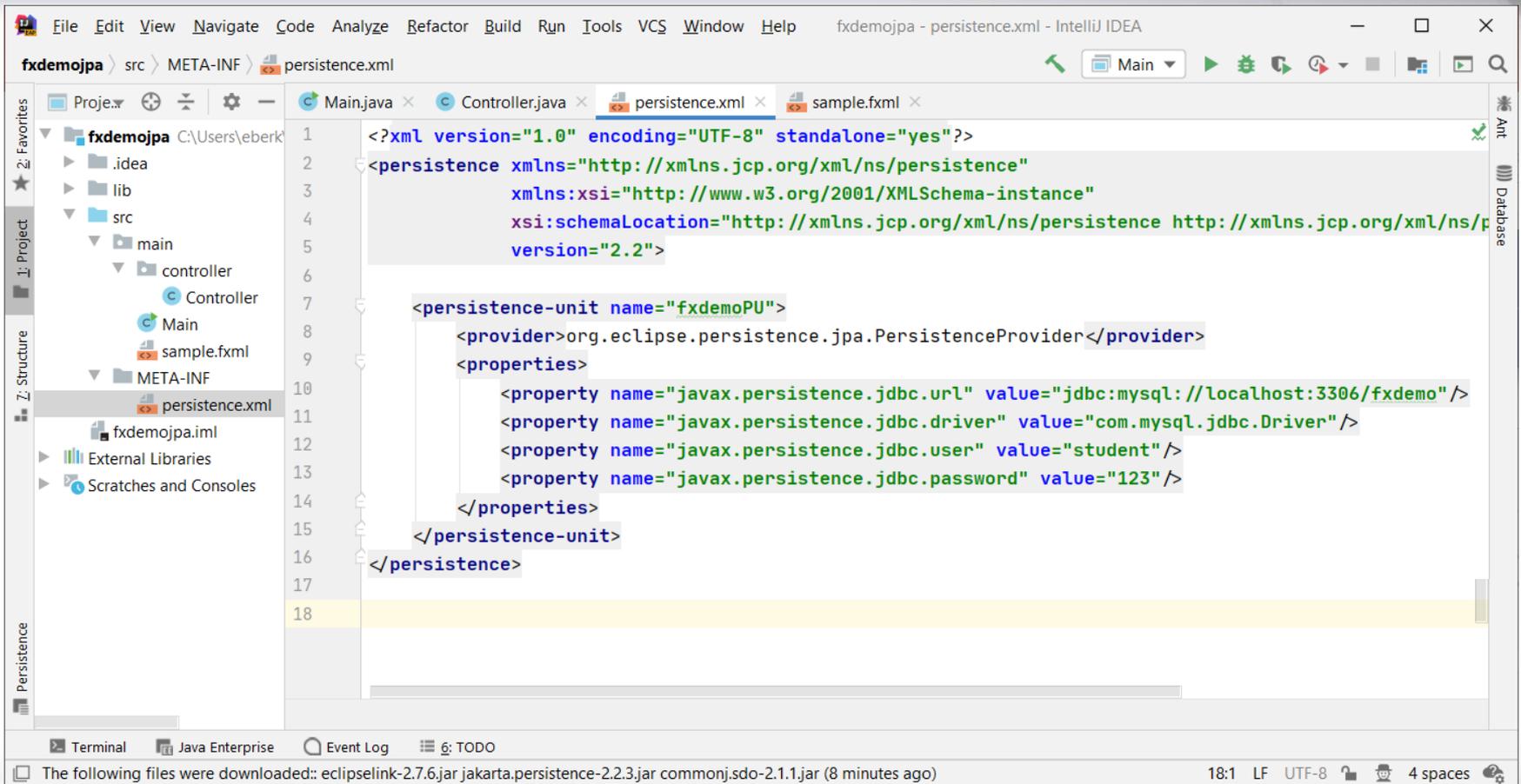
- C:\Users\eberk\IdeaProjects\fxdemojpa\lib\commonj.sdo-2.1.1.jar
- C:\Users\eberk\IdeaProjects\fxdemojpa\lib\eclipselink-2.7.6.jar
- C:\Users\eberk\IdeaProjects\fxdemojpa\lib\jakarta.persistence-2.2.3.jar

Включаем поддержку Persistence

ПКМ на узле проекта и выбрать «Add Frameworks Support...»



Редактируем Persistence Unit



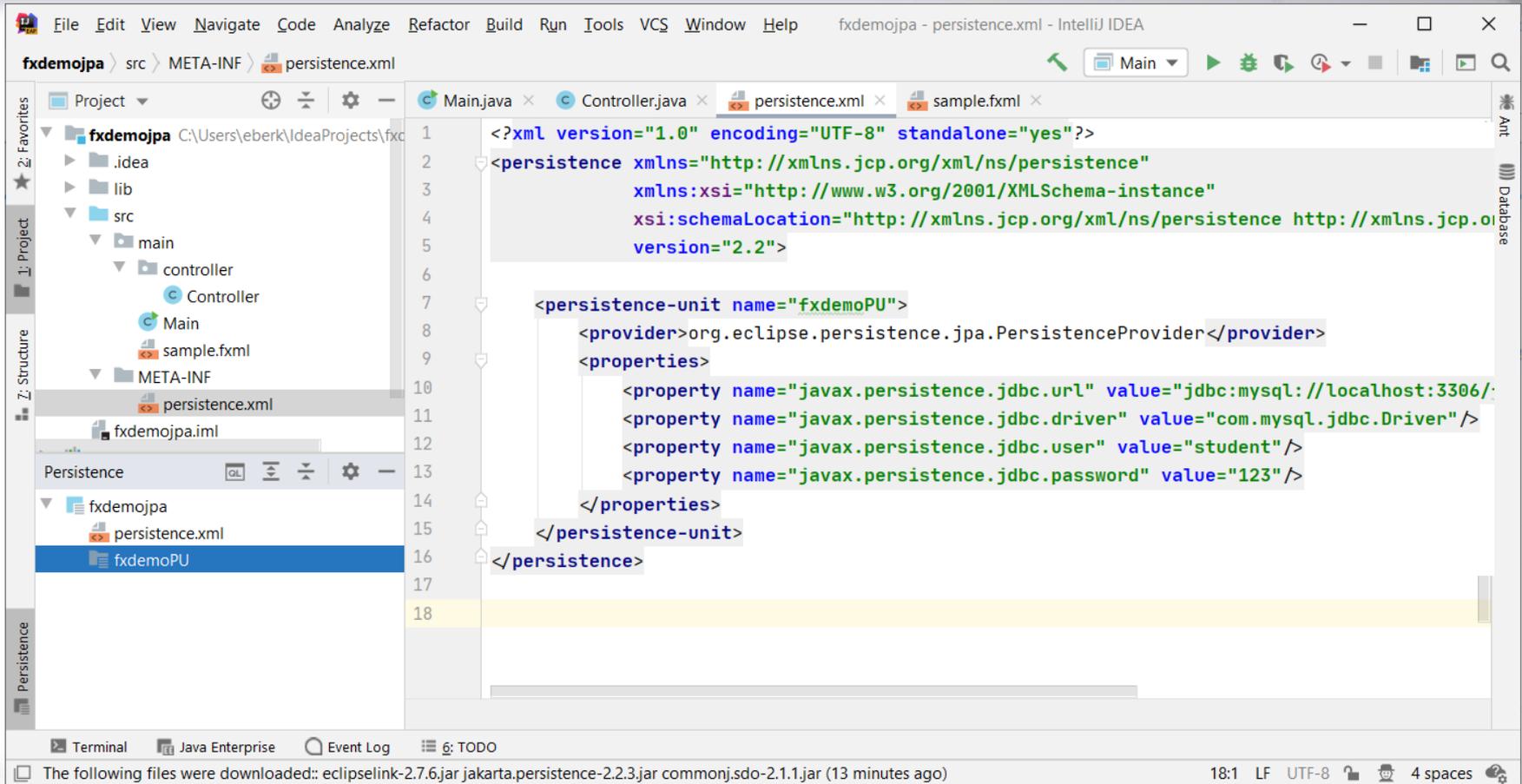
fxdemojpa - persistence.xml - IntelliJ IDEA

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<persistence xmlns="http://xmlns.jcp.org/xml/ns/persistence"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/persistence http://xmlns.jcp.org/xml/ns/persistence-2.2"
  version="2.2">
  <persistence-unit name="fxdemoPU">
    <provider>org.eclipse.persistence.jpa.PersistenceProvider</provider>
    <properties>
      <property name="javax.persistence.jdbc.url" value="jdbc:mysql://localhost:3306/fxdemo"/>
      <property name="javax.persistence.jdbc.driver" value="com.mysql.jdbc.Driver"/>
      <property name="javax.persistence.jdbc.user" value="student"/>
      <property name="javax.persistence.jdbc.password" value="123"/>
    </properties>
  </persistence-unit>
</persistence>
```

The following files were downloaded: eclipselink-2.7.6.jar jakarta.persistence-2.2.3.jar commonj.sdo-2.1.1.jar (8 minutes ago)

18:1 LF UTF-8 4 spaces

Открываем окно Persistence



```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<persistence xmlns="http://xmlns.jcp.org/xml/ns/persistence"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/persistence http://xmlns.jcp.org/xml/ns/persistence-2.2"
  version="2.2">
  <persistence-unit name="fxdemoPU">
    <provider>org.eclipse.persistence.jpa.PersistenceProvider</provider>
    <properties>
      <property name="javax.persistence.jdbc.url" value="jdbc:mysql://localhost:3306/...">
      <property name="javax.persistence.jdbc.driver" value="com.mysql.jdbc.Driver"/>
      <property name="javax.persistence.jdbc.user" value="student"/>
      <property name="javax.persistence.jdbc.password" value="123"/>
    </properties>
  </persistence-unit>
</persistence>
```

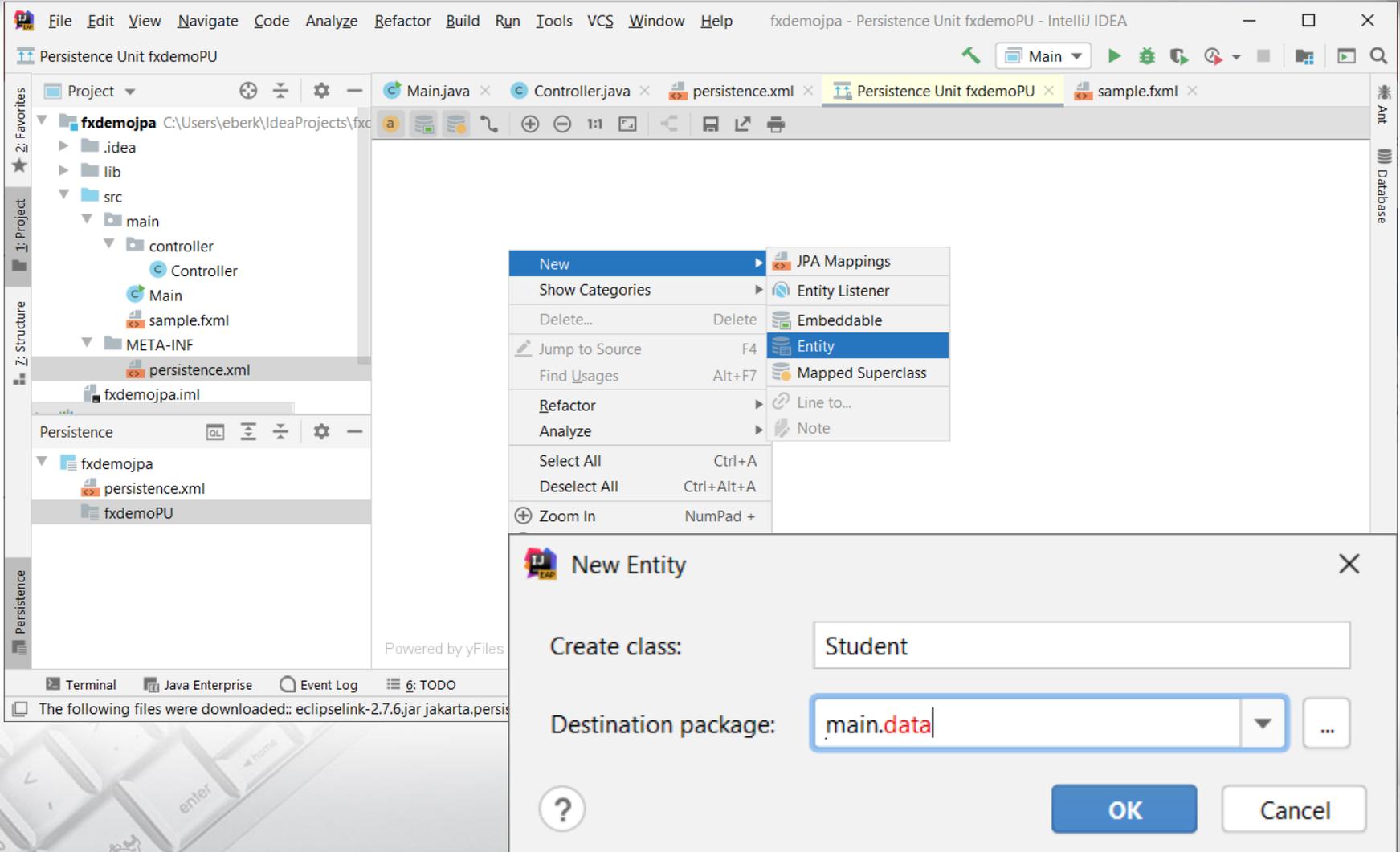
The following files were downloaded: eclipselink-2.7.6.jar jakarta.persistence-2.2.3.jar commonj.sdo-2.1.1.jar (13 minutes ago)

18:1 LF UTF-8 4 spaces

Создаем ER-диаграмму

The screenshot shows the IntelliJ IDEA IDE interface for a project named 'Persistence Unit fxdemoPU'. The main editor area is currently blank. On the left, the 'Project' and 'Structure' toolbars are visible, showing the project's directory tree. The 'Persistence' toolbar at the bottom left indicates that the ER diagram tool is active. A green box with the text 'ER Diagram' is overlaid on the bottom right of the main editor area. The status bar at the bottom shows a message: 'The following files were downloaded: eclipselink-2.7.6.jar jakarta.persistence-2.2.3.jar commonj.sdo-2.1.1.jar (14 minutes ago)'. The bottom of the image features a faint background image of a computer keyboard.

Добавляем классы модели

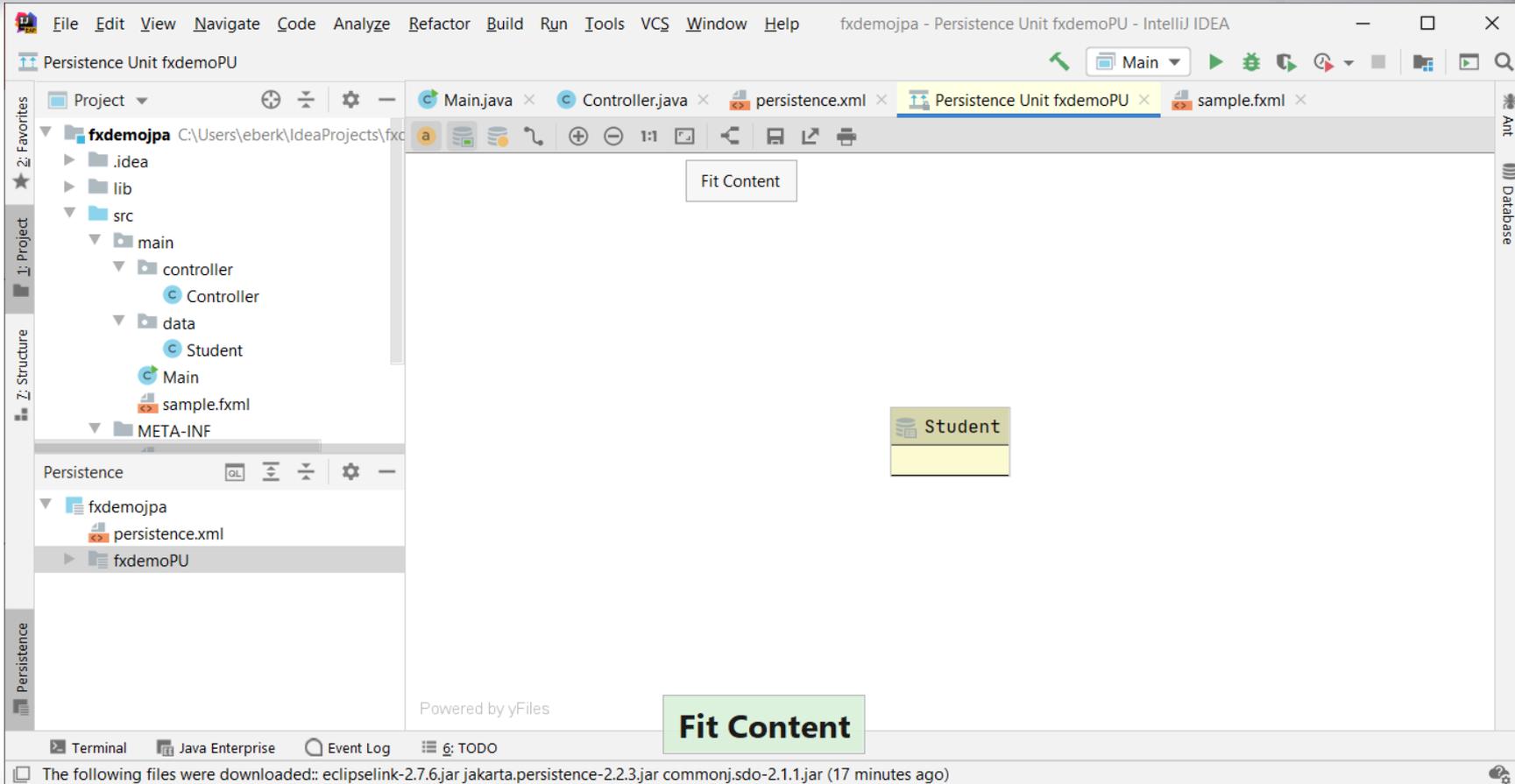


The screenshot shows the IntelliJ IDEA IDE interface. The main window displays the project structure for 'Persistence Unit fxdemoPU'. The 'New' menu is open, and the 'Entity' option is selected. A 'New Entity' dialog box is open in the foreground, showing the following fields:

- Create class:** Student
- Destination package:** main.data

The dialog also includes an 'OK' button, a 'Cancel' button, and a help icon (?).

Добавляем классы модели



The screenshot shows the IntelliJ IDEA IDE interface for a project named "Persistence Unit fxdemoPU". The main editor window displays a diagram with a "Fit Content" button and a "Student" class icon. The Project Structure view on the left shows the following hierarchy:

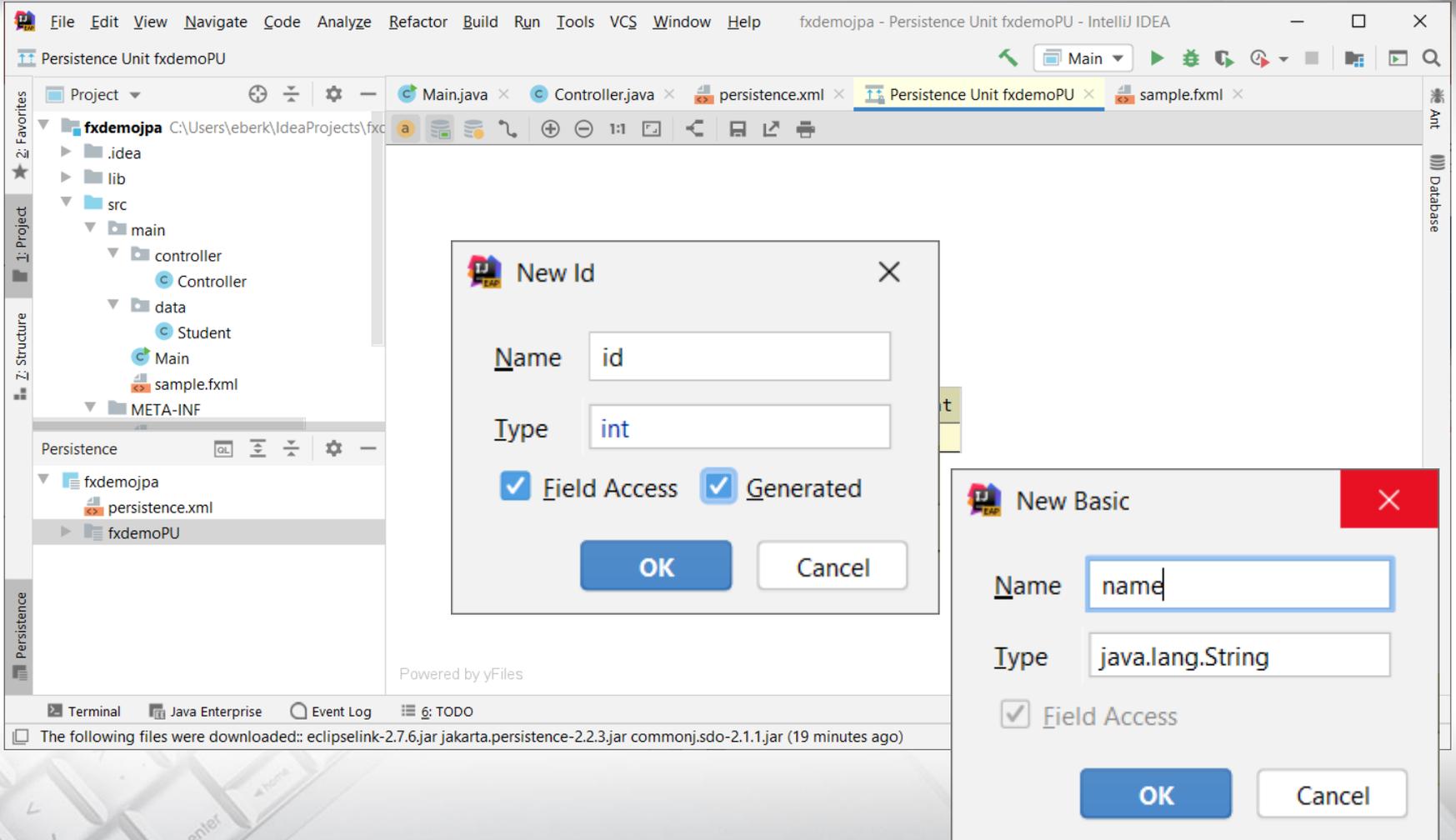
- fxdemojpa
 - .idea
 - lib
 - src
 - main
 - controller
 - Controller
 - data
 - Student
 - Main
 - sample.fxml
 - META-INF

The Persistence view at the bottom shows the project structure for the persistence unit:

- fxdemojpa
 - persistence.xml
 - fxdemoPU

The status bar at the bottom indicates that the following files were downloaded: `eclipselink-2.7.6.jar`, `jakarta.persistence-2.2.3.jar`, and `commonj.sdo-2.1.1.jar` (17 minutes ago).

Добавляем поля

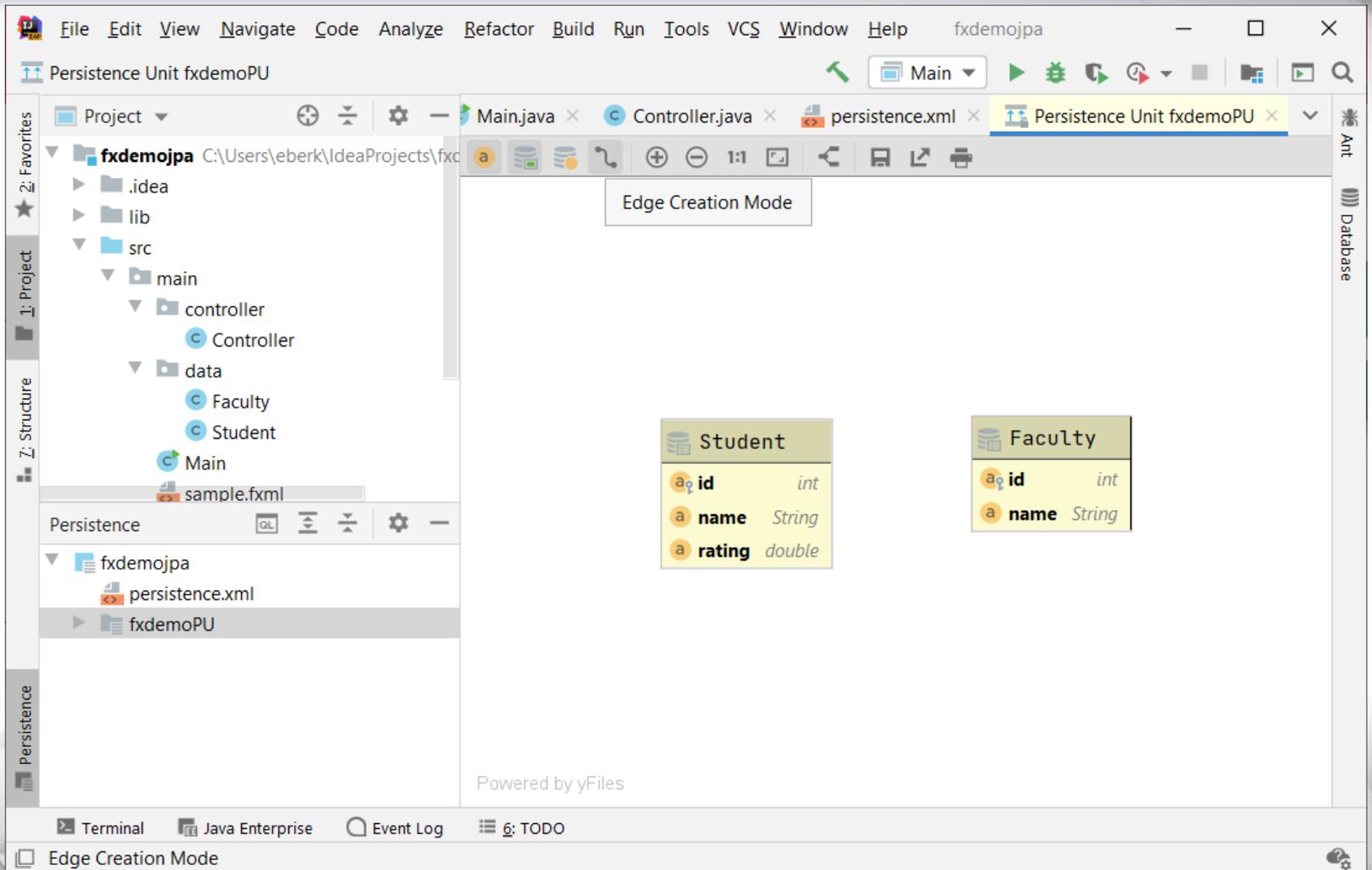


The screenshot displays the IntelliJ IDEA interface for a project named 'fxdemojpa'. The 'Persistence' tool window is open, showing the configuration for the 'fxdemoPU' unit. Two dialog boxes are overlaid on the interface:

- New Id Dialog:** This dialog is used to create a new ID field. The 'Name' field contains 'id', and the 'Type' field contains 'int'. Both 'Field Access' and 'Generated' checkboxes are checked. The 'OK' button is highlighted in blue.
- New Basic Dialog:** This dialog is used to create a new basic field. The 'Name' field contains 'name', and the 'Type' field contains 'java.lang.String'. The 'Field Access' checkbox is checked. The 'OK' button is highlighted in blue.

The background shows the project structure in the 'Project' tool window, including folders like 'main', 'controller', 'data', and 'META-INF'. The 'Persistence' tool window shows the 'fxdemoPU' unit with a 'persistence.xml' file. The bottom status bar indicates that several files were downloaded recently.

Добавляем классы модели



File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help fxdemojpa

Persistence Unit fxdemoPU

Project

- fxdemojpa C:\Users\eberk\IdeaProjects\fxd
- .idea
- lib
- src
 - main
 - controller
 - Controller
 - data
 - Faculty
 - Student
 - Main
- sample.fxml

Persistence

- fxdemojpa
 - persistence.xml
 - fxdemoPU

Edge Creation Mode

Student

- id int
- name String
- rating double

Faculty

- id int
- name String

Terminal Java Enterprise Event Log TODO

Edge Creation Mode

Powered by yFiles

«Рисуем» связь между классами

The screenshot shows an IDE window titled "Persistence Unit fxdemoPU". The left sidebar displays the project structure for "fxdemojpa", including folders ".idea", "lib", "src", "main", "controller", "data", and files "Controller", "Faculty", "Student", "Main", and "sample.fxml". The main editor area shows a class diagram with two classes: "Student" and "Faculty". The "Student" class has attributes "id" (int), "name" (String), and "rating" (double). The "Faculty" class has attributes "id" (int) and "name" (String). A line connects the "id" attribute of the "Student" class to the "id" attribute of the "Faculty" class, indicating an association. The bottom status bar shows a message: "The following files were downloaded: eclipselink-2.7.6.jar jakarta.persistence-2.2.3.jar commonj.sdo-2.1.1.jar (28 minutes ago)".

Настраиваем мощность связи

Create Relationship [X]

Class: Student Class: Faculty

Attribute: Attribute:

Multiplicity: Multiplicity:

Optional Owner Optional Owner

Fetch Type Cascade Type Fetch Type Cascade Type

eager all eager all
 lazy detach lazy detach
 merge merge

[?] [OK] [Cancel]

Полученная модель

The screenshot shows an IDE window titled "Persistence Unit fxdemoPU". The main editor displays a class diagram with two entities: Faculty and Student. Both entities have attributes: id (int), name (String), and rating (double). A bidirectional blue arrow connects the two entities, indicating a relationship. The left sidebar shows the project structure with folders for .idea, lib, src, and main, and files for Controller, Faculty, Student, Main, and sample.fxml. The bottom status bar shows a message: "The following files were downloaded: eclipselink-2.7.6.jar jakarta.persistence-2.2.3.jar commonj.sdo-2.1.1.jar (28 minutes ago)".

Переходим от модели к коду

The screenshot displays the IntelliJ IDEA IDE interface. On the left, the Project Structure tool window shows the project 'fxdemojpa' with a package structure including 'main', 'controller', and 'data'. The 'Persistence' tool window shows the 'fxdemojpa' persistence unit. The main editor area shows a UML class diagram with two classes: 'Faculty' and 'Student'. The 'Faculty' class has attributes 'id' and 'name'. The 'Student' class has attributes 'id', 'name', and 'rating'. A context menu is open over the 'Faculty' class, listing various actions such as 'New', 'Show Categories', 'Delete...', 'Jump to Source' (F4), 'Find Usages' (Alt+F7), 'Refactor', 'Analyze', 'Select All' (Ctrl+A), 'Deselect All' (Ctrl+Alt+A), 'Zoom In', 'Zoom Out', 'Actual Size' (Ctrl+NumPad /), 'Fit Content', 'Layout', and 'Save UML Diagram'. The 'Jump to Source' option is highlighted in blue. The bottom status bar shows 'Terminal', 'Java Enterprise', 'Event Log', and 'TODO' tabs, along with the instruction 'Open editor for the selected item and give focus to it'.

Перестраиваем сгенерированный код

Code -> Rearrange Code

```
@Entity
public class Faculty {
    @GeneratedValue
    @Id
    private int id;
    @Basic
    private String name;
    @OneToMany(mappedBy = "faculty")
    private Collection<Student> students;

    public int getId() { return id; }

    public void setId(int id) { this.id = id; }

    public String getName() { return name; }

    public void setName(String name) { this.name = name; }

    public Collection<Student> getStudents() { return students; }

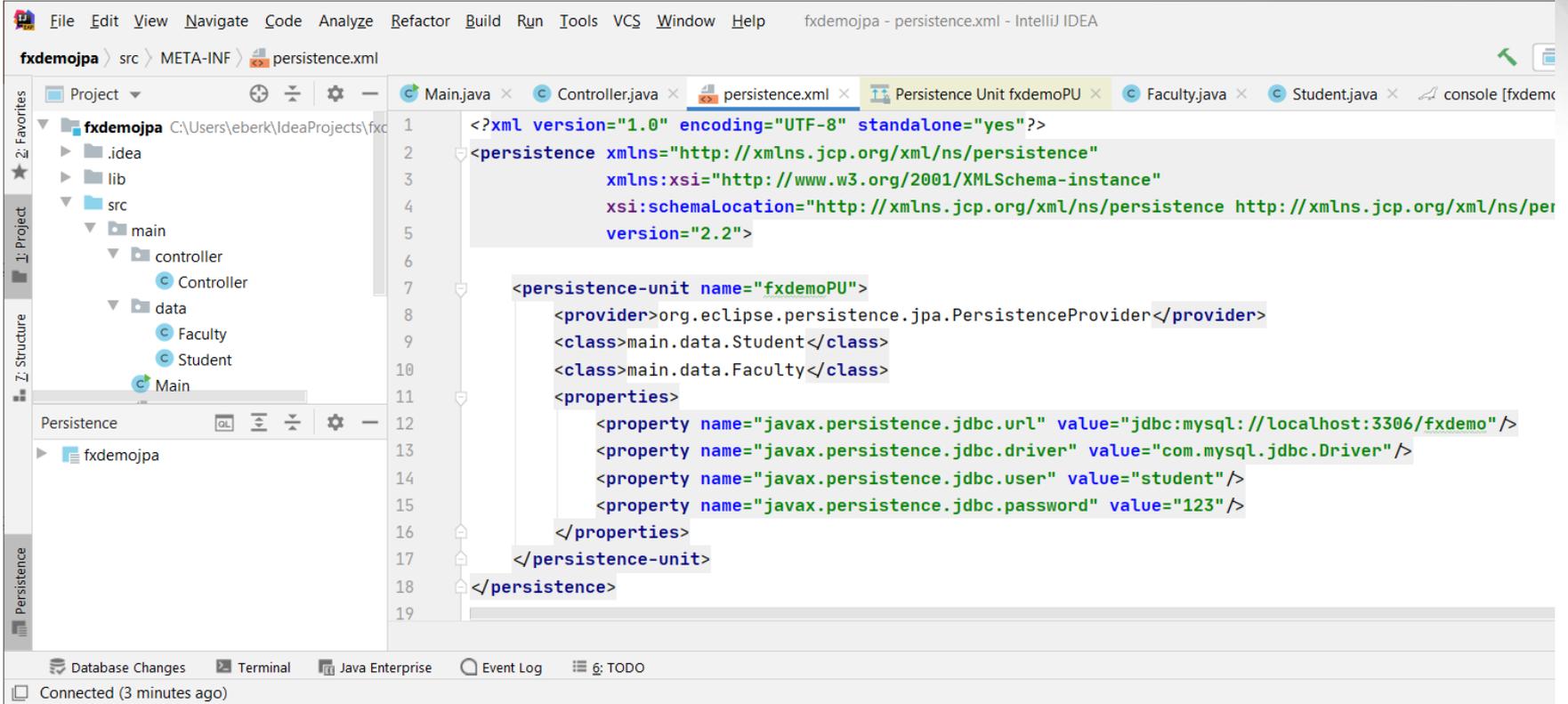
    public void setStudents(Collection<Student> students) { this.students = students; }
}
```

Перестраиваем и дописываем сгенерированный код

```
@Entity
@NamedQueries({
    @NamedQuery(name = "Faculty.findAll", query = "select f from Faculty f")
})
public class Faculty {
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    @Id
    private int id;
    @Basic
    private String name;
    @OneToMany(mappedBy = "faculty")
    private Collection<Student> students;
```

Перестраиваем и дописываем сгенерированный код

```
@Entity
@NamedQueries({
    @NamedQuery(name = "Student.findAll", query = "select s from Student s")
})
public class Student {
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    @Id
    private int id;
    @Basic
    private String name;
    @Basic
    private double rating;
    @ManyToOne(optional = false)
    private Faculty faculty;
```



The screenshot shows the IntelliJ IDEA IDE with the file `persistence.xml` open. The XML content is as follows:

```
1 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
2 <persistence xmlns="http://xmlns.jcp.org/xml/ns/persistence"
3     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4     xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/persistence http://xmlns.jcp.org/xml/ns/per
5     version="2.2">
6
7     <persistence-unit name="fxdemoPU">
8         <provider>org.eclipse.persistence.jpa.PersistenceProvider</provider>
9         <class>main.data.Student</class>
10        <class>main.data.Faculty</class>
11        <properties>
12            <property name="javax.persistence.jdbc.url" value="jdbc:mysql://localhost:3306/fxdemo"/>
13            <property name="javax.persistence.jdbc.driver" value="com.mysql.jdbc.Driver"/>
14            <property name="javax.persistence.jdbc.user" value="student"/>
15            <property name="javax.persistence.jdbc.password" value="123"/>
16        </properties>
17    </persistence-unit>
18 </persistence>
```

The IDE interface includes a menu bar (File, Edit, View, Navigate, Code, Analyze, Refactor, Build, Run, Tools, VCS, Window, Help), a toolbar, and several tool windows: Project (showing the project structure), Persistence (showing the persistence configuration), Database Changes, Terminal, Java Enterprise, Event Log, and TODO. The status bar at the bottom indicates "Connected (3 minutes ago)".

Підключаємося к БД из IDEA

The screenshot shows the IntelliJ IDEA IDE interface. The main editor displays the code for `Student.java` in the `main.data` package. The code includes annotations for JPA entities and a class with attributes and methods. The `Database` tool window is open on the right, showing a list of data sources. The `MySQL` option is selected. The `Project` tool window on the left shows the project structure, including the `data` package and the `Student` class. The `Persistence` tool window at the bottom left shows the persistence configuration for the project.

```
1 package main.data;
2
3 import javax.persistence.*;
4
5 @Entity
6 @NamedQueries({
7     @NamedQuery(name = "Student.findAll", query = "select s from Student s")
8 })
9 public class Student {
10     @GeneratedValue(strategy = GenerationType.IDENTITY)
11     @Id
12     private int id;
13     @Basic
14     private String name;
15     @Basic
16     private double rating;
17     @ManyToOne(optional = false)
18     private Faculty faculty;
19
20     public int getId() { return id; }
21
22     public void setId(int id) { this.id = id; }
23
24     public String getName() { return name; }
25
26     public void setName(String name) { this.name = name; }
27
28     public double getRating() { return rating; }
29 }
```

Database

- Data Source
 - Amazon Redshift
 - Apache Cassandra
 - Apache Derby
 - Apache Hive
 - Azure SQL Database
 - ClickHouse
 - Exasol
 - Greenplum
 - H2
 - HSQldb
 - IBM Db2
 - MariaDB
 - Microsoft SQL Server
 - MongoDB
 - MySQL
 - Oracle
 - PostgreSQL
 - SQLite
 - Snowflake
 - Sybase ASE
 - Vertica
- DDL Data Source
- Data Source from URL
- Data Source from Path
- Driver and Data Source
- Driver
- Import Data Sources...

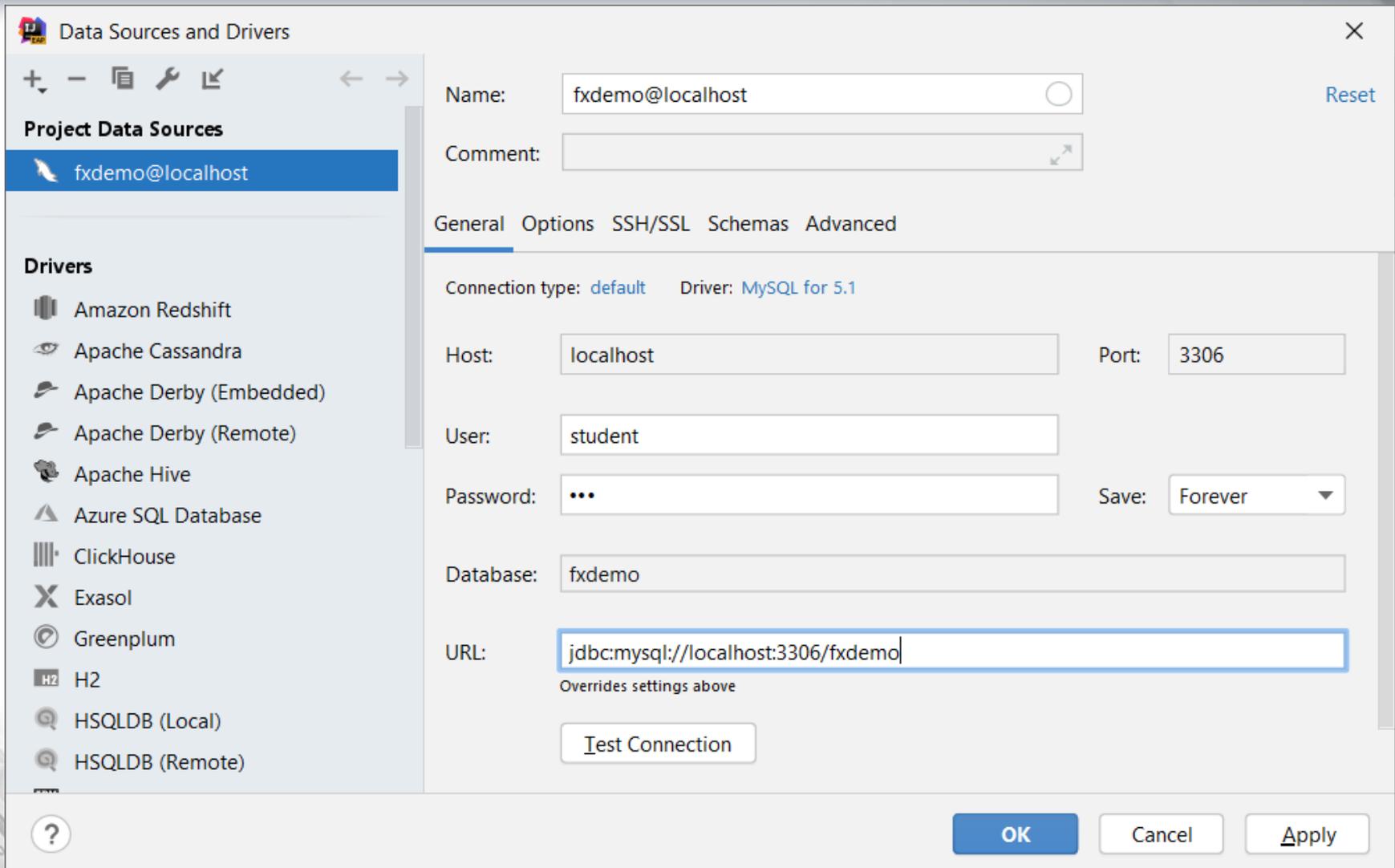
Create a data source

Terminal Java Enterprise Event Log 6: TODO

The following files were downloaded: eclipseLink-2.7.6.jar jakarta.persistence-2.2.3.jar commonj.sdo-2.1.1.jar (51 minutes ago)

1:4 CRLF UTF-8 4 spaces

Указываем параметры ПОДКЛЮЧЕНИЯ



Data Sources and Drivers

Project Data Sources

- fxdemo@localhost

Drivers

- Amazon Redshift
- Apache Cassandra
- Apache Derby (Embedded)
- Apache Derby (Remote)
- Apache Hive
- Azure SQL Database
- ClickHouse
- Exasol
- Greenplum
- H2
- HSQldb (Local)
- HSQldb (Remote)

Name:

Comment:

General Options SSH/SSL Schemas Advanced

Connection type: default Driver: MySQL for 5.1

Host: Port:

User:

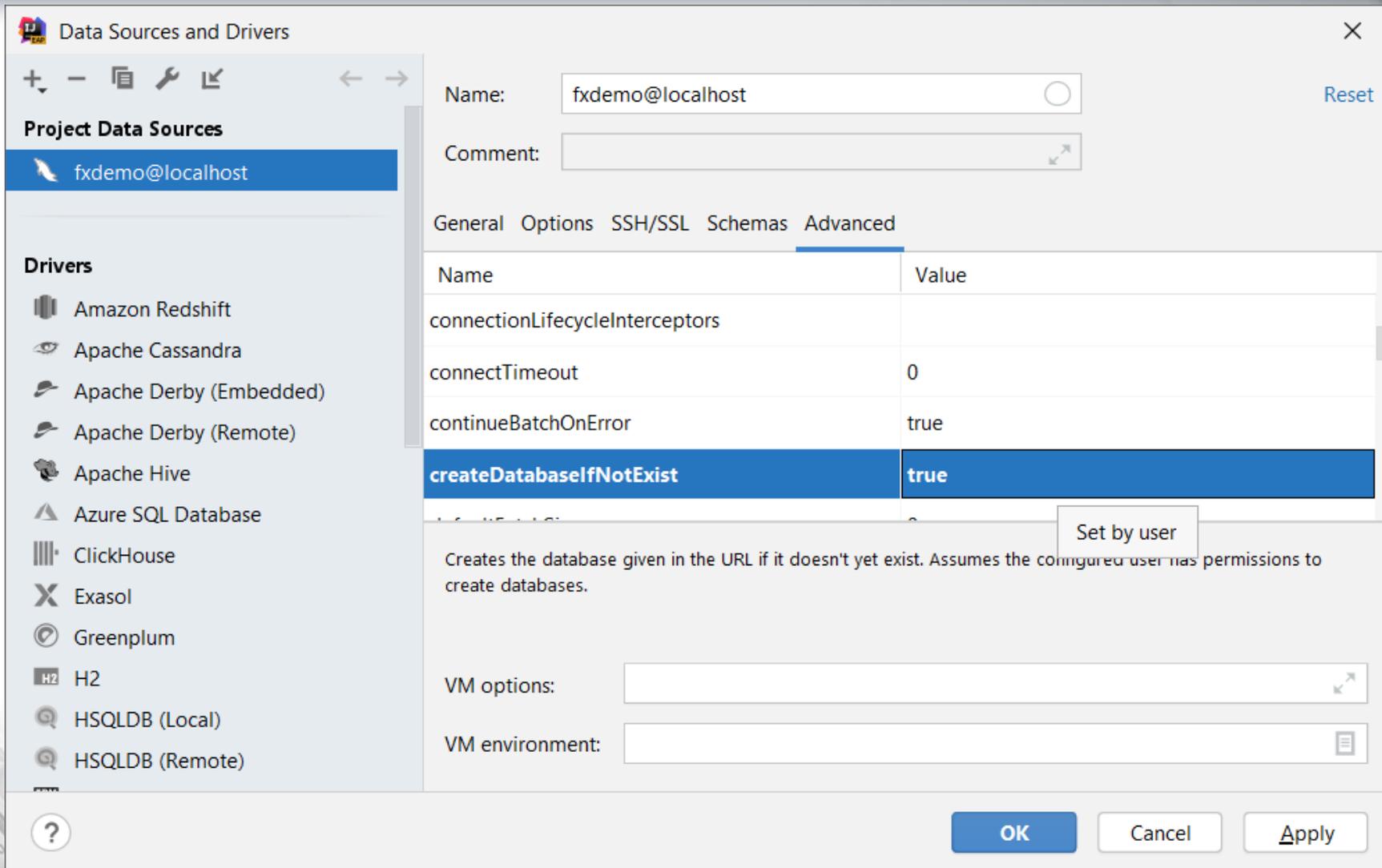
Password: Save:

Database:

URL:

Overrides settings above

... и дополнительные параметры ПОДКЛЮЧЕНИЯ



Data Sources and Drivers

Name: Reset

Comment:

General Options SSH/SSL Schemas **Advanced**

Name	Value
connectionLifecycleInterceptors	
connectTimeout	0
continueBatchOnError	true
createDatabaseIfNotExist	true

Set by user

Creates the database given in the URL if it doesn't yet exist. Assumes the configured user has permissions to create databases.

VM options:

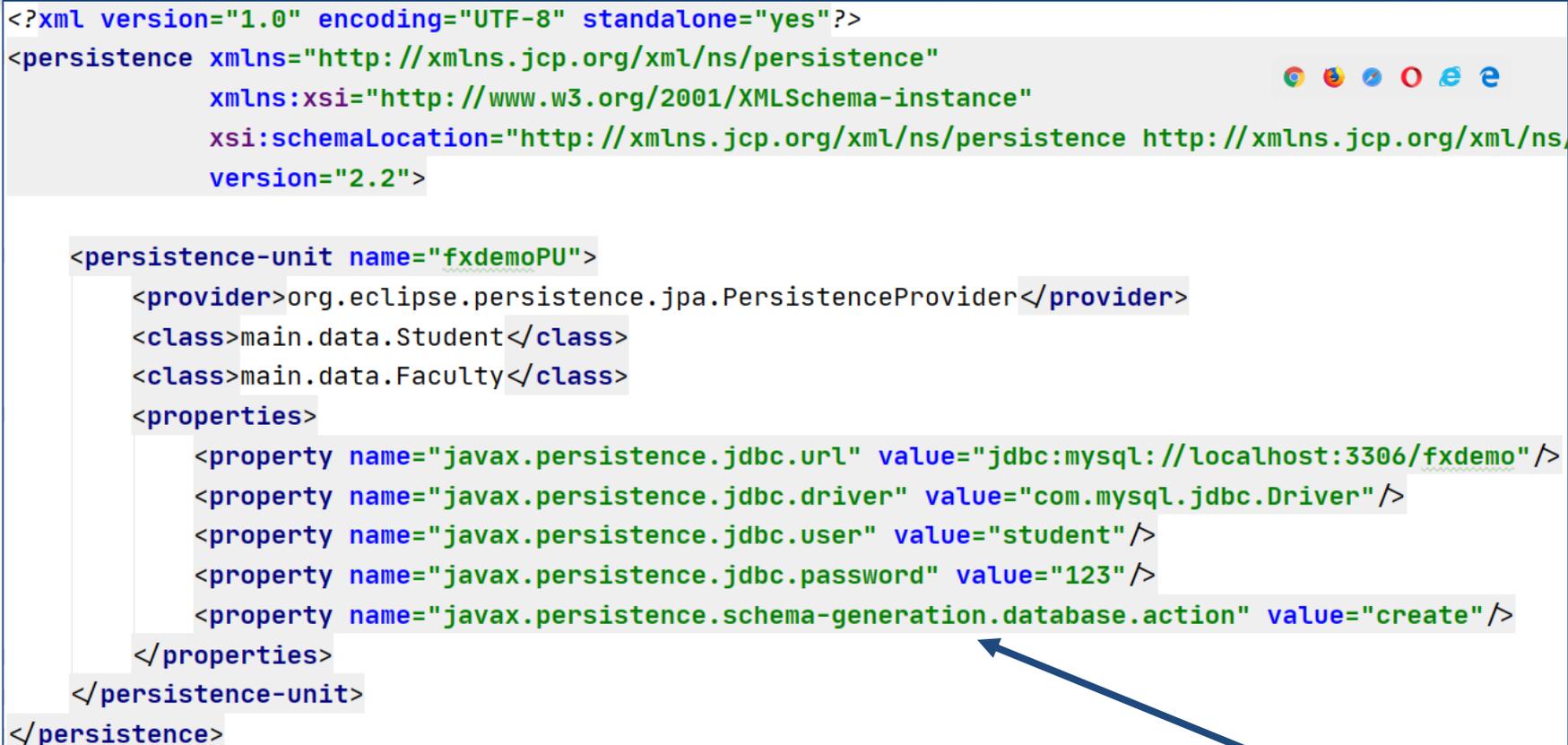
VM environment:

OK Cancel Apply

Возвращаемся в Persistence Unit. Добавляем строку

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<persistence xmlns="http://xmlns.jcp.org/xml/ns/persistence"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/persistence http://xmlns.jcp.org/xml/ns,
  version="2.2">

  <persistence-unit name="fxdemoPU">
    <provider>org.eclipse.persistence.jpa.PersistenceProvider</provider>
    <class>main.data.Student</class>
    <class>main.data.Faculty</class>
    <properties>
      <property name="javax.persistence.jdbc.url" value="jdbc:mysql://localhost:3306/fxdemo"/>
      <property name="javax.persistence.jdbc.driver" value="com.mysql.jdbc.Driver"/>
      <property name="javax.persistence.jdbc.user" value="student"/>
      <property name="javax.persistence.jdbc.password" value="123"/>
      <property name="javax.persistence.schema-generation.database.action" value="create"/>
    </properties>
  </persistence-unit>
</persistence>
```



Пишем код (!)

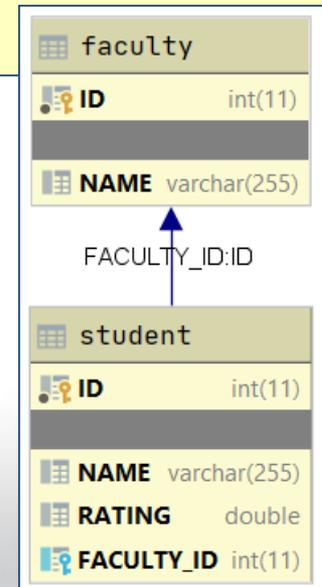
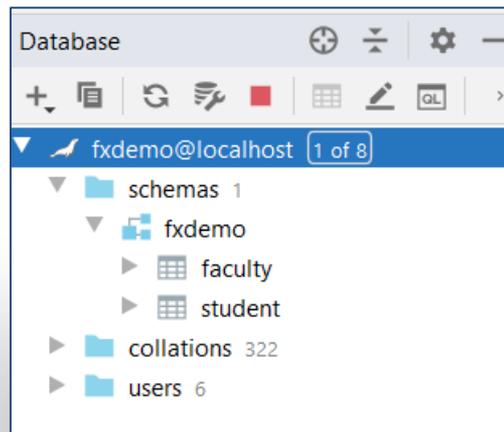
Вносим изменения в Controller

```
public class Controller {  
  
    public void initialize() {  
        EntityManagerFactory factory = Persistence  
            .createEntityManagerFactory("fxdemoPU");  
        EntityManager em = factory.createEntityManager();  
    }  
}
```

И запускаем...

Но, таблиц в БД еще нет?..

Или уже есть?

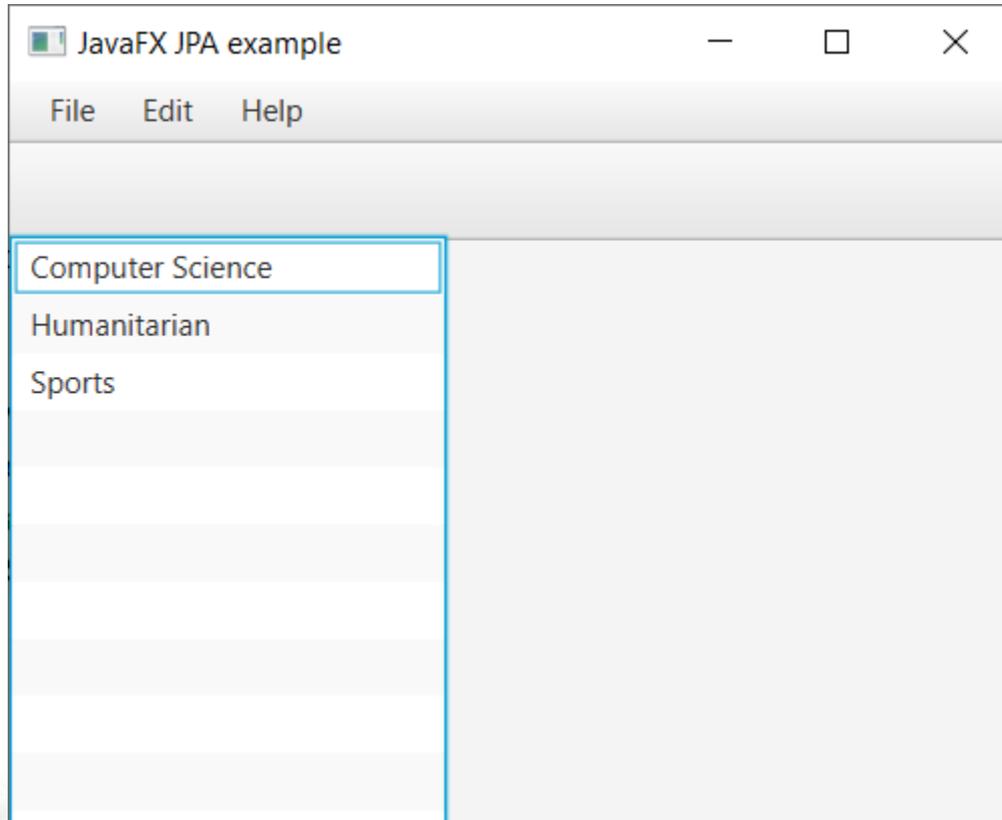


```
public class FacultyService {  
    private EntityManager em;  
  
    public FacultyService(EntityManager em) {  
        this.em = em;  
    }  
  
    public List<Faculty> findAll() {  
        return em.createNamedQuery(  
            "Faculty.findAll",  
            Faculty.class)  
            .getResultList();  
    }  
}
```

Снова пишем код!

```
public class Controller {  
  
    @FXML private ListView<Faculty> facultyList;  
  
    private FacultyService fs;  
  
    public void initialize() {  
        EntityManagerFactory factory = Persistence  
            .createEntityManagerFactory("fxdemoPU");  
        EntityManager em = factory.createEntityManager();  
        fs = new FacultyService(em);  
  
        facultyList.setItems(  
            FXCollections.observableList(fs.findAll()));  
    }  
}
```

Запускаєм



Дописываем код

```
public void initialize() {
    EntityManagerFactory factory =
        Persistence.createEntityManagerFactory("fxdemoPU");
    EntityManager em = factory.createEntityManager();

    idColumn.setCellValueFactory(new PropertyValueFactory<>("id"));
    nameColumn.setCellValueFactory(new PropertyValueFactory<>("name"));
    ratingColumn.setCellValueFactory(
        new PropertyValueFactory<>("rating"));

    fs = new FacultyService(em);

    facultyList.setItems(FXCollections.observableList(fs.findAll()));
    facultyList.getSelectionModel().selectedItemProperty().addListener(
        (observable, oldValue, newValue) -> studentTable.setItems(
            FXCollections.observableArrayList(newValue.getStudents()))
    );
}
```


Код по ссылке

