НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ФАКУЛЬТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

БАЗЫ ДАННЫХ

**Технический отчет по проекту**

**“Информационная система Вуза”**

Студент группы 17210

Борзов А. С.

Преподаватель

Казаков В.Г.

2020 г.

**ВВЕДЕНИЕ**

Описание системы:

Студенты, организованные в группы, учатся на одном из факультетов, возглавляемом деканатом, в

функции которого входит контроль за учебным процессом. В учебном процессе участвуют

преподаватели кафедр, административно относящиеся к одному из факультетов. Преподаватели

подразделяются на следующие категории: ассистенты, преподаватели, старшие преподаватели,

доценты, профессора. Ассистенты и преподаватели могут обучаться в аспирантуре, ст.

преподаватели, доценты, могут возглавлять научные темы, профессора - научные направления.

Преподаватели любой из категории в свое время могли защитить кандидатскую, а доценты и

профессора и докторскую диссертацию, при этом преподаватели могут занимать должности

доцента и профессора только, если они имеют соответственно звания доцента и профессора.

Учебный процесс регламентируется учебным планом, в котором указывается, какие учебные

дисциплины на каких курсах и в каких семестрах читаются для студентов каждого года набора, с

указанием количества часов на каждый вид занятий по дисциплине (виды занятий: лекции,

семинары, лабораторные работы, консультации, курсовые работы, ИР и т.д.) и формы контроля

(зачет, экзамен). Перед началом учебного семестра деканаты раздают на кафедры учебные

поручения, в которых указываются какие кафедры (не обязательно относящиеся к данному

факультету), какие дисциплины и для каких групп должны вести в очередном семестре.

Руководствуясь ими, на кафедрах осуществляется распределение нагрузки, при этом по одной

дисциплине в одной группе разные виды занятий могут вести один или несколько разных

преподавателей кафедры (с учетом категории преподавателей, например, ассистент не может

читать лекции, а профессор никогда не будет проводить лабораторные работы). Преподаватель

может вести занятия по одной или нескольким дисциплинам для студентов как своего, так и

других факультетов. Сведения о проведенных экзаменах и зачетах собираются деканатом.

По окончании обучения студент выполняет дипломную работу, руководителем которой является

преподаватель с кафедры, относящейся к тому же факультету, где обучается студент, при этом

преподаватель может руководить несколькими студентами.

Виды запросов в информационной системе:

1. Получить перечень и общее число студентов указанных групп либо указанного курса

(курсов) факультета полностью, по половому признаку, году рождения, возрасту, признаку

наличия детей, по признаку получения и размеру стипендии.

2. Получить список и общее число преподавателей указанных кафедр либо указанного

факультета полностью либо указанных категорий (ассистенты, доценты, профессора и т.д.)

по половому признаку, году рождения, возрасту, признаку наличия и количеству детей,

размеру заработной платы, являющихся аспирантами, защитивших кандидатские,

докторские диссертации в указанный период.

3. Получить перечень и общее число тем кандидатских и докторских диссертаций,

защитивших сотрудниками указанной кафедры либо указанного факультета.

4. Получить перечень кафедр, проводящих занятия в указанной группе либо на указанном

курсе указанного факультета в указанном семестре, либо за указанный период.

5. Получить список и общее число преподавателей, проводивших (проводящих) занятия по

указанной дисциплине в указанной группе либо на указанном курсе указанного факультета.

6. Получить перечень и общее число преподавателей проводивших (проводящих)

лекционные, семинарские и другие виды занятий в указанной группе либо на указанном

курсе указанного факультета в указанном семестре, либо за указанный период.

7. Получить список и общее число студентов указанных групп, сдавших зачет либо экзамен

по указанной дисциплине с указанной оценкой.

8. Получить список и общее число студентов указанных групп или указанного курса

указанного факультета, сдавших указанную сессию на отлично, без троек, без двоек.

9. Получить перечень преподавателей, принимающих (принимавших) экзамены в указанных

группах, по указанным дисциплинам, в указанном семестре.

10. Получить список студентов указанных групп, либо которым заданный преподаватель

поставил некоторую оценку за экзамен по определенным дисциплинам, в указанных

семестрах, за некоторый период.

11. Получить список студентов и тем дипломных работ, выполняемых ими на указанной

кафедре либо у указанного преподавателя.

12. Получить список руководителей дипломных работ с указанной кафедры, либо факультета

полностью и раздельно по некоторым категориям преподавателей.

13. Получить нагрузку преподавателей (название дисциплины, количество часов), ее объем по

отдельным видам занятий и общую нагрузку в указанном семестре для конкретного

преподавателя либо для преподавателей указанной кафедры.

Цель: Разработать информационную систему вуза в рамках описанных бизнес процессов.

Задачи:

1. Анализ системы:
2. Описание основных сущностей и отношений, для описания системы.
3. Выявление требований к обеспечению целостности данных, и основных путях обеспечения их выполнения
4. Выявление основных ролей пользователей приложения и основных сценариев использования

2) Проектирование системы

1. Определение общей архитектуры приложения и ее основные части
2. Построение алгоритмов обеспечения целостности данных
3. Определение основных форм приложения для взаимодействия системы с пользователями.
4. Реализация системы
   1. Описание структуры проекта
   2. Определение используемых технологий
   3. Описание реализации пользовательского интерфейса

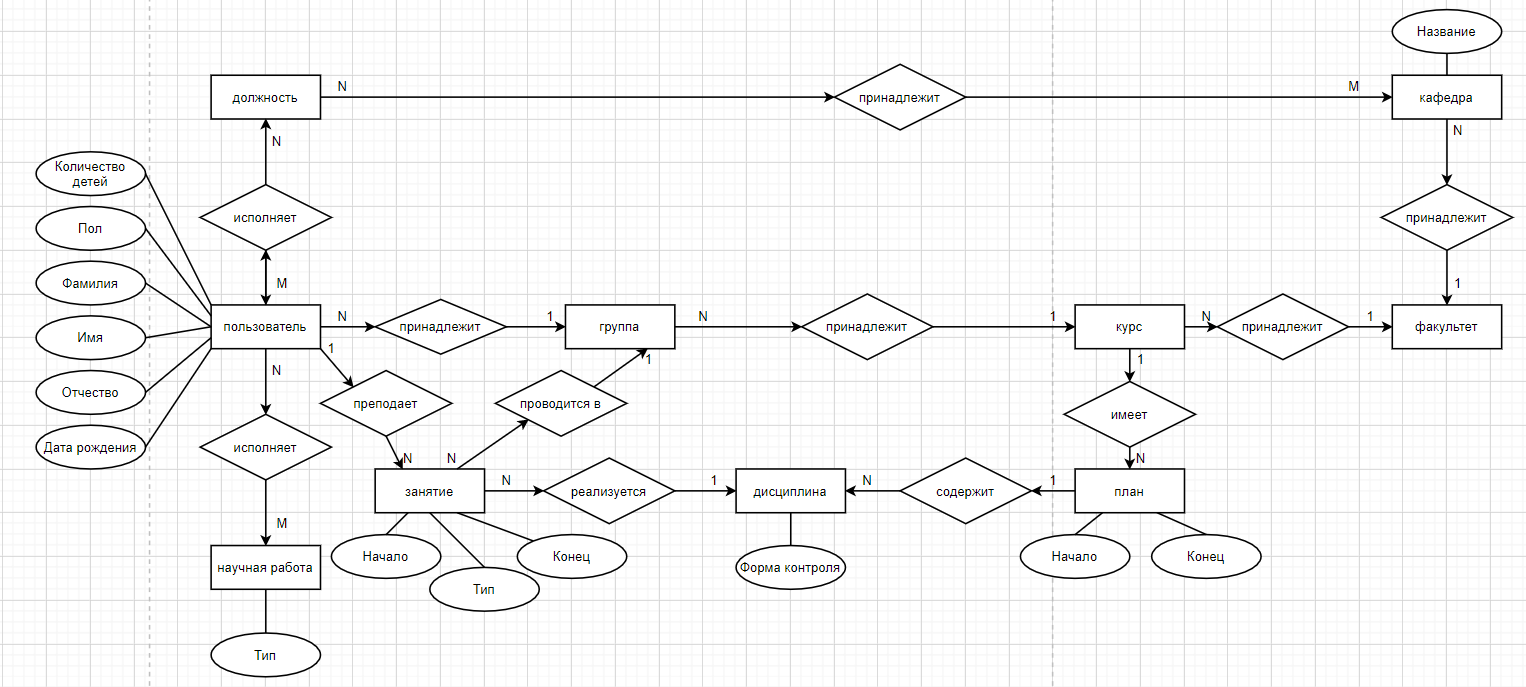
4) Тестирование системы

**Анализ системы**

В результате анализа системы были выделены следующие основные сущности:

* Пользователь
* Группа
* Курс
* План
* Дисциплина
* Занятие
* Научная работа
* Должность
* Кафедра
* Факультет

Взаимодействия между ними описаны в ER-модели:



Обеспечение целостности данных:

* У Пользователя должны быть заполнены поля Имя и Фамилия.
* У Научной работы должен быть Тип и, хотя бы один Автор.
* У Дисциплины должна быть форма контроля.
* Дисциплина должна быть привязана к курсу.
* Должность должна принадлежать кафедре.
* Занятие должно проводиться в определенной группе.
* У Занятия должны быть заполнены все поля.
* Группа обязана принадлежать курсу.
* Курс обязан принадлежать факультету.
* Кафедра обязана принадлежать Факультету.
* План должен принадлежать какому-нибудь курсу.
* В таблицах, созданных для обеспечения связи «многие ко многим» должны быть заполнены все поля.
* Ассистент не может вести лекции
* Занятия могу вести только преподаватели
* Контроль дисциплины могут осуществлять только преподаватели

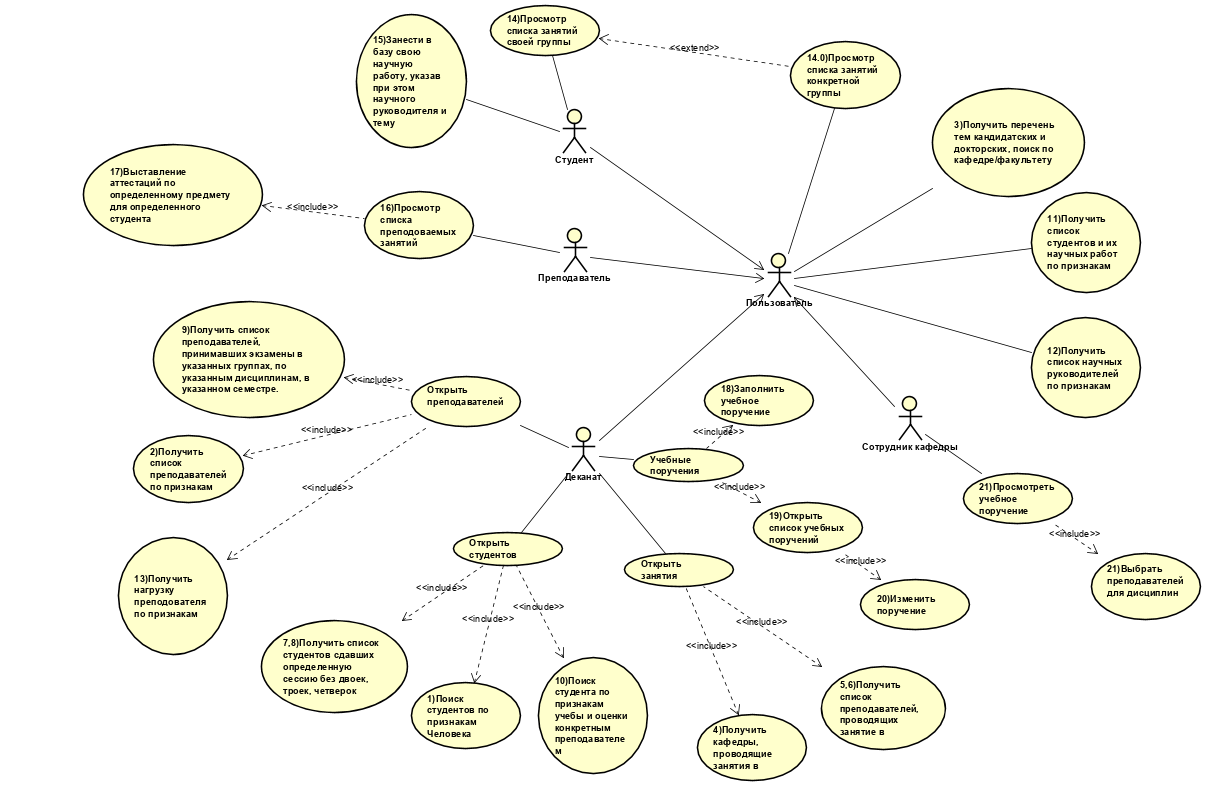
**Решение:**

Большая часть требований обеспечивается при помощи ключевых слов NOT NULL, UNIQUE, PRIMARY KEY. Контроль должности обеспечивается (будет обеспечиваться) при помощи тригеров проверяющих совпадение должности с требуемой.

Прецеденты:

Роли:

* Пользователь
* Преподаватель
* Сотрудник Деканата
* Сотрудник Кафедры
* Студент



Прецеденты описаны в диаграмме прецедентов:

(В приложении приведена картинка в высоком качестве)

**Проектирование системы**

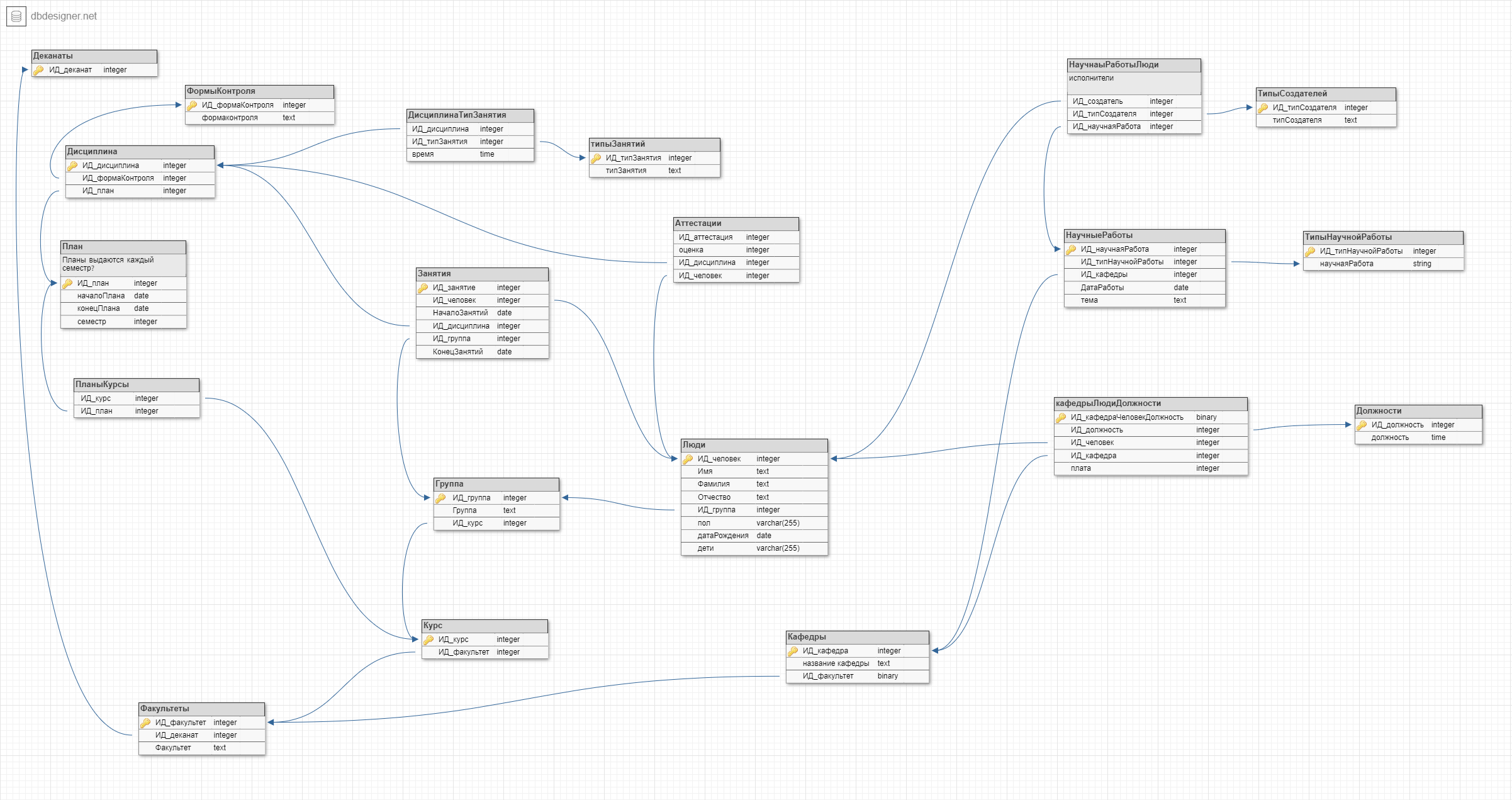
Для реализации приложения была выбрана схема разделения классов на model, dao, utils, configuration, controller и view.

Для реализации пользовательского интерфейса была выбрана библиотека Swing.

Всё взаимодействие с базой данный инкапсулировано в отдельном классе, который в свою очередь использует Oracle JDBC.

Список таблиц в базе данных:

* Деканаты
* Факультеты
* Дисциплины
* Планы
* ПланыКурсы (для обеспечения связи многие ко многим)
* ФормыКонтроля (для обеспечения связи многие ко многим)
* ДисциплинаТипЗанятия (для обеспечения связи многие ко многим)
* Занятия
* ТипыЗанятий
* Аттестации
* Люди
* НаучныеРаботыЛюди (для обеспечения связи многие ко многим)
* ТипыСоздателей
* НаучныеРаботы
* ТипыНаучнойРаботы (для обеспечения связи многие ко многим)
* КафедрыЛюдиДолжности (для обеспечения связи многие ко многим)
* Должности
* Группы
* Курсы
* Факультеты
* Кафедры

Схема базы данных:

(в приложении приведена картинка в высоком качестве)

**Алгоритмы целостности данных**

В клиентском приложении производятся основные проверки вводимых данных, такие как проверки на пустые поля и формат ввода.

В СУБД обеспечение целостности данных обеспечивается за счет NOT NULL, UNIQUE, PRIMARY KEY. При добавлении новых записей в таблицу срабатывает триггер автоинкримента, что избавляет пользователя от ручного ввода идентификатора. Также используется PL SQL для проверки что пользователь имеет достаточную должность для того чтобы быть присвоенным в некоторые поля.

В Java форма представляет из себя набор элементов интерфейса с которыми возможно взаимодействие. Реакция на события, возникающие при работе с формой (нажатия, перетаскивания и т. д.) определены в классе, реализующем поведение (Контроллер).

Описание основных форм интерфейса:

(также приведены в приложении)

Студент:





Преподаватель:





Деканат:















Пользователь:







Кафедра:



Классы разбиты на пакеты:

1) для взаимодействия с базой данных (dao)

2)для определения данных (model)

3)управления пользовательским интерфейсом (controller)

4)для построения пользовательского интерфейса (view)

5)Утилиты для построения пользовательского интерфейса (utils)

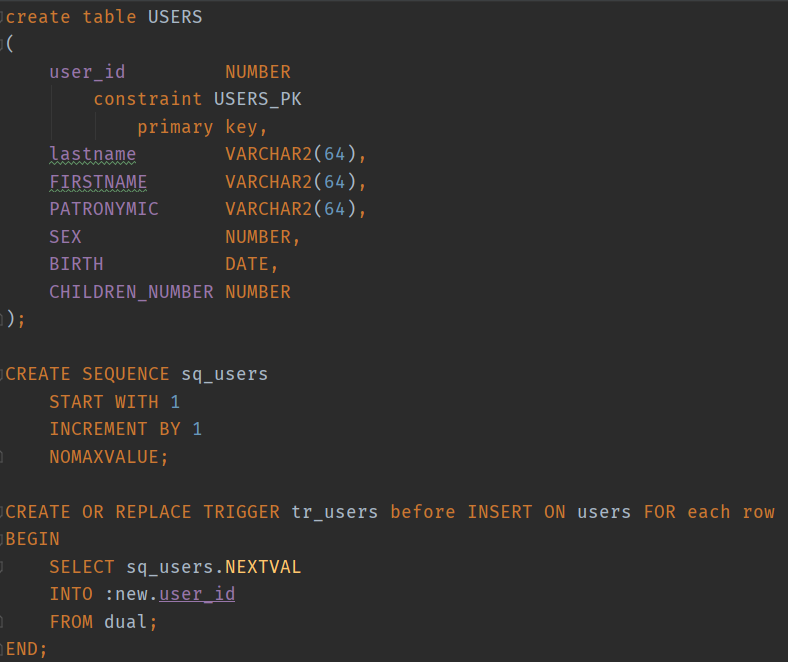
Каждый из пакетов также содержит пакеты для каждой роли:

* User
* Teacher
* Student
* Deanery
* Department

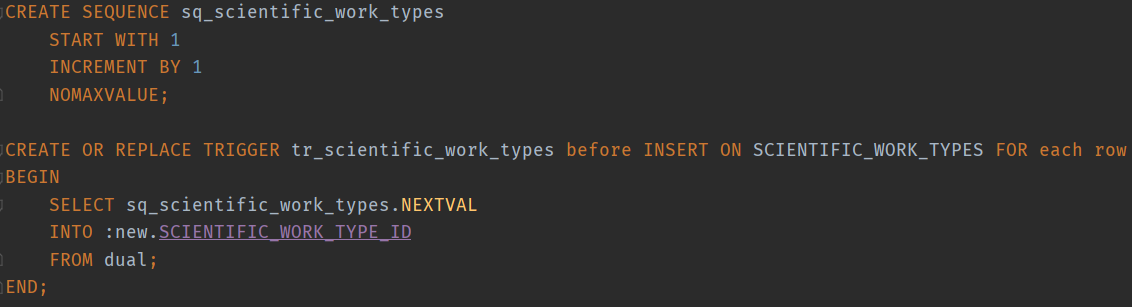
Для создания пользовательского интерфейса используется библиотека Swing.

Для взаимодействия с сервером используется библиотека Oracle OJDBC.

Пример создания таблицы:

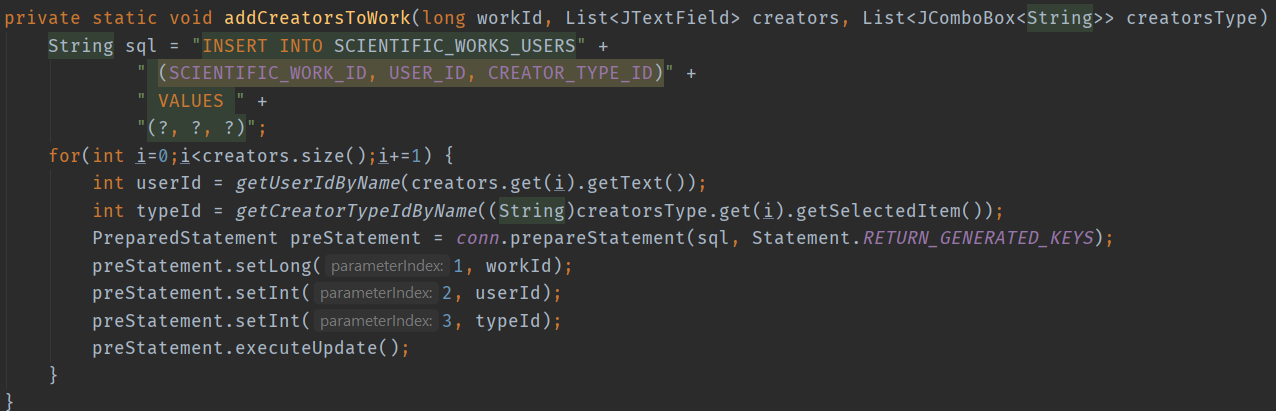


PL SQL скрипт используется для автоинкримента:



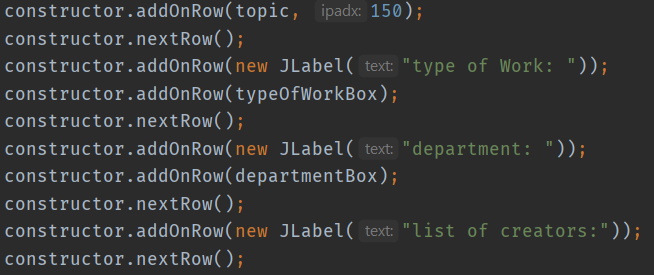
Подключение к БД: 

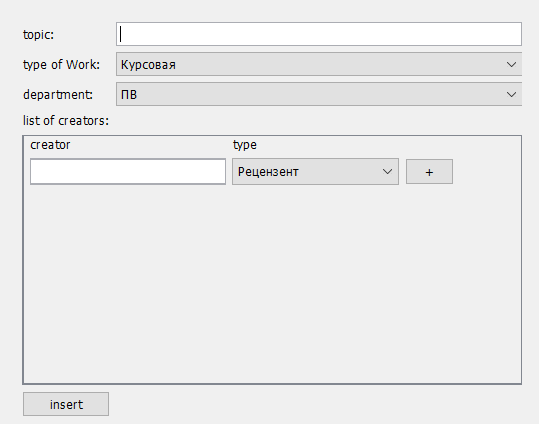
Пример запроса на добавление:

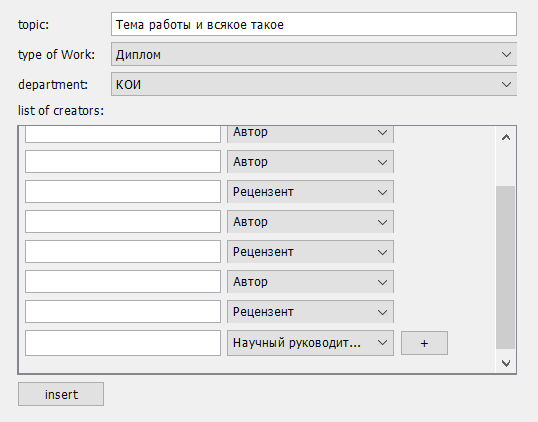


Как можно видеть на картинке, в качестве базовой защиты от sql инъекций используются параметризованные запросы

Пример использования класса, упрощающего создания форм:



В результате появляется возможность быстро создавать вложенные формы со сложной логикой и слайдерами, при этом избегая копипасты:



**Тестирование**

Для тестирования интерфейса все таблицы были заполнены основными данными.

В доработке.

**Вывод**

В доработке.