Министерство образования, науки и молодежной политики

Нижегородской области

Государственное бюджетное профессиональное образовательное учреждение

НИЖЕГОРОДСКИЙ РАДИОТЕХНИЧЕСКИЙ КОЛЛЕДЖ

**Специальность 09.02.07 Информационные системы и программирование**

**Квалификация Специалист по информационным системам**

**КУРСОВАЯ РАБОТА**

[МДК 02.02 Инструментальные средства разработки программного обеспечения](https://moodle.nntc.nnov.ru/course/view.php?id=1049)

**Тема: «разработка программного модуля «Оформление подписки на печатное издание» для информационной системы «Печатное издание»**»

|  |  |  |
| --- | --- | --- |
| Выполнил обучающийся  группы 3ИСиП-17-1  \_\_\_\_\_\_\_\_\_\_\_  ФИО |  | Проверил преподаватель Гутянская Е.М.  Проект защищен с оценкой  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Дата защиты\_\_\_\_\_\_\_\_\_\_\_\_  Подпись\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

2019 г.

**ЗАДАНИЕ**

**на курсовую работу**

**Специальность 09.02.07 Информационные системы и программирование**

**Квалификация Специалист по информационным системам.**

[МДК 02.02 Инструментальные средства разработки программного обеспечения](https://moodle.nntc.nnov.ru/course/view.php?id=1049)

Обучающемуся \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Группа 3ИСиП-17-1

**Тема курсовой работы: разработка программного модуля «Оформление подписки на печатное издание» для информационной системы «Печатное издание»**

**Функциональные требования к разрабатываемой системе**

1. Вывод популярных изданий.
2. Оформление подписки на издание.

Дата выдачи задания « » 201 г.

Срок сдачи работы « » 201 г.

**Перечень вопросов, подлежащих разработке:**

* + - Введение (название выбранной темы, обзор раскрываемых вопросов).
    - Разработка технического задания.
    - Проектирование информационной системы.
    - Разработка информационной системы.
    - Тестирование информационной системы.
    - Руководство пользователя.
    - Заключение.
    - Список использованной литературы.

**Перечень диаграмм проектирования:**

1. Информационно-логическая модель базы данных.
2. Диаграмма прецедентов. Сценарий варианта использования.
3. Диаграмма классов.
4. Диаграмма последовательностей.
5. Диаграмма компонентов.

6). Диаграмма состояний

**Задание выдал преподаватель** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

/Е.М. Гутянская/

Задание принял обучающийся\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Содержание**

**Введение**

# Развитие средств вычислительной техники обеспечило почву для создания и широкого использования систем обработки данных разнообразного назначения. Разрабатываются информационные системы для обслуживания различных систем деятельности, систем управления хозяйственными и техническими объектами, модельные комплексы для научных исследований, системы автоматизации проектирования и производства, всевозможные тренажеры и обучающие системы. Одной из важных предпосылок создания таких систем стала возможность оснащения их «памятью» для накопления, хранения и систематизация больших объемов данных. Другой существенной предпосылкой нужно признать разработку подходов, а также создание программных и технических средств конструирования систем, предназначенных для коллективного пользования. В этой связи потребовалось разработать специальные методы и механизмы управления такого рода совместно используемыми ресурсами данных, которые стали называться базами данных. Исследования и разработки, связанные с проектированием, созданием и эксплуатации баз данных, а также необходимых для этих целей языковых и программных инструментальных средств, привели к появлению самостоятельной ветви информатики, получившей название системы управления данными.

# Цель курсового проекта состоит в создании информационной системы (далее ИС) «Печатное издание». Как правило, городские отделения связи по всей стране все еще не избавились от пережитков прошлого, и продолжают пользоваться кучей ненужных бумаг. Во многом для экономии времени, рабочих ресурсов и средств почтового отделения, и разрабатывалась данная ИС.

**1.Анализ предметной области**

В качестве предметной области рассматривается работа оформлении и доставки печатных изданий

Проектируемая ИС разрабатывается для организаций работающих в сфере почтовых рассылок, таких как городские почтовые отделения. Деятельность организации направлена на: сведения по изданиям, сведения о подписчике, оформление подписок на издания, сведения о доставке.

Целью составления ИС «печатное издание» является более эффективное обслуживание клиентов, учёт подписок на издания. Пользователями разрабатываемой базы данных предполагается администратор, клиент.

**1.2 Обоснование выбора средств разработки информационной системы**

В данном проекте для разработки информационной и реализации сервера базы данных был выбран язык программирования Java, а для реализации базы данных — СУБД MySQL.

СУБД MySQL появилась в 1995 году и до сих пор обновляется, имеет бесплатный тип распространения. Для ИС «Печатное издание», не имеющей больших оборотов данных, данная СУБД подходит лучше всего.

Язык программирования Java-  типизированный, объектно-ориентированный, С-подобный язык программирования. Программы, написанные на этом языке можно использовать на большинстве операционных систем.

Также для разработки ИС использована Интегрированная Среда Разработки NetBeans 11.0, в которой был разработан интерфейс программы. В NetBeans реализовано соединение между программным кодом, написанным на Java и базой данных, созданной на языке MySQL.

**2.Разработка технического задания**

**2.1 Назначение и цели создания системы**

ИС «печатное издание» создаётся для более эффективного обслуживания клиентов и учёта подписок на издания.

**2.2. Основания для разработки**

Основанием для разработки БД является задание на курсовое проектирование по междисциплинарному курсу МДК.02.02 «Инструментальные средства разработки программного обеспечения».

**2.3 Исходные данные**

* Сведения по изданиям: будет содержать данные об издании, цены издания.
* Сведения о подписчике: будет содержать данные фамилии, имя и отчества подписчика и адреса.
* Оформление подписок на издания: будет содержать данные об даты подписки, количество подписки и суммы подписки.
* Сведения о доставке: будет содержать данные о дате доставки и id подписки на издание.
* Сведения о данных для входа: содержит данные для входа в систему, доступны для просмотра и редактирования только администратором системы.

**2.4.1 Требования к функциональным характеристикам**

Программа должна выполнять следующие функции:

* Ведение базы данных.
* Сведения о пользователях системы.
* Оформление подписки на издание.
* Вывод статистики о популярных изданиях.
* Разграничение прав доступа в системе.

**2.4.1.1 Разграничение доступа к информации.**

Доступ к информации должен предоставляться только авторизованным пользователям по логину и паролю с соответствующими правами доступа. Права на соответствующий доступ определяются системным администратором в соответствии с должностными инструкциями.

**3.Проектикрование информационной системы**

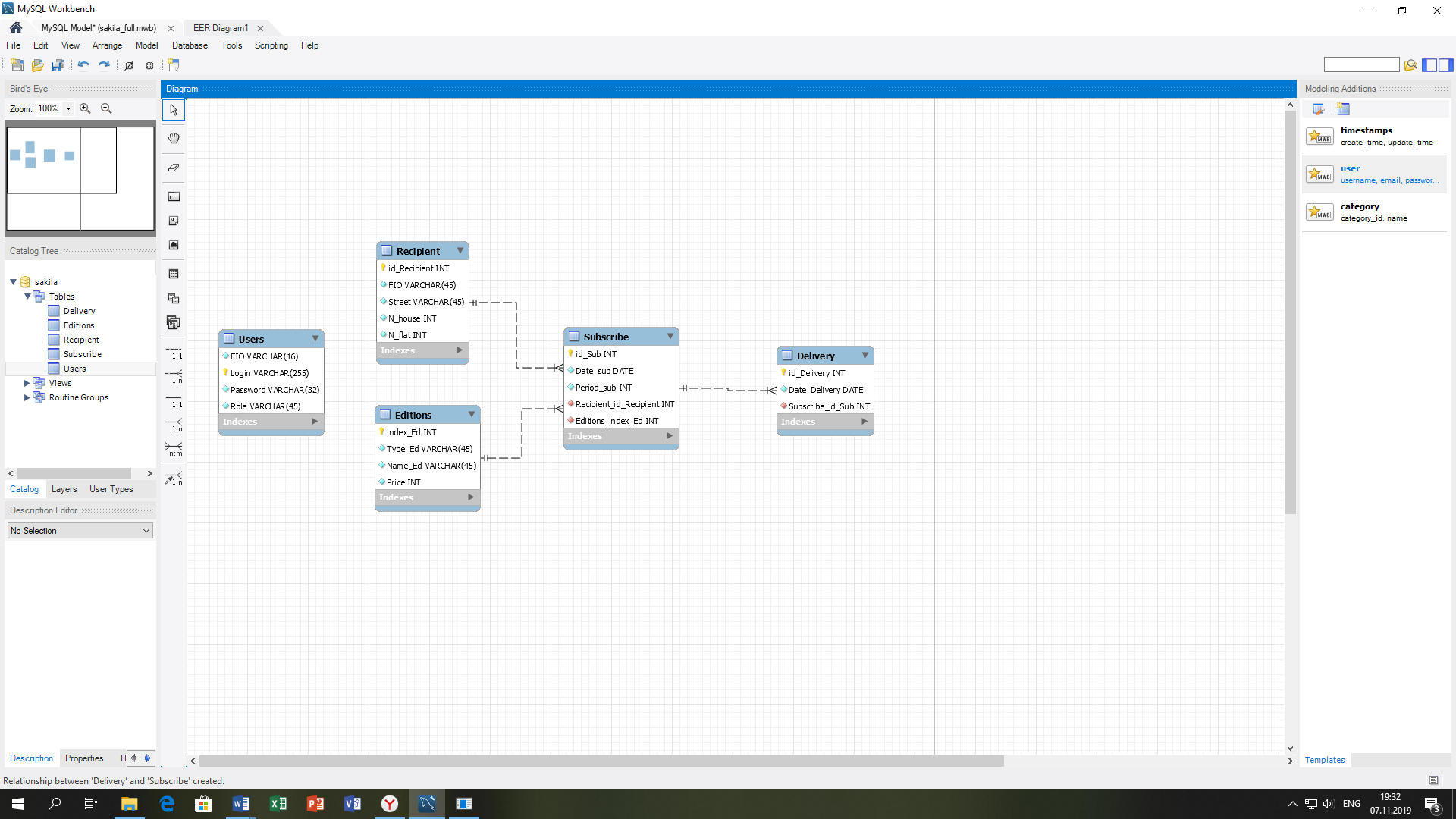
**3.1 Информационно-логическая модель базы данных (ERD-модель)**

Рис.1. ERD-модель.

**3.2 Словарь данных**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Издания** | | | | | |
| **Ключ** | **Поле** | | **Обязательное** | | **Примечание** |
| Первичный | Индекс издания | | Да | | Индекс издания |
|  | Тип издания | | Да | | Газета или журнал |
|  | Название издания | | Да | | Название издания |
|  | Цена | | Да | | Цена за 1 штуку |
| **Получатели** | | | | | |
| **Ключ** | **Поле** | | **Обязательное** | | **Примечание** |
| Первичный | Id получателя | | Да | | Идентификатор получателя |
|  | ФИО | | Да | | ФИО получателя |
|  | Улица | | Да | | Улица проживания |
|  | Номер дома | | Да | | Номер дома проживания |
|  | Номер квартиры | | Да | | Номер квартиры проживания |
| **Подписка** | | | | | |
| **Ключ** | **Поле** | | **Обязательное** | | **Примечание** |
| Первичный | Id подписки | | Да | | Идентификатор подписки |
|  | Дата подписки | | Да | | Дата начала подписки |
|  | Срок подписки | | Да | | Срок подписки |
| Внешний | Id получателя | | Да | | Идентификатор получателя |
| Внешний | Индекс издания | | Да | | Индекс издания |
| **Доставка** | | | | | |
| **Ключ** | **Поле** | | **Обязательное** | | **Примечание** |
| Первичный | Id Доставки | | Да | | Идентификатор доставки |
|  | Дата доставки | | Да | | Дата доставки |
| Внешний | Id подписки | | Да | | Идентификатор подписки |
| **Пользователи** | | | | | |
| **Ключ** | **Поле** | **Обязательное** | | **Примечание** | |
| Первичный | ФИО пользователя | Да | | Фамилия, имя, отчество пользователя | |
|  | Логин | Да | | Логин пользователя | |
|  | Пароль | Да | | Пароль пользователя | |
|  | Роль | Да | | Роль пользователя в системе | |

**3.3 Диаграмма прецедентов**

Используется для представления функциональной модели информационной системы.

Прецеденты (варианты использования — Use Cases) — это подробные процедурные описания вариантов использования системы всеми заинтересованными лицами, а также внешними системами, т. е. всеми, кто (или что) может рассматриваться как актёры (actors) — действующие лица. По сути, это своего рода алгоритмы работы с системой с точки зрения внешнего мира.

Прецеденты являются основой функциональных требований к системе, позволяют описывать границы проектируемой системы, ее интерфейс, а затем выступают как основа для тестирования системы заказчиком с помощью приемочных тестов

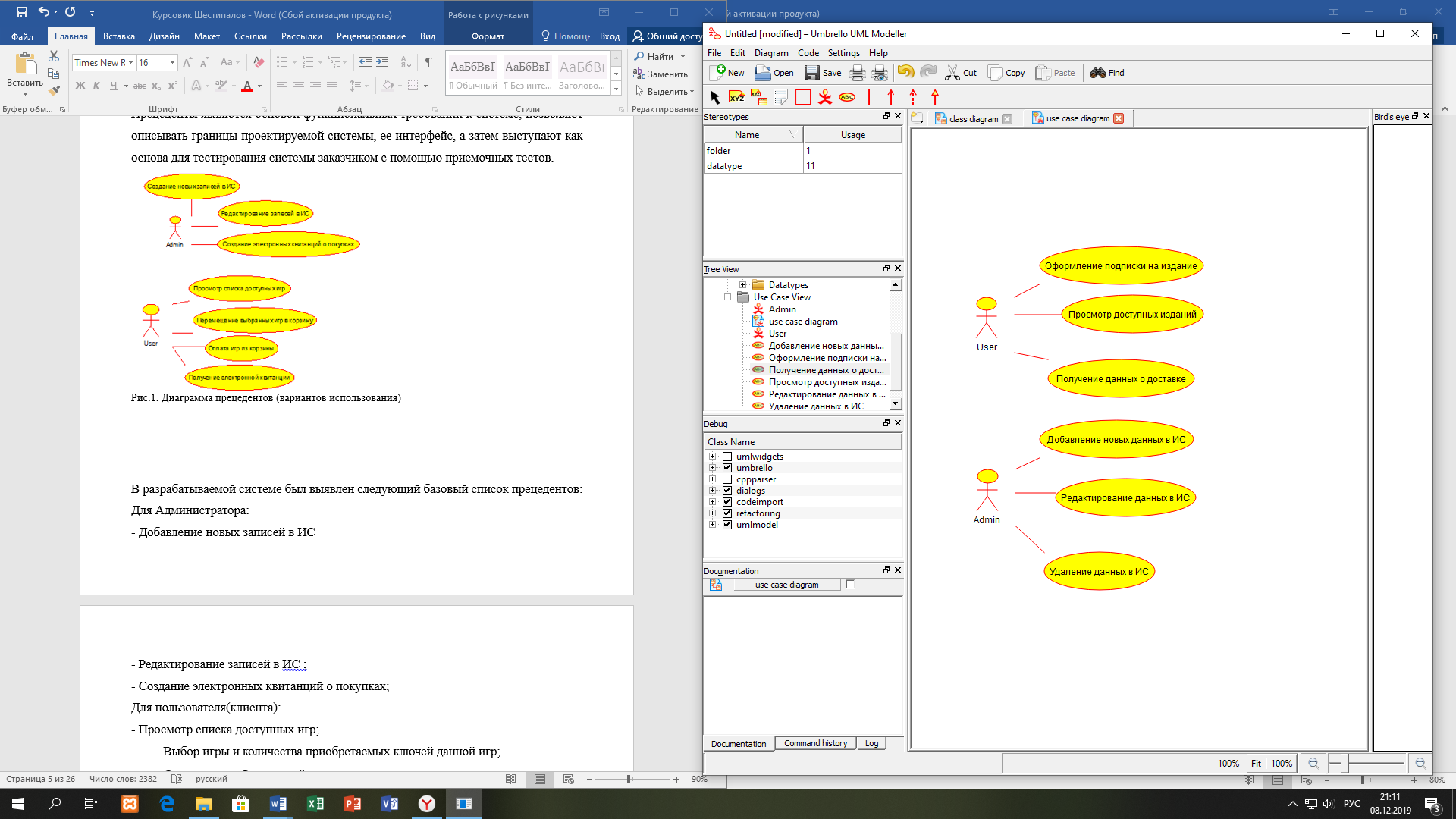


Рис.2. Диаграмма прецедентов

**Основной сценарий прецедента "Работа с записями базы данных":**

1). Войти в систему под своим логином или паролем.

2). Система проверяет правильность ввода логина и пароля.

3). Пользователь выбирает нужную таблицу с данными.

4). Система выводит список записей из базы данных.

5). В зависимости от роли доступны следующие действия:

а). Для пользователя- нахождение записи по ключу;

б). Для администратора- поиск, добавление, редактирование и удаление записей из базы данных.

**Альтернативный сценарий 2а "Неверный логин или пароль":**

1). Система предотвращает вход.

2). Выводит сообщение "неверный логин или пароль".

3). Пользователь переходит на начало основного сценария.

**3.4 Диаграмма классов**

Диаграмма классов (англ. Static Structure diagram) — структурная диаграмма языка моделирования UML, демонстрирующая общую структуру иерархии классов системы, их атрибутов(полей), методов, интерфейсов и взаимосвязей между ними.

Диаграмма представлена в разделе Приложение(Рис.3).

**3.5 Диаграмма последовательностей**

Диаграмма последовательности — диаграмма, на которой показано взаимодействие объектов (обмен между ними сигналами и сообщениями), упорядоченное по времени, с отражением продолжительности обработки и последовательности их проявления.

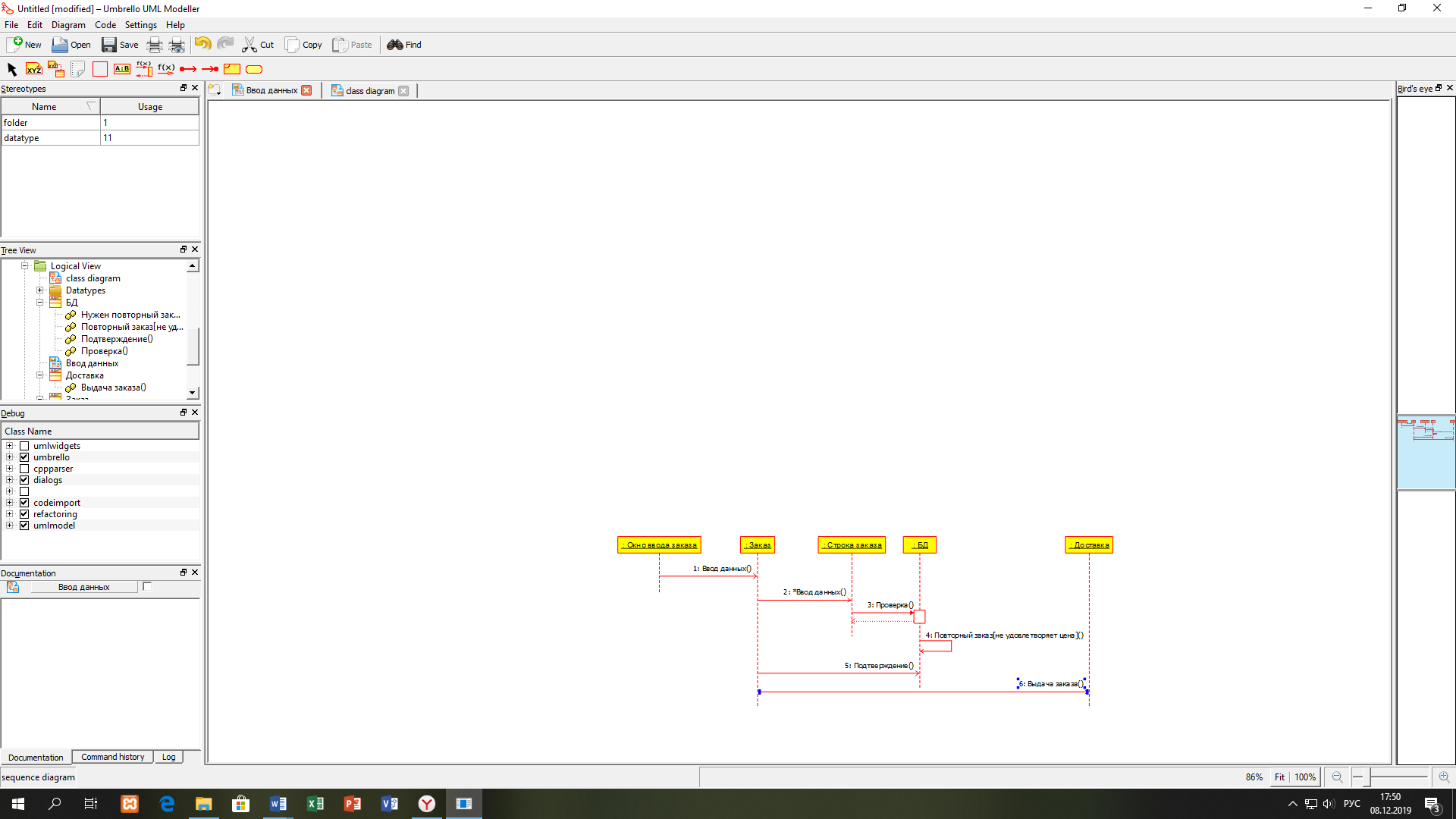
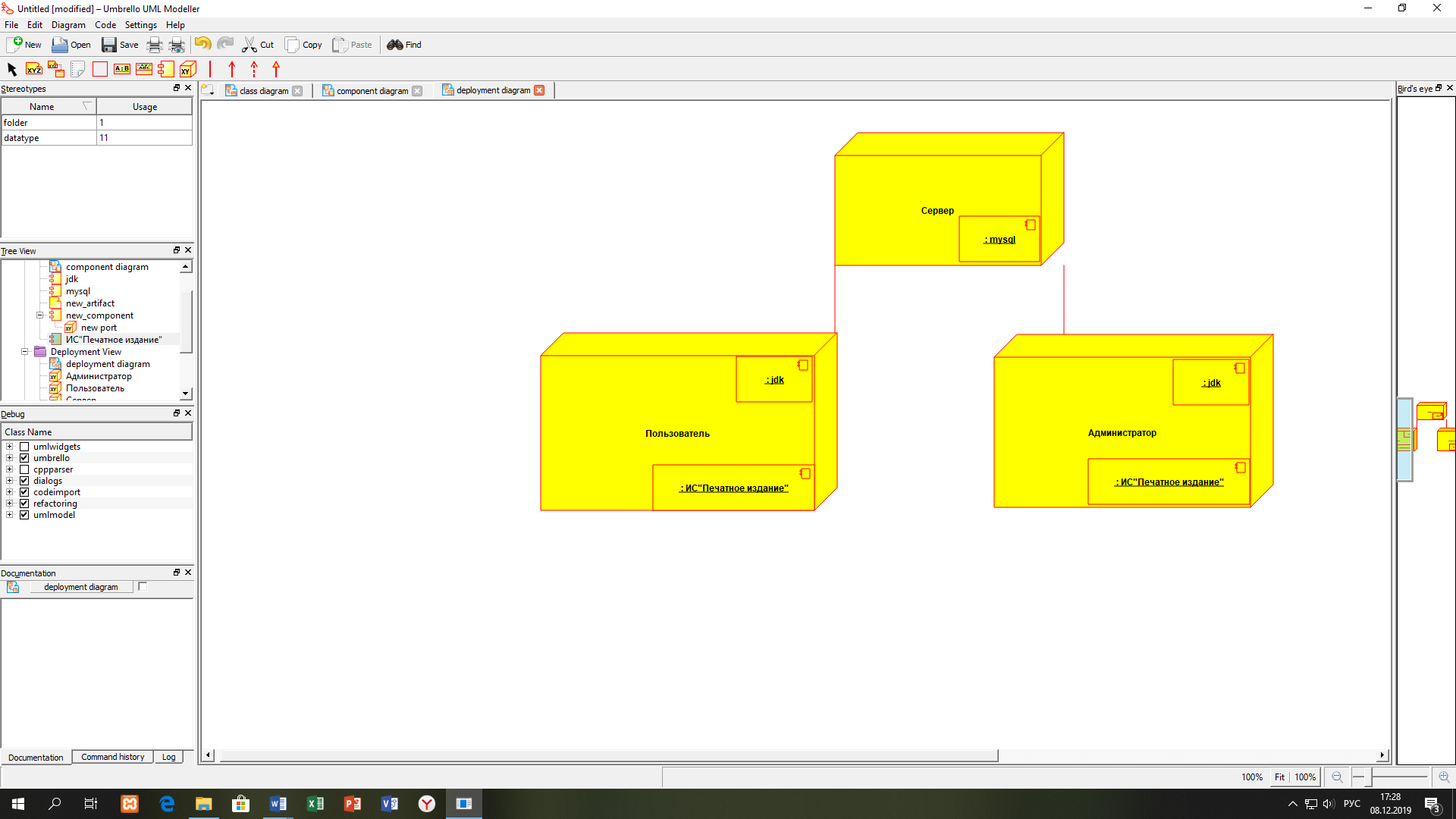


Рис.4. Диаграмма последовательностей.

**3.6. Диаграмма Компонентов**

Диаграммы компонентов - это один из двух видов диаграмм, применяемых при моделировании физических аспектов объектно-ориентированной системы. Они показывают организацию наборов компонентов и зависимости между ними.

Рис.5. Диаграмма компонентов.

**3.7 Диаграмма состояний**

Диаграмма состояний (state diagram) определяет все возможные состояния, в которых может находиться конкретный объект, а также процесс смены состояний объекта в результате влияния некоторых событий. Диаграммы состояний строятся для единственного класса и описывают поведение единственного объекта.

Диаграмма состояний представляет собой граф состояний в которых может находиться объект и связей между ними.

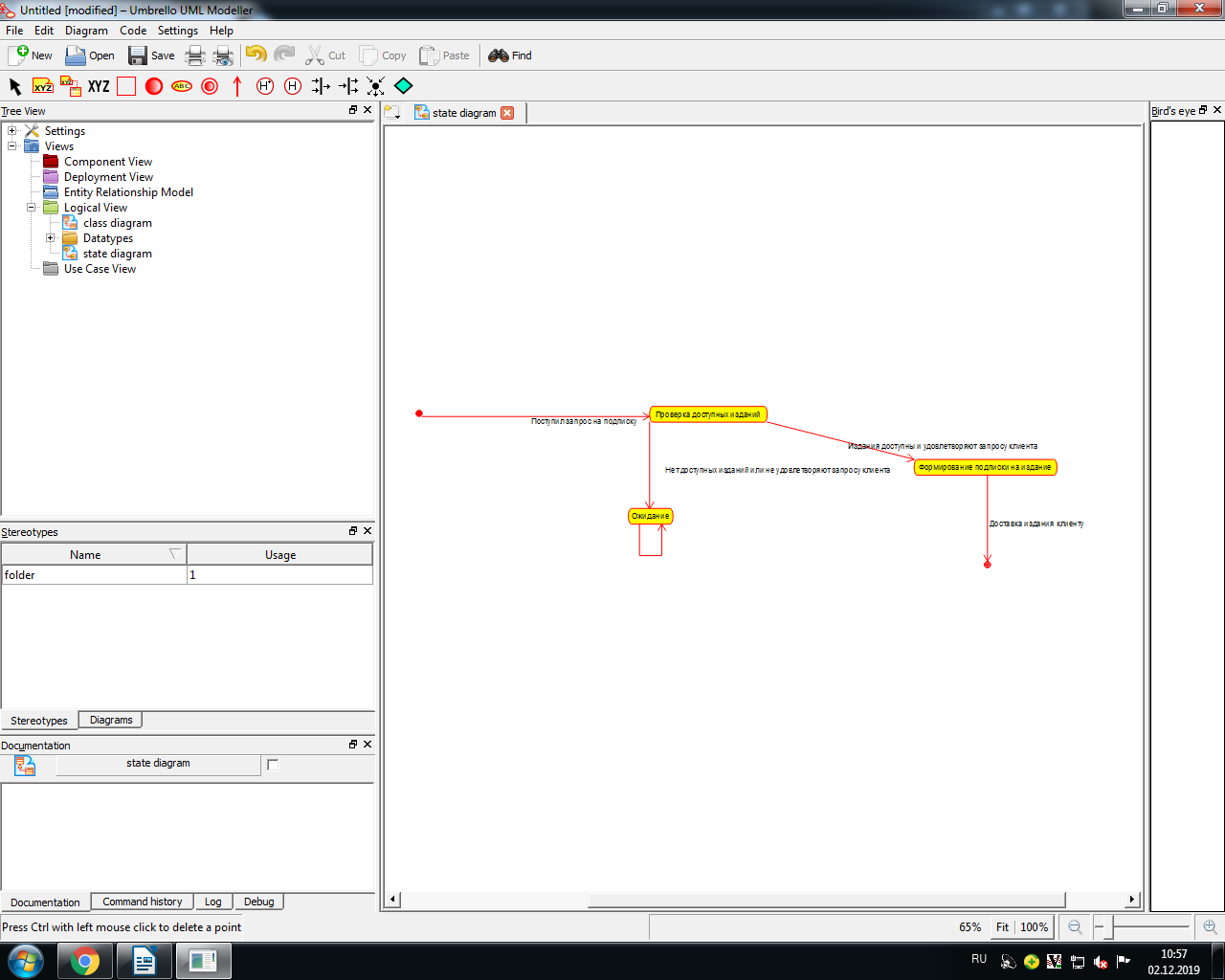


Рис.6. Диаграмма состояний

**4.Разработка информационной системы.**

**4.1 Описание структуры информационной системы**

При запуске приложения пользователь видит форму авторизации, содержащую в себе два поля (логин и пароль) и кнопки «Вход» и «Регистрация». При успешной авторизации пользователю открывается доступ к интерфейсу, если пользователь не зарегистрирован то открывается форма регистрации после чего необходимо пройти авторизацию. На появившейся после успешной авторизации форме пользователь может выбрать доступные ему пункты меню для переключения между формами. По нажатию на каждый из пунктов меню, отвечающий за свои функции, открывается выбранная форма. Для разных ролей пользователя реализованы свои меню, то есть для пользователя с ролью User реализовано соответствующее меню, соответственно для пользователя с ролью Admin реализовано отдельное меню;

**4.2 Разработка графического пользовательского интерфейса**

Для создания графического интерфейса использовался пакет Swing, так как является основой для большинства современных графический приложений Java.

* Интерфейс меню авторизации
* Интерфейс меню регистрации
* Интерфейс меню пользователя с ролью user
* Интерфейс меню пользователя с ролью admin

Интерфейсы для пользователя:

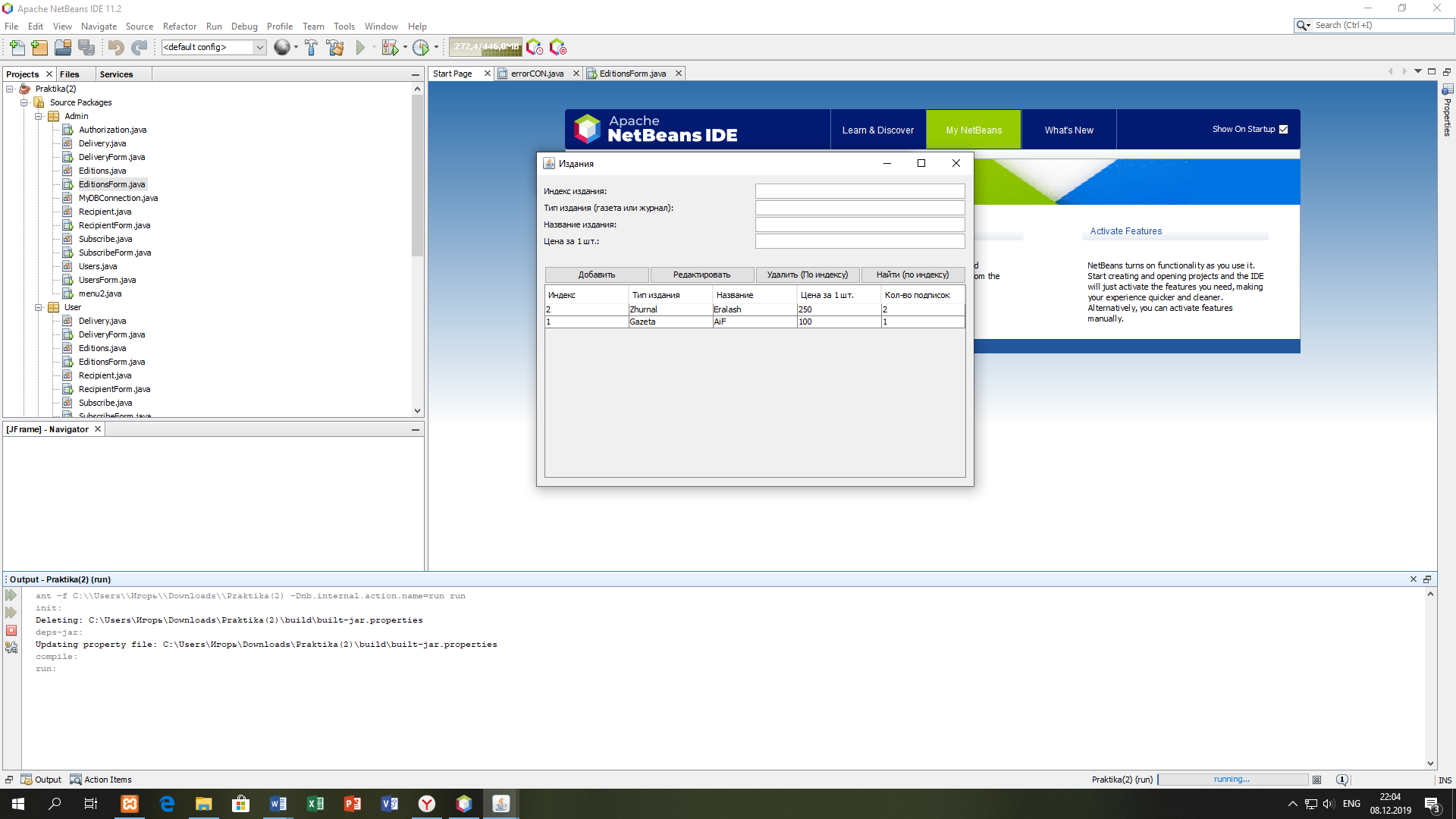
* Интерфейс меню Издания
* Интерфейс меню Подписки
* Интерфейс меню Учётная запись
* Интерфейс меню Доставка

Интерфейсы для администратора:

* Интерфейс меню Данные для входа
* Интерфейс меню Издания
* Интерфейс меню Подписки
* Интерфейс меню Данные пользователей
* Интерфейс меню Доставка

Задание 1

Реализовать вывод популярных изданий (большее кол-во подписок на данный момент)



**Код реализации**

public class EditionsForm extends javax.swing.JFrame {

private ResultSet rs;

private boolean insertStr;

/\*\*

\* Creates new form SubscribeForm

\* @throws java.lang.Exception

\*/

public EditionsForm () throws Exception{

mdbc=new MyDBConnection();

mdbc.init();

Connection conn=mdbc.getMyConnection();

stmt= conn.createStatement();

initComponents();

}

public ResultSet getResultFromEd() {

ResultSet rs=null;

try{

rs=stmt.executeQuery("Select \* from Editions ORDER BY `Editions`.`cnt` DESC"); }

catch(SQLException e){

this.dispose();

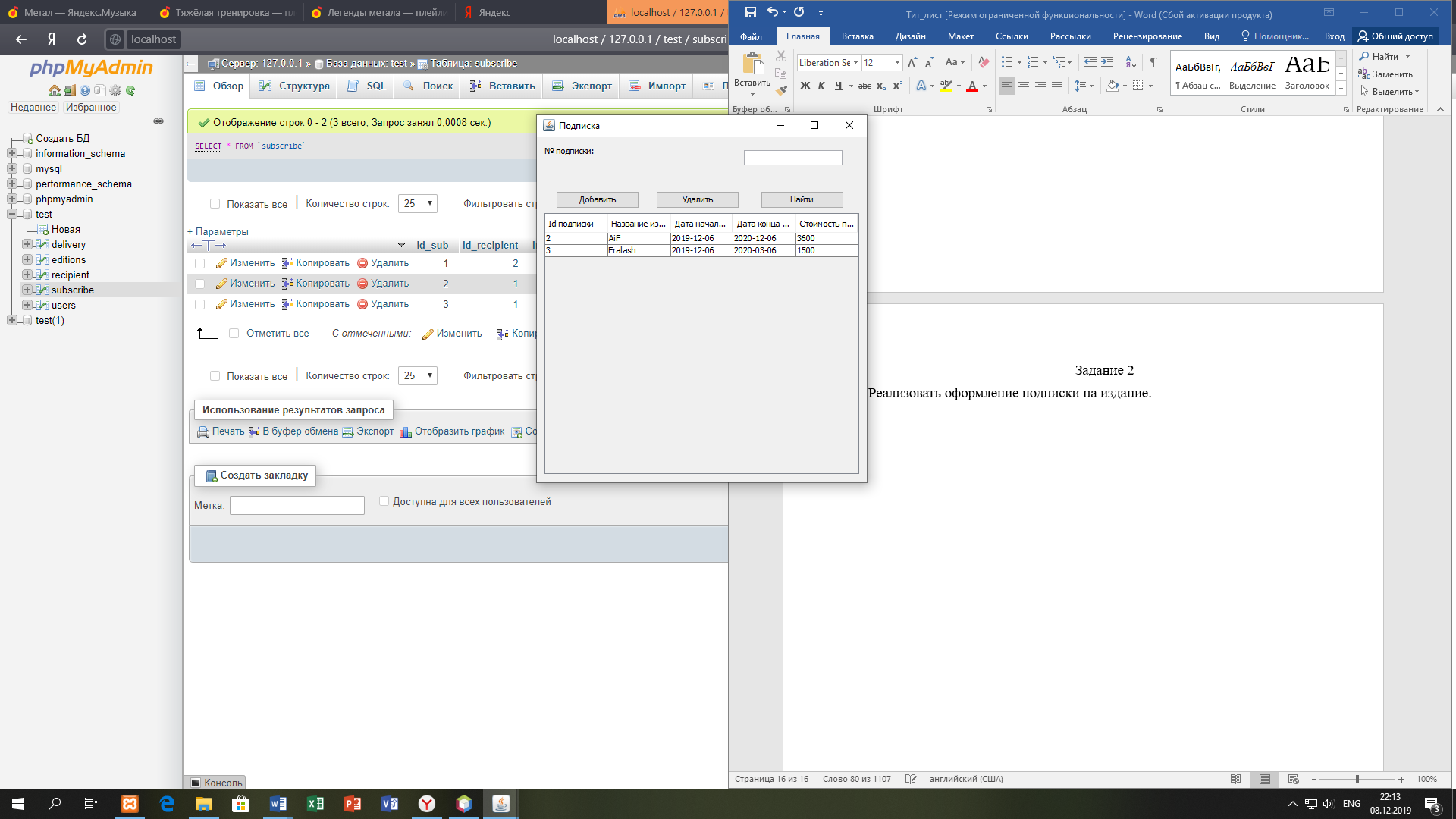
new message.errorCON().setVisible(true);

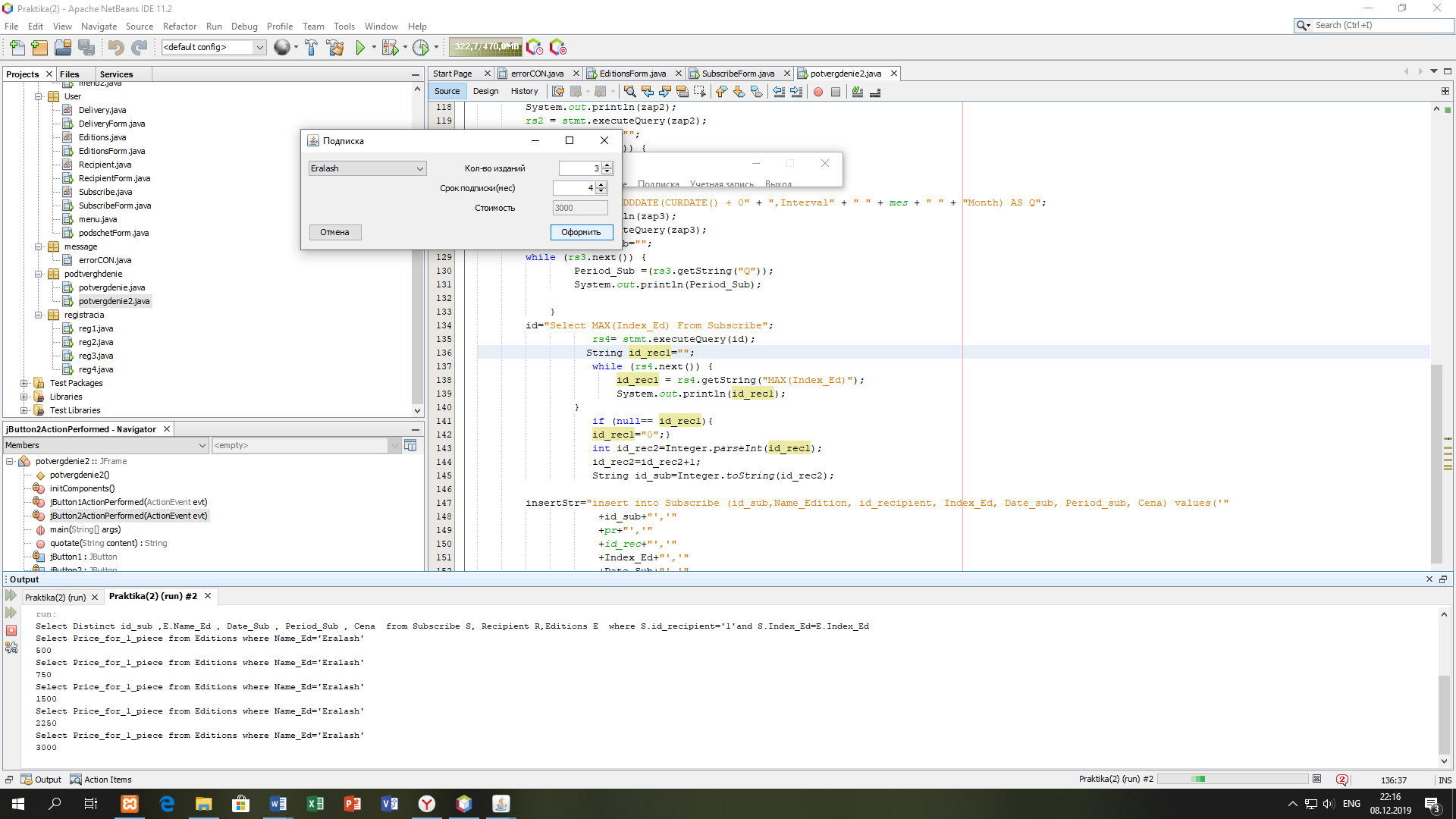
} return rs;

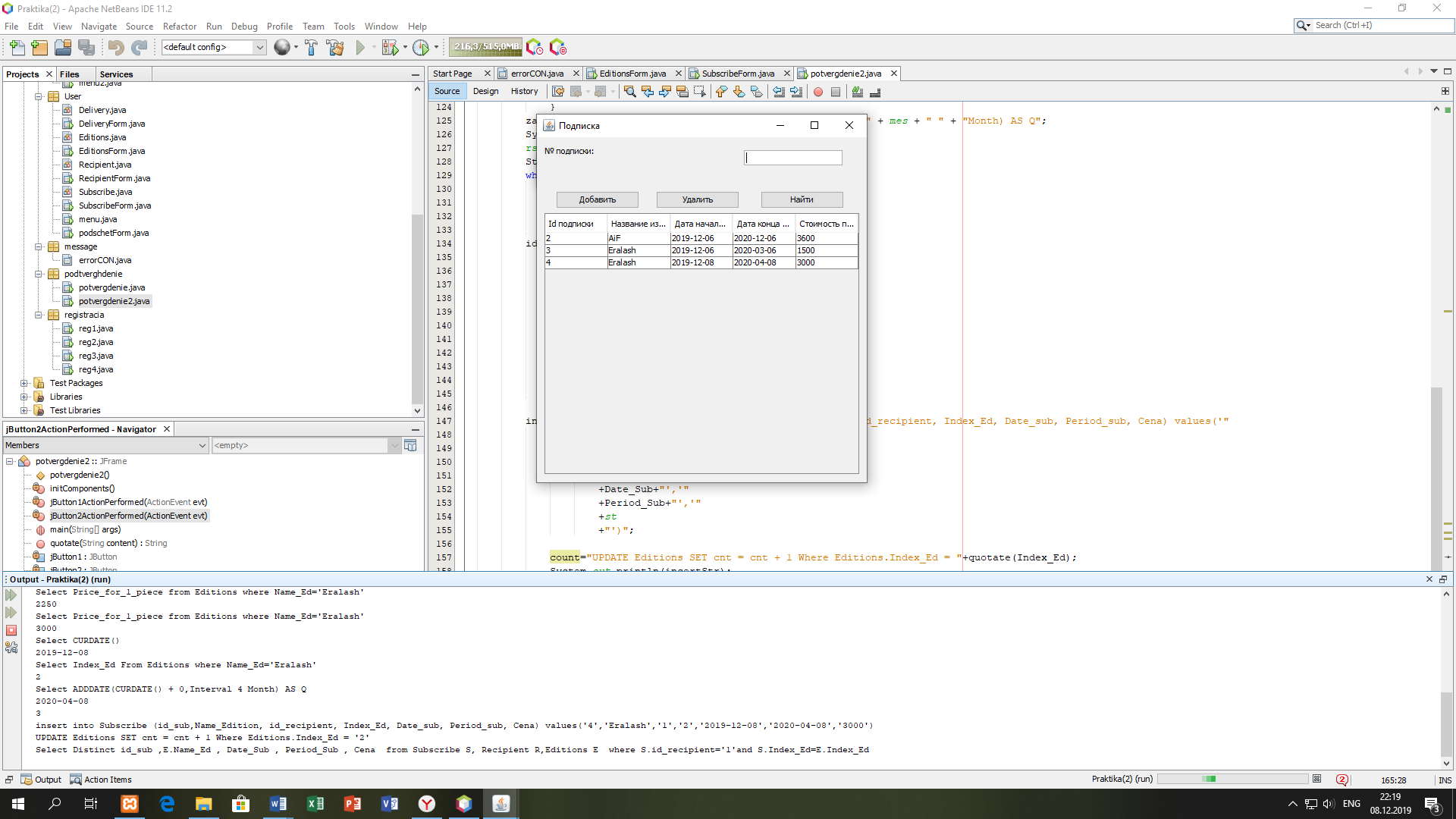
}

Задание 2

Реализовать оформление подписки на издание.







**Код реализации**

**Код из файла PodschetForm.java:**

private void jComboBox2ActionPerformed(java.awt.event.ActionEvent evt) {

String zap = "";

try {

pr = jComboBox2.getItemAt(jComboBox2.getSelectedIndex());

kol = 1;

jSpinner1.setValue(kol);

mes = 1;

jSpinner2.setValue(mes);

zap = "Select Price\_for\_1\_piece from Editions where Name\_Ed=" + "'" + pr + "'";

rs = stmt.executeQuery(zap);

int price = 0;

while (rs.next()) {

price = Integer.parseInt(rs.getString("Price\_for\_1\_piece"));

}

st = price \* kol;

System.out.println(st);

jTextField2.setText(Integer.toString(st));

} catch (SQLException ex) {

Logger.getLogger(podschetForm.class.getName()).log(Level.SEVERE, null, ex);

}

}

**Код из файла Potvergdenie2.java:**

import static Admin.Authorization.id\_rec;

import static User.podschetForm.mes;

import static User.podschetForm.pr;

import static User.podschetForm.st;

public class potvergdenie2 extends javax.swing.JFrame {

private ResultSet rs1;

private ResultSet rs2;

private ResultSet rs3;

private ResultSet rs4;}

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {

String zap1 = "";

String zap2 = "";

String zap3 = "";

String insertStr="";

String count="";

String id="";

try{

zap1 = "Select CURDATE()";

System.out.println(zap1);

rs1 = stmt.executeQuery(zap1);

String Date\_Sub="";

while (rs1.next()) {

Date\_Sub = (rs1.getString("CURDATE()"));

System.out.println(Date\_Sub);

}

zap2 = "Select Index\_Ed From Editions where Name\_Ed=" + "'" + pr + "'";

System.out.println(zap2);

rs2 = stmt.executeQuery(zap2);

String Index\_Ed="";

while (rs2.next()) {

Index\_Ed = rs2.getString("Index\_Ed");

System.out.println(Index\_Ed);

}

zap3 = "Select ADDDATE(CURDATE() + 0" + ",Interval" + " " + mes + " " + "Month) AS Q";

System.out.println(zap3);

rs3 = stmt.executeQuery(zap3);

String Period\_Sub="";

while (rs3.next()) {

Period\_Sub =(rs3.getString("Q"));

System.out.println(Period\_Sub);

}

id="Select MAX(id\_sub) From subscribe";

rs4= stmt.executeQuery(id);

String id\_rec1="";

while (rs4.next()) {

id\_rec1 = rs4.getString("MAX(id\_sub)");

System.out.println(id\_rec1);

}

if (null== id\_rec1){

id\_rec1="0";}

int id\_rec2=Integer.parseInt(id\_rec1);

id\_rec2=id\_rec2+1;

String id\_sub=Integer.toString(id\_rec2);

insertStr="insert into Subscribe (id\_sub,Name\_Edition, id\_recipient, Index\_Ed, Date\_sub, Period\_sub, Cena) values('"

+id\_sub+"','"

+pr+"','"

+id\_rec+"','"

+Index\_Ed+"','"

+Date\_Sub+"','"

+Period\_Sub+"','"

+st

+"')";

count="UPDATE Editions SET cnt = cnt + 1 Where Editions.Index\_Ed = "+quotate(Index\_Ed);

System.out.println(insertStr);

System.out.println(count);

int done4=stmt.executeUpdate(count);

int done=stmt.executeUpdate(insertStr);

getContentPane().removeAll();

initComponents();

this.dispose();

}

catch(SQLException e){

this.dispose();

new message.errorCON().setVisible(true);

}

}

public String quotate(String content){

return "'"+content+"'";

}

**5. Тестирование информационной системы**

|  |  |  |
| --- | --- | --- |
| Метод тестирования | Ожидаемый результат | Фактический результат |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**6. Руководство пользователя**

**7. Заключение**

Итогом курсового проекта является рабочая информационная система «Печатное издание». Данная ИС позволяет более эффективно обслуживать клиентов и вести учёт подписок на издания.

Проект выполнен в полном соответствии с заданием. В проекте разработана эксплуатационная документация, для подробного ознакомления пользователей с работой в системе.

Таким образом, можно сделать вывод, что во время разработки были достигнуты все цели создания системы и, что её внедрение должно привести к повышению производительности и качества работы информационной системы «Печатное издание»

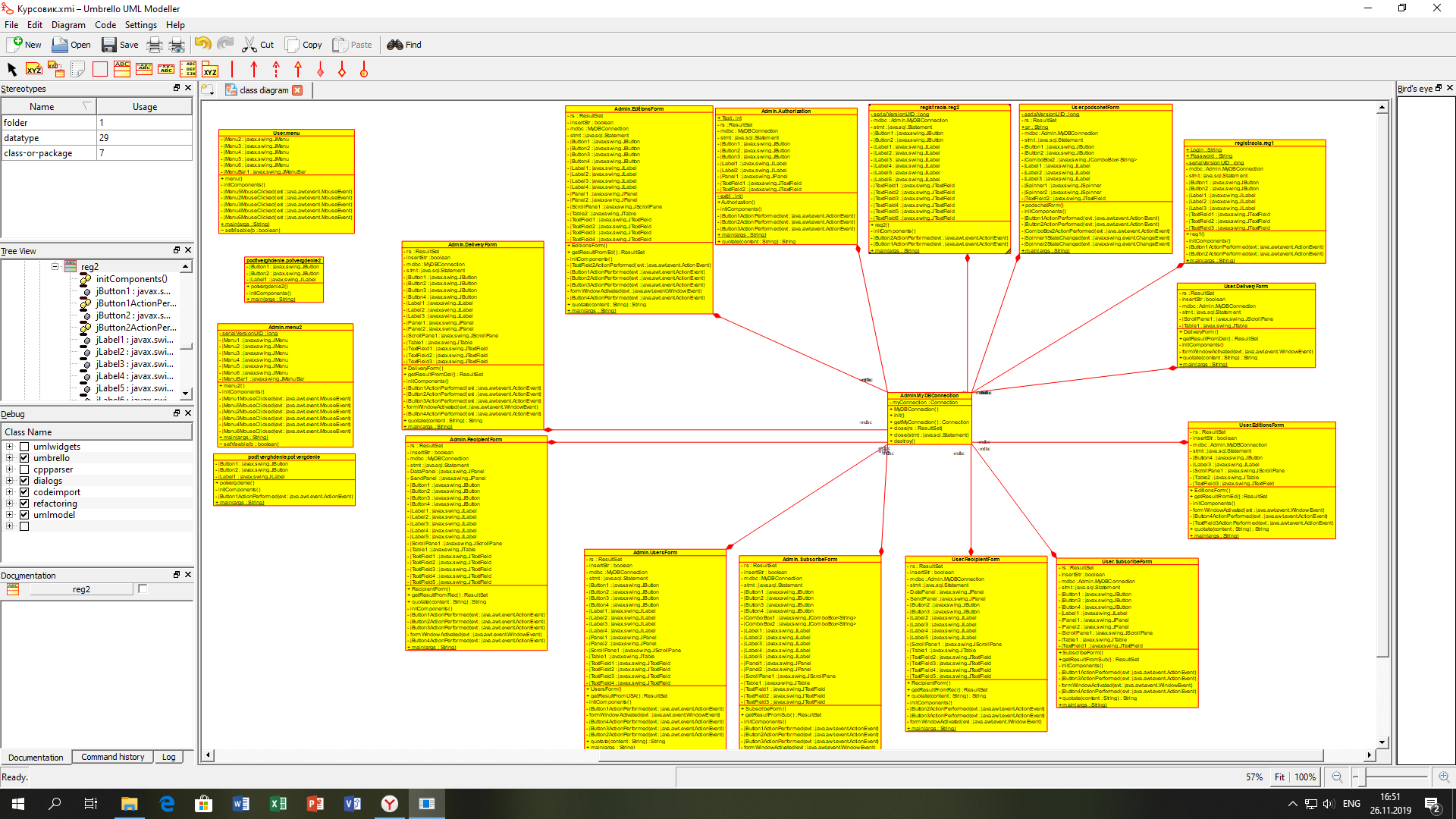
Приложение

Рис.3. Диаграмма классов