Министерство образования Республики Беларусь

Учреждение Образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерного проектирования

Кафедра инженерной психологии и эргономики

Дисциплина: Базы данных

Лабораторная работа № 5

«Реализация библиотеки функций для взаимодействия с базой данных на языке высокого уровня»

Вариант

Проверила: Василькова А.Н. Выполнила:

ст. гр. 113801

Мелкумова А.Э.

Минск 2022

Цель работы - изучение реализация библиотеки функций для взаимодействия с БД с использованием языка программирования Java, средствами разработки IntelliJ IDEA, Scene Builder и с использованием сервера баз данных MySQL Server.

Для начала, создадим новый javafx проект в среде разработки IntelliJ IDEA. Мы получили проект, в котором есть созданные автоматически классы HelloController и HelloApplication, а также файлы .xml и .fxml.

Далее мы запустили визуальный инструмент дизайна Scene Builder и подключили к нему .fxml файл из нашего проекта. В Scene Buider мы спроектировали внешний вид оконного приложения для подключения к таблицам БД. Полученный макет изображен на рисунке 1.

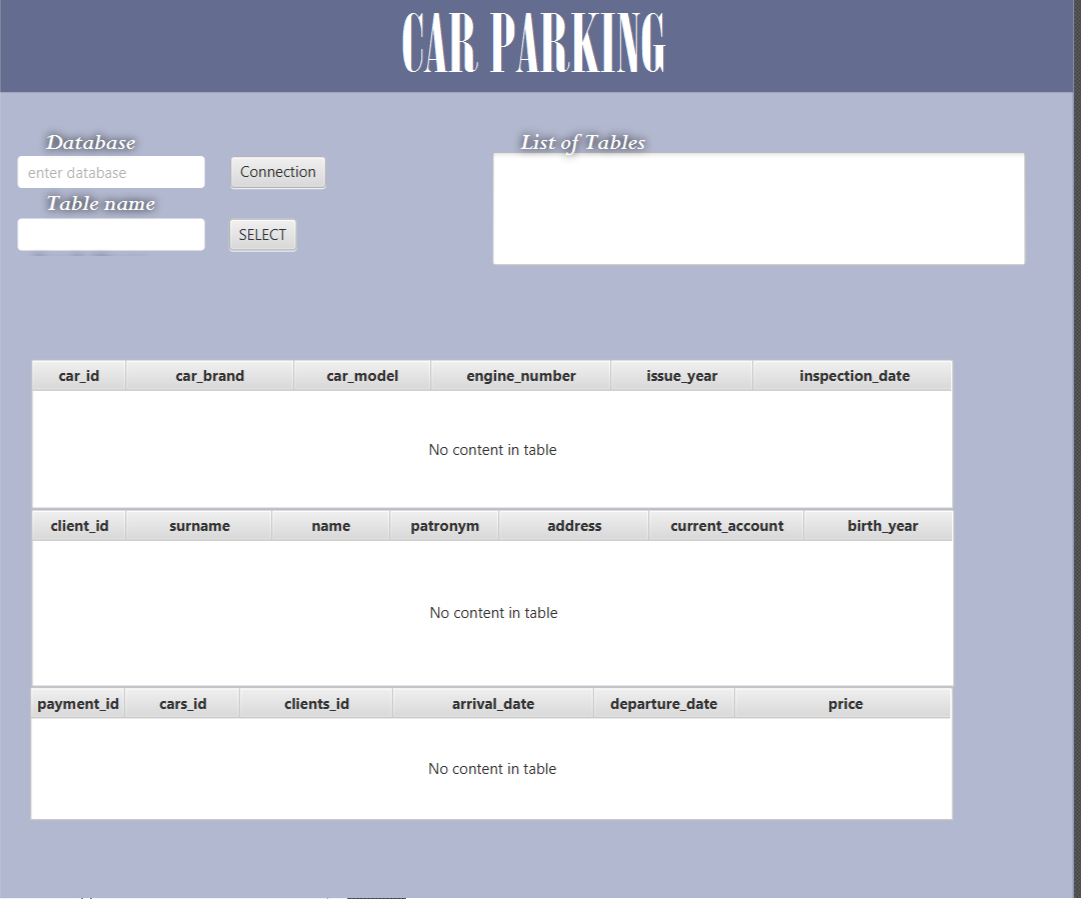


Рисунок 1 – Макет оконного приложения в Scene Builder

На данном макете мы создали два текстовых поля для ввода названия базы данных и таблицы, две кнопки для подключения к БД и вывода таблицы, текстовое поле для вывода перечня имеющихся в БД таблиц, объекты класса Label для подписи элементов. Также были созданы три слоя с таблицами класса TableView.

Всем элементам макета, кроме элементов Label, присвоены уникальные ID.

Созданные в таблицах поля соответствуют полям таблиц entrant, education и exams. На макете таблицы отображаться не будут, так как параметр видимость visiable таблиц имеет значение false и будет менять своё значение на true при выборе соответствующей таблицы. Конечный вид макета оконного приложения приведен на рисунке 2.



Рисунок 2 – Конечный вид оконного приложения

Затем мы сохранили изменения .fxml файла. Они автоматически отобразились в нашем проекте. Код для класса HelloConnector также возьмем в Scene Builder. Окно с кодом для коннектора приведено на рисунке 3.

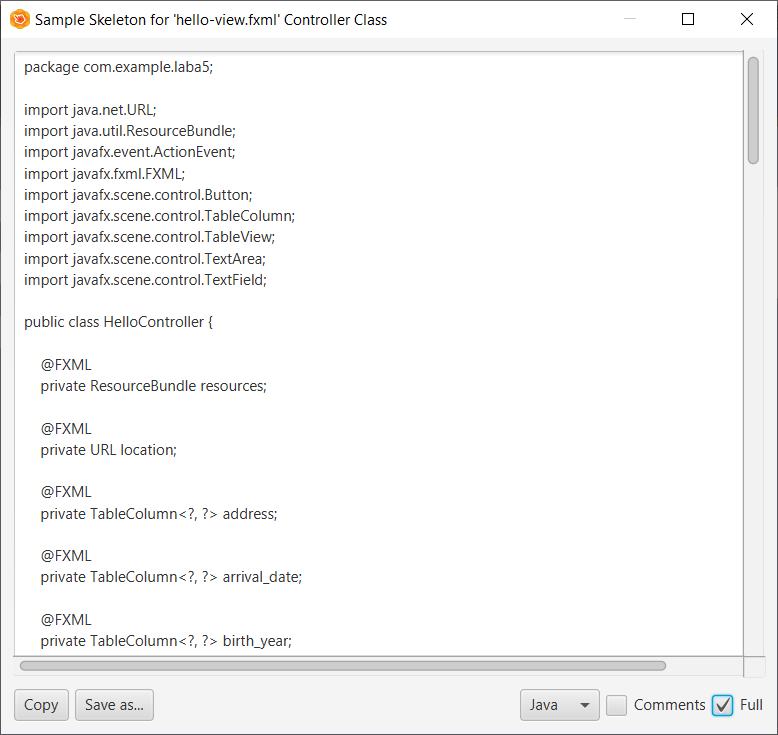


Рисунок 3 – Получение кода для класса HelloConnector

В нашем javafx проекте создадим три класса: Entranr, Education, Exams. Поля этих классов получат названия столбцов соответствующих таблиц, а тип данных будет соответствовать типу данных этих столбцов. На рисунке 4 приведен пример создания полей для класса Exams:

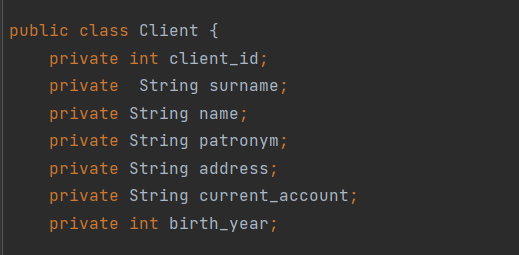


Рисунок 4 – Создание полей класса Exams

Таким же образом были прописаны поля для классов Entrant и Exams. Также для всех классов были созданы геттеры и конструкторы.

Следующим этапом в классе HelloController были заполнены названия классов и тип данных, хранимых в столбцах таблиц. Часть заполненных данных приведена на рисунке 5.

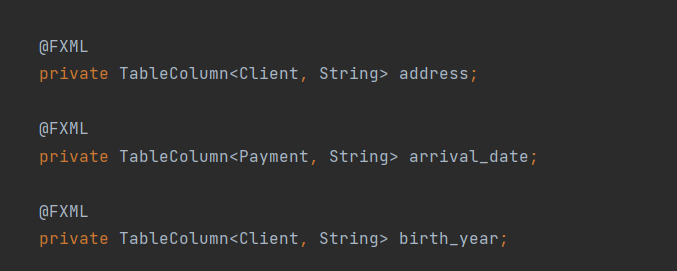


Рисунок 5 – Указание классов и типов данных для столбцов таблиц

В классе HelloController будет реализован интерфейс Initializable. Создадим функию initialize и пропишем в ней инициализацию всеx полей таблиц. Часть полученного результата представлена на рисунке 6.

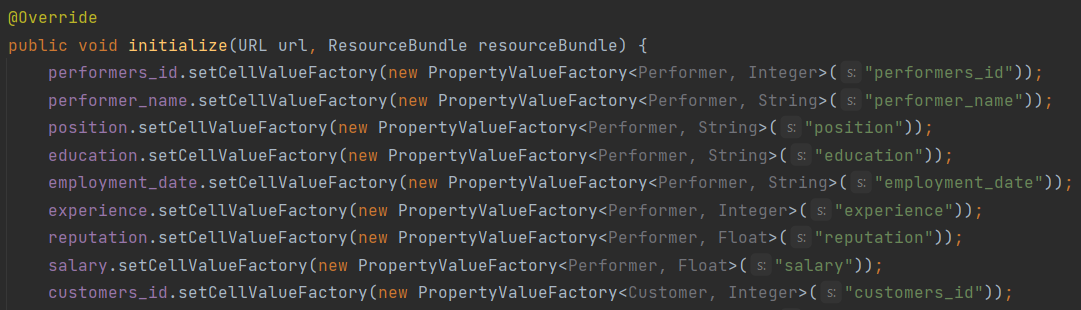


Рисунок 6 – Реализация метода initialize

Для подключения к базе данных мы создали класс DB. Объявление полей класса DB приведено на рисунке 7.

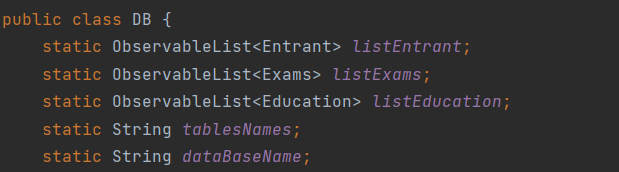


Рисунок 7 – Объявление полей класса DB

Подключение к базе данных будет происходить по нажатию копки Connection с id=btnConnection. При нажатии на кнопку, вызывается метод onConnectButtonClick, реализация которого приведена на рисунке 8.

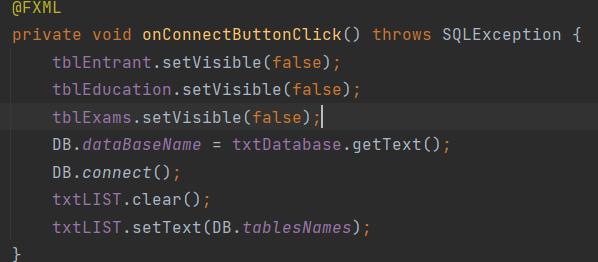


Рисунок 8 – Реализация метода onConnectButtonClick

При вызове метода атрибуту visiable таблиц присваивается значение false. Затем поле databaseName класса DB инициализируется значением, полученным из текстового поля txtLIST. Далее в классе DB вызывается метод connect, текстовое поле с именами таблиц базы данных очищается и заполняется новыми значениями или сообщением об ошибке.

Реализация метода connect приведена на рисунке 9.

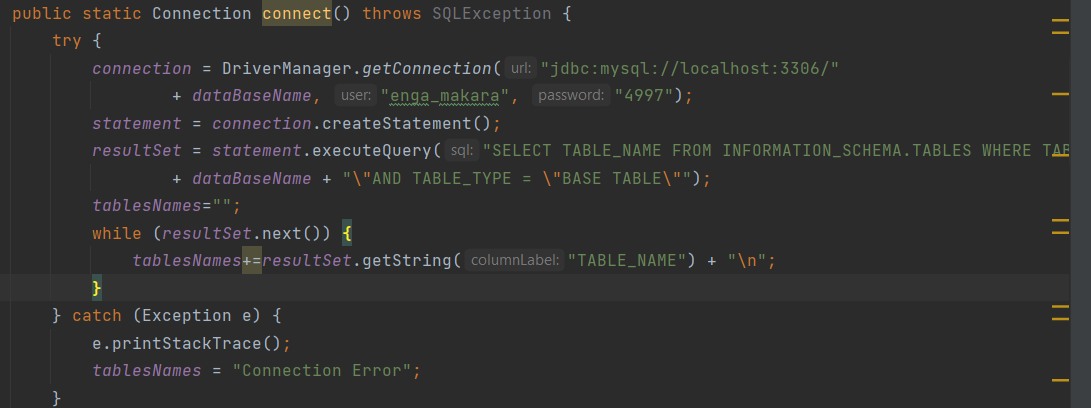


Рисунок 9 – Реализация метода connect

Для получения данных из таблицы по нажатию на кнопку Select вызывается метод select класса DB. В данный метод передается название таблицы, которое получается из текстового поля txtLIST. В методе select формируется запрос к БД для получения данных из желаемой таблицы. Реализация запроса приведена на рисунке 10.

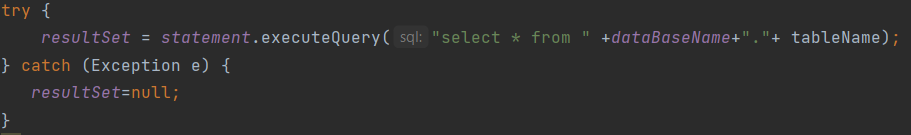


Рисунок 10 – Реализация запроса к БД

После получения ответа от БД происходит заполнение необходимого списка. В классе DB существуют три списка для хранения объектов трех созданных нами классов: Entrant, Education, Exams. В зависимости от введенного нами имени таблицы, заполнится определенный список. Пример заполнения списка для объектов класса Entrant приведен на рисунке 11.

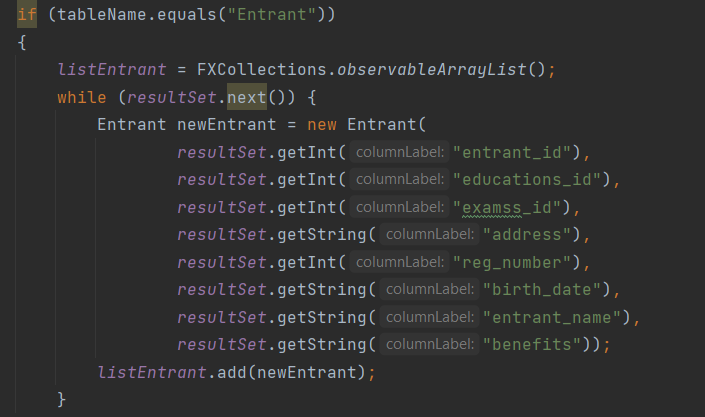


Рисунок 11 – Заполнение списка listEntrant

После заполнения необходимого списка, выполнение программы возвращается в функцию onSelectButtonClick, вызванную ранее нажатием кнопки Select. В зависимости от того, какое название таблицы было введено, в функции происходят следующие действия: происходит скрытие ненужных таблиц изменением атрибута visiable, отображение и заполнение значениями из списка желаемой таблицы. Часть кода функции onSelectButtonClick приведена на рисунке 12.

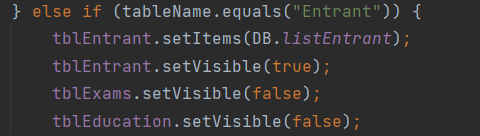


Рисунок 12 – Реализация onSelectButtonClick при выводе таблицы Entrant

На каждом этапе выполнения программы при вводе некорректных данных, ошибке подключения или отсутствии желаемых таблиц пользователь получает сообщение об ошибке. Пример работы оконного приложения приведен на рисунке 13.



Рисунок 13 – Пример корректной работы приложения

Пример ошибки подключения приведен на рисунке 14.



Рисунок 14 – Пример ошибки подключения