|  |  |
| --- | --- |
| МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ  РОССИЙСКОЙ ФЕДЕРАЦИИ | |
| Федеральное государственное бюджетное образовательное учреждение высшего образования  «Пермский государственный национальный  исследовательский университет» | |
|  | Кафедра математического обеспечения  вычислительных систем |
|  |  |
| **РАЗРАБОТКА СИСТЕМЫ КОЛЛЕГИАЛЬНОГО ОЦЕНИВАНИЯ И ЕЕ ИНТЕГРАЦИЯ С MOODLE**  *Курсовая работа* | |
|  | Работу выполнил студент  ПМИ-1,2 группы 3 курса механико-математического факультета  Копылов Н. Р. |
|  | Научный руководитель:  Кандидат физико-математических наук, доцент  Дацун Н. Н.  “\_\_\_\_”\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_20\_\_г. |
| Пермь 2017 | |

# Глоссарий

МООС-платформа (МООС-провайдер) – сайт (платформа), на которой размещаются массовые открытые онлайн-курсы.

Обучающийся – персона, проходящая обучение на курсе.

Грейдер – обучающийся, который оценивает работы, выполненные другими обучающимися.

Рубрики – это руководство, в котором перечислены конкретные критерии оценки данного задания и уровни соответствия данному критерию с указанием начисляемых баллов за соответствие этому критерию.

Инструктор – организатор курса, который ведет лекции на курсе и проверяет задания, выполненные обучающимися.

Относительная шкала оценивания – это шкала оценивания, на основании которой оценка за работу обучающегося вычисляется относительно лучшей работы в группе.

# Введение

Данная работа посвящена разработке и проектированию информационной системы, которая позволит внедрить коллегиальное оценивание (оценивание сверстниками) в систему электронного обучения Moodle.

В последнее время Массовые Открытые Онлайн-Курсы (МООС) получили очень широкое распространение по всему миру. Они предлагают неограниченный доступ к курсам, которые раньше читали только в высших учебных заведениях. МООС-платформы имеют огромный потенциал для распространения бесплатного образования. Большое разнообразие курсов позволяет любому желающему получить знания по интересующему его предмету. Широкое распространение МООС-платформы получили совсем недавно (в 2011 году)[], что объясняется повсеместным распространением интернета, увеличением скорости интернета и развитием веб-технологий. Большим преимуществом МООС-платформ является то, что они позволяют получить доступ к курсам университетского уровня, которые ведут лучшие преподаватели мира.

Существенным недостатком МООС-платформ является то, что количество студентов, которые полностью заканчивают курс достаточно мало (до 30% от количества всех обучающихся)[]. Это объясняется отсутствием мотивации у студентов и тем, что на онлайн-курсах достаточно сложно получить обратную связь от инструктора, так как в онлайн-курсах, как правило, участвуют несколько тысяч человек. Так же из-за большого количества участников у инструкторов нет возможности ввести задания, в которых в качестве ответа требуется код программы или текст на несколько десятков слов, потому что один инструктор не сможет проверить несколько тысяч работ, отправленных обучающимися.

Решением данной проблемы является коллегиальное оценивание (оценивание сверстников). Этот подход к оценке работ, отправленных студентами, не только обеспечивает обучающимся своевременную обратную связь, но и позволяет студентам лучше освоить изученный ими материал.

К сожалению, коллегиальное оценивание является недостаточно изученной темой. Научных работ, посвященных исследованию коллегиального оценивания, крайне мало, а русскоязычные статьи на эту тему практически отсутствуют. Так же многие люди, изучающие данную тему, сходятся во мнении, что у этого подхода имеется ряд недостатков, в том числе недостаточная объективность обучающихся при оценивании друг друга.

Несмотря на недостатки коллегиальное оценивание применяется на практике во многих МООС-платформах (Coursera, EdX и т.д.). Но большинство систем коллегиального оценивания используют неэффективные методы вычисления итоговых оценок обучающихся, как правило, итоговая оценка студента получается, как среднее арифметическое от всех оценок, выставленных грейдерами.

Актуальность работы подтверждается тем, что в настоящее время МООС-платформы получили очень большое распространение. Даже несмотря на то, что во многих МООС-платформах существует возможность добавления задания с коллегиальным оцениванием, для подсчета итоговых баллов студента до сих пор используются неэффективные алгоритмы (например, среднее арифметическое). Также МООС-платформы можно создавать самому при помощи бесплатной системы электронного обучения Moodle, но, к сожалению, в этой системе нет встроенных возможностей для создания заданий с использованием коллегиального оценивания.

Таким образом, целью данной работы является разработка информационной системы, позволяющей внедрить коллегиальное оценивание в систему электронного обучения Moodle, с использованием эффективных алгоритмов вычисления итоговых консолидированных оценок обучающихся на основе выставленных грейдерами оценок.

Для достижения поставленной цели были поставлены следующие задачи:

1. Проанализировать существующие МООС-платформы с целью выявления наличия сервиса коллегиального оценивания;
2. Проанализировать LMS Moodle с целью исследования имеющихся инструментов, позволяющих добавлять задания с коллегиальным оцениванием;
3. Рассмотреть существующие подходы к коллегиальному оцениванию;
4. Рассмотреть существующие алгоритмы вычисления итоговых консолидированных оценок обучающихся, использующие различные подходы к коллегиальному оцениванию;
5. Выбрать наиболее подходящий алгоритм вычисления итоговых консолидированных оценок;
6. Выработать и обосновать требования к системе коллегиального оценивания;
7. Разработать проект информационной системы, позволяющей внедрить количественное коллегиальное оценивание в LMS Moodle;
8. Протестировать выбранный алгоритм вычисления итоговых консолидированных оценок обучающихся на синтетических данных;
9. Проанализировать полученные результаты.

Логика исследования обусловила структуру работы, состоящей из введения, четырех глав, заключения, списка использованной литературы и приложений. Первая глава посвящена анализу существующих МООС-платформ и инструментов коллегиального оценивания, которые на них присутствуют, анализу LMS Moodle и инструментов, которые позволяют добавлять в задания коллегиальное оценивание, анализу существующих подходов к коллегиальному оцениванию, алгоритмов, которые совершенствуют эти подходы, а также нефункциональных требований к разрабатываемой информационной системы, которые содержат рекомендации по тому, как наиболее эффективно реализовать систему коллегиального оценивания. Вторая глава содержит постановку задачи, где рассматриваются исходные данные, которые подаются на вход системе, ограничения на исходные данные, результирующие данные, которые получаются на выходе из системы, а также связь между исходными и результирующими данными. Третья глава посвящена анализу функциональных требований информационной системы, этапу анализа, этапу проектирования и построению модели реализации будущей системы коллегиального оценивания. Четвертая глава содержит тестирование выбранного алгоритма вычисления итоговых консолидированных оценок обучающихся и анализ полученных результатов. В заключении подводятся итоги проведенного исследования, результаты и идеи, позволяющие оценить возможные перспективы их развития и использования.

# Обзор

## 1.1 МООС-платформы

Массовый открытый онлайн-курс (англ. Massive open online courses, MOOC) — это обучающий курс с массовым интерактивным участием c применением технологий электронного обучения и открытым доступом через сеть Интернет. Данная форма дистанционного обучения появилась сравнительно недавно, в 2012 году. Газета The New York Times назвала 2012 год – годом МООС (The Year of MOOC)[сколково]. Благодаря широкому развитию сети Интернет, а также усовершенствование веб-технологий, массовые открытые онлайн-курсы получили такое широкое распространение среди большого количества людей.

МООС-платформы, как правило, рассчитаны на слушателей с различным уровнем подготовки – как на новичков, так и на опытных специалистов. Большое разнообразие курсов позволяет найти что-то интересное человеку с любой сферой интересов и уровнем подготовки.

Еще одна причина, по которой МООС-платформы стали так популярны, это то, что ведущие университеты мира стали выкладывать в открытый доступ курсы, которые читаются студентам этих университетов. Теперь любой желающий может получать знания от лучших преподавателей.

На рисунке 1.1 изображена диаграмма, показывающая то, как увеличивалось количество массовых открытых онлайн-курсов в течении последних шести лет[class central]. По этой диаграмме видно, что количество курсов увеличивается довольно быстро.

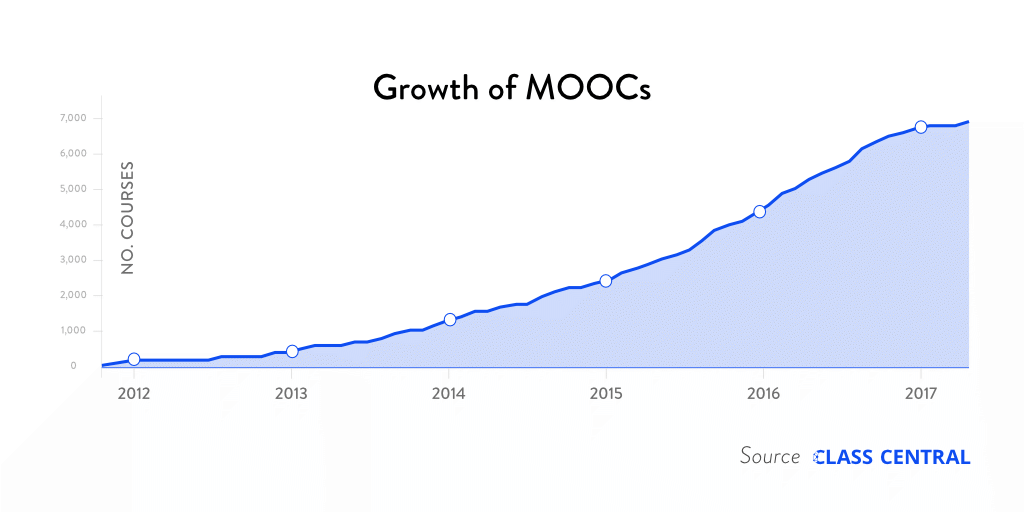


Рисунок 1.1 - Количество массовых открытых онлайн-курсов по годам

В таблице 1.1 показано количество курсов по каждому предмету среди всех МООС-платформ (представлены данные за 2016 год) [ссылка EdSurge]. По этой таблице можно сделать вывод, что самые популярные курсы на МООС-провайдерах – это бизнес-курсы.

Таблица 1.1 - Количество курсов по каждому предмету

| Предмет | Количество курсов по данному предмету (на 2016 год) |
| --- | --- |
| Бизнес | 16,75% |
| Наука | 11,34% |
| Социальные науки | 10,77% |
| Информатика | 9,74% |
| Гуманитарные науки | 9,41% |
| Педагогика | 9,36% |
| Медицина | 8,27% |
| Программирование | 7,44% |
| Искусство и дизайн | 6,73% |
| Инженерные науки | 6,11% |
| Математика | 4,09% |

В таблице 1.2 показано количество курсов у каждого МООС-провайдера[ссылка EdSurge]. Из этой таблицы видно, что больше всего курсов (35.8% от общего количества) содержится в МООС-провайдере Coursera.

Таблица 1.2 - МООС-провайдеры по количеству курсов

| МООС-провайдер | Количество курсов (в % от общего числа, 2016 год) | Наличие коллегиального оценивания |
| --- | --- | --- |
| Coursera | 35.58% | + |
| EdX | 18.06% | + |
| Canvas.net | 6.92% | - |
| FutureLearn | 5.68% | - |
| Miriada X | 3.66% | - |
| France Universite Numerique | 3.33% | - |
| Udacity | 2.95% | + |
| Open Education | 2.12% | - |
| Rwaq | 1.83% | - |
| iversity | 1.78% | - |
| NovoEd | 1.66% | - |
| Другие | 16.44% | - |
| Stepic |  | + |
| OpenSAP |  | + |

В таблице 1.3 можно увидеть 5 самых популярных МООС-провайдеров и количество пользователей, которые на них зарегистрированы.

Таблица 1.3 - Таблица, показывающая количество зарегистрированных пользователей в 5 самых популярных МООС-провайдерах

|  |  |
| --- | --- |
| МООС-провайдер | Количество зарегистрированных пользователей (на 2016 год), млн |
| Coursera | 23 |
| EdX | 10 |
| XuetangX | 6 |
| FutureLearn | 5,3 |
| Udacity | 4 |

В некоторых МООС-платформах используются задания с коллегиальным оцениванием. Вот некоторые из них: Coursera, EdX, Udacity, OpenSAP, Stepic. В целом, достаточно сложно понять использует ли данный МООС-провайдер коллегиальное оценивание, так как на сайте явным образом это нигде не указывается. Обнаружить задания с коллегиальным оцениванием можно только записавшись на курс.

## 1.2 Moodle

Moodle (англ. Modular Object-Oriented Dynamic Learning Environment (модульная объектно-ориентированная динамическая обучающая среда)) – представляет собой свободное (распространяется по лицензии GNU GPL) [сайт мудла], предоставляющее возможность создавать сайты для онлайн-обучения.

Уже зарегистрировано около 80 тысяч сайтов, использующие систему электронного обучения Moodle. На самом деле не представляется возможным абсолютно точно посчитать, сколько сайтов используют данную платформу, поскольку она является свободно распространяемой [moodle usage].

Moodle используется для онлайн-обучения в 234 странах по всему миру. [Moodle statistics] В таблице 1.4 показаны десять стран, в которых Moodle наиболее распространен.

Таблица 1.4 - Десять стран, в которых популярен Moodle

|  |  |
| --- | --- |
| Страна | Количество сайтов |
| Соединенные Штаты Америки | 10,453 |
| Испания | 7,342 |
| Бразилия | 4,502 |
| Мексика | 4,158 |
| Великобритания | 3,493 |
| Италия | 2,890 |
| Германия | 2,570 |
| Австралия | 2,402 |
| Колумбия | 2,336 |
| Индия | 2,284 |

Несмотря на то, что Moodle, как правило, используется для небольшого количества участников, есть сайты, основанные на данной платформе, которые насчитывают более 20000 пользователей [Moodle stat]. На рис. 1.2 изображено соотношение количества сайтов к количеству пользователей, на них зарегистрированных. Из этой диаграммы видно, что зарегистрировано более 1500 сайтов, использующих Moodle, на которых зарегистрировано более 20000 пользователей. Также стоит отметить, что существует 3 сайта, на которых зарегистрировано более 1,000,000 пользователей. Из этого следует, что платформа Moodle вполне может применяться в качестве МООС-платформы.

Примерами МООС-платформ, созданный на базе LMS Moodle являются [ссылка на Дацун]:

1. Box Hill Institute (<http://mooc.bhtafe.edu.au/>);
2. Saylor.org academy (<http://www.saylor.org/>);
3. School of Business and Trade (<http://sobat.org/>);
4. Sophia (<http://www.sophia.ac.jp/eng/e_top>);
5. UP2U (<http://up2u.ipleiria.pt/en/>);
6. Virtual Linguistics Campus (<http://linguistics.online.uni-marburg.de/>).

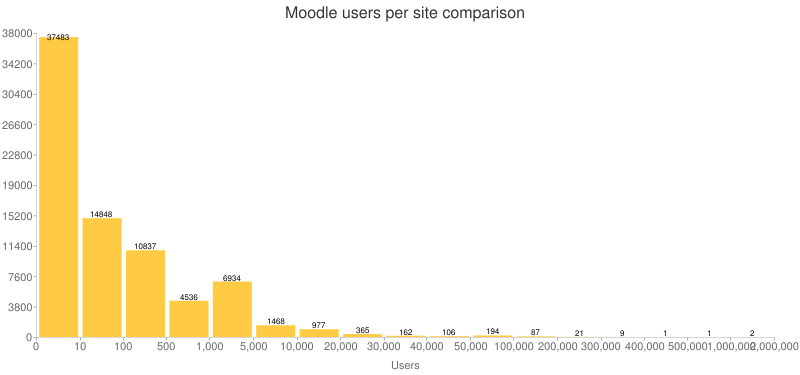


Рисунок 1.2 - Диаграмма показывающая отношение количества сайтов к количеству пользователей, на них зарегистрированных

В платформе Moodle существуют инструменты для создания рубрик для оценивания, но в ней отсутствует полноценное коллегиальное оценивание. Тот инструментарий, который доступен на данный момент, позволяет назначать для каждой работы лишь одного грейдера. Это является существенным недостатком данной платформы.

## Количественное коллегиальное оценивание

Количественное коллегиальное оценивание – это метод оценивания заданий, суть которого заключается в том, что оценка за выполненную работу является численным количественным значением. В настоящее время данный метод является наиболее распространенным в МООС-платформах.

Как правило, количественный подход к оценке работ используется вместе с рубриками. Рубрики очень сильно упрощают процесс оценивания, предоставляя грейдеру всю необходимую информацию для оценивания, тем самым улучшая объективность оценки, выставленной грейдером за работу. При использовании рубрик требования к навыкам оценивания у грейдера сводятся к минимуму. Так же проблемой является то, что многие студенты склонны лояльно оценивать других студентов, поэтому итоговая оценка получается менее объективной. Использование рубрик максимально унифицирует данный процесс. Нам уже не нужно полагаться на индивидуальный подход каждого студента к оцениванию работ или на навыки оценивания каждого конкретного студента, так как в рубриках можно очень подробно описать то, какими качествами должна обладать проверяемая работа для того, чтобы можно было начислить определенное количество баллов. Творческая инициатива студента при выборе количества баллов, которое он выставит за работу, сводится к минимуму.

На данный момент существенной проблемой количественного подхода к коллегиальному оцениванию является то, каким способом должна вычисляться итоговая оценка каждого обучающегося. В МООС-платформах, как Coursera, OpenSAP, в системе коллегиального PEAS, которая является расширением МООС-платформы EdX, итоговая оценка обучающегося вычисляется как среднее арифметическое по всем оценкам, которые выставили ему другие обучающиеся. Данный подход имеет довольно большую погрешность. Еще в таком способе не учитывается, что оценка обучающегося будет очень неточной, если из-за недобросовестных обучающихся, которые всем ставят чересчур завышенные оценки, он получит большую поддержку тех студентов, которые плохо оценивают. Вычисление недобросовестных студентов является первостепенной задачей алгоритмов, которые совершенствуют количественный коллегиальный подход к оцениванию работ.

Так же важной задачей не только количественного коллегиального оценивания, но и коллегиального оценивания в целом, является стимулирование студентов проверять работы качественно и добросовестно. Так как на проверку работ уходит достаточно много времени, и не у всех студентов есть желание тратить на это свои силы. Во всех МООС-платформах, которые каким-либо образом используют коллегиальное оценивание, стимулирование студентов происходит за счет начисления дополнительных баллов за проверку нескольких назначенных работ (как правило, назначают около 3 работ). Опять же при использования данного способа студент никак не контролируется. Качество обратной связи, которое он дал, может быть любым. Мы не можем знать добросовестно ли он подходил к процессу проверки работ или нет. При использовании такого подхода и добросовестным, и не добросовестным студентам начисляется одинаковое количество баллов, вне зависимости от качества проделанной ими работы. Это так же совсем не мотивирует обучающихся подходить к процессу оценивания работ ответственно. Как раз из-за отсутствия мотивации у студентов к оцениванию работ, большинство людей не рассматривает всерьез такой мощный инструмент обратной связи на МООС-платформах, как коллегиальное оценивание.

К таким методам относятся: PeerRank[ссылка], Vancouver Algorythm[ссылка]

Преимущества количественного подхода к коллегиальному оцениванию:

1. Возможность вводить рубрики, поясняющие тот или иной критерий оценивания. Такая возможность крайне необходима для упрощения процесса выставления баллов за задание, потому что большинство обучающихся не имеют нужных навыков выставления оценок. Так же многие студенты склонны ставить более высокие отметки своим сокурсникам. Исходя из вышесказанного, рубрики повышают объективность процесса рецензирования;
2. Инструктор получает больше информации о выполненных работах, так как использование рубрик помогает ему оценить то, какие знания и навыки студенты усвоили хуже, чтобы впоследствии на лекциях уделить этому вопросу большее внимание. В целом, данных подход увеличивает обратную связь между студентами и инструкторами. По большей части это заслуга рубрик, которые идеально подходят для их совместного использования с количественным подходом;
3. Использование абсолютной шкалы оценки заданий. Это означает, что оценки не зависят от качества работ студентов в этой группе. То есть если все студенты выполнили задание не удовлетворительно, то все они получат плохие оценки (баллы высчитываются не относительно лучшей работы в группе, как в порядковом подходе).

Недостатки количественного подхода к коллегиальному оцениванию:

1. Большая предвзятость у оценщиков. Для более-менее объективного оценивания нужен достаточный опыт в рецензировании работ. Каждый оценщик по-разному может оценить одну и ту же работу;
2. Абсолютная шкала оценки сама по себе довольно сложна для освоения. Людям по большей части гораздо проще сравнивать две работы и выявлять, какая из них лучше, а какая хуже (такой подход используется в порядковом методе коллегиального оценивания).

## Порядковое коллегиальное оценивание

Порядковое коллегиальное оценивание – это метод оценивания, основанный на попарном сравнении работ, который осуществляет ранжирование всех выполненных студентами заданий. Данный подход менее распространен и на практике он не применяется, однако, существует несколько статей, в которых рассматривают данный метод в теории и даже приводят формализованную математическую модель этого порядкового подхода.

Главная идея порядкового подхода к коллегиальному оцениванию заключается в том, что студент при проверке работ выставляет за работу не количество баллов, которое ему кажется подходящим, а проводит попарное сравнение нескольких пар назначенных ему работ. То есть в каждой паре он выбирает лучшую и худшую работу. Естественно, при использовании порядкового подхода использовать рубрики не представляется возможным, поэтому порядковое коллегиальное оценивание не получится применить для достаточно объемных заданий, где требуется всесторонняя проверка каждой работы.

Основным недостатком данного подхода является то, что итоговая оценка, полученная при использовании порядкового коллегиального оценивания, является не количеством баллов, а положением (рангом) в рейтинге работ всех обучающихся в группе. То есть оценка выражает не качество работы, а ее положение в рейтинге относительно лучшей работы. В случае, если все студенты в группе выполнили работу плохо, то они не получат плохую оценку.

Еще одной проблемой, которая должна решаться алгоритмами, совершенствующими порядковый подход к коллегиальному оцениванию, является то, что итоговую оценку нужно вычислить абсолютно для всех студентов, то есть необходимо охватить максимально возможное число студентов в группе. Иначе многие обучающиеся рискуют не получить оценку за задание. Это касается не только порядкового подхода, но и коллегиального оценивания в целом.

К таким методам относятся: Bayesian Peer Grading[ссылка], Mallows Model[], Bradley-Terry Model[].

Преимущества порядкового подхода к коллегиальному оцениванию:

1. Не требует от оценщика каких-либо специальных навыков оценивания. Для человека гораздо проще указать, что проект А хуже или лучше проекта Б, чем выставлять баллы в абсолютной шкале;
2. В некоторых математических моделях, которые использует в порядковом коллегиальном оценивании, есть специальные методы, позволяющие выявить недобросовестного оценщика;
3. Порядковый подход к оценке работ снижает нагрузку на оценщика, ему не приходится тратить дополнительное время на выставление баллов по абсолютной шкале. Следовательно, такой подход уменьшает вероятность того, что в какой-то момент ему надоест оценивать назначенные ему работы, и он выставит оценки наугад. Благодаря этому повышается качество обратной связи от оценщиков;
4. Так как люди лучше справляются со сравнением предметов, оценщик может быть в состоянии предложить более проницательные замечания по положительным и отрицательным аспектам решения студента.

Недостатки порядкового подхода к коллегиальному оцениванию:

1. Оценка, полученная при использовании порядкового метода, выражается в перцентилях (т.е. оценка вычисляется относительно самой лучшей работы в группе). Такая оценка несет в себе намного меньше информации о работе студента, так как оценивает не его знания, а его рейтинг в группе;
2. Необходимость переводить оценки в баллы по абсолютной шкале, потому что во всех МООС-платформах используются баллы в качестве вознаграждения за каждое выполненное задание;
3. Порядковая оценка содержит в себе меньше информации, как для инструктора, так и для оценщика. Чаще всего количественные рубрики более информативны и позволяют выяснить то, как обучающийся освоил тот или иной навык;
4. Не подходит для оценивания объемных работ таких как, эссе на более чем 150 слов и т.д.;
5. Обучающимся гораздо проще оценивать, используя рубрики, т.к. они помогают прояснить некоторые детали процесса рецензирования;
6. Каждый оценщик охватывает лишь небольшую часть упорядочиваний работ.

## Сравнение количественного и порядкового подхода к коллегиальному оцениванию

В таблице 1.5 приведено сравнение количественного и порядкового подходов к коллегиальному оцениванию.

Таблица 1.5 - Сравнение количественного и порядкового подходов

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Метод | Простота оценивания для студента | Методы определения недобросовестных оценщиков | Относительная шкала оценивания | Возможность использования рубрик | Подходит для оценивания объемных работ | Предполагает стимулы для грейдеров оценивать корректно |
| Количественное | Требует навыков оценивания | Нет | Нет | Да | Да | Нет |
| Порядковое | Просто и интуитивно | Да | Да | Нет | Нет | Нет |

Исходя из этой таблицы, можно сделать вывод, что количественный метод больше подходит для большинства практических задач из-за того, что он предполагает использование рубрик, подходит для оценивания достаточно объемных работ (например, эссе на 150 слов).

Порядковый метод так же может использоваться для ряда заданий с использованием коллегиального оценивания, но из-за того, что с помощью него нельзя вычислить оценку по абсолютной шкале, и, учитывая, что в МООС-платформах обычно используется абсолютная шкала оценивания, применять этот подход будет не целесообразно.

Несмотря на то, что для коллегиального оценивания больше подходит количественный подход, в нем присутствуют несколько недостатков. Количественное оценивание никак не стимулирует грейдеров оценивать корректно, в нем отсутствуют методы наказания недобросовестных грейдеров, и так же в большинстве алгоритмов, которые высчитывают итоговую консолидированную оценку, полученную с помощью количественного подхода, используется среднее арифметическое всех оценок, которые грейдеры поставили за работу. Недостаток, связанный с вычислением итоговых баллов, как среднего от всех оценок, является самым существенным. Он сильно влияет на погрешность вычисления оценки обучающегося. Так же количественный подход меньше подходит для неподготовленных грейдеров, потому что людям гораздо проще сравнить предметы, чем выставлять оценку в баллах. Нерешенной так же является проблема слишком завышенных оценок, которые ставят студенты.

Далее будут рассмотрены алгоритмы, использующие количественный и порядковый подходы к коллегиальному оцениванию. Эти алгоритмы устраняют некоторые вышеперечисленные недостатки порядкового и количественного подхода к коллегиальному оцениванию.

## PeerRank

### Общее описание алгоритма

PeerRank – алгоритм, который совершенствует количественный подход к коллегиальному оцениванию. Он был предложен Тоби Уолшем, профессором, изучающим искусственный интеллект, из австралийского Университета Нового Южного Уэльса[]. Название PeerRank было выбрано из-за того, что принцип работы алгоритма похож на принцип работы алгоритма ссылочного ранжирования PageRank[].

С помощью этого алгоритма оценка обучающегося вычисляется на основе оценок, которые ему поставили другие обучающиеся. При чем этот метод учитывает те оценки, которые поставил студент другим студентам, поскольку оценка – мера способности обучающегося правильно оценивать работы. Так же этот метод обеспечивает стимулы для студентов ставить оценки корректно, так как его собственная оценка зависит от оценок, которые он выставил другим обучающимся. Метод в том числе увеличивает оценку студентам, которые добросовестно подходят к процессу оценивания и уменьшает оценку тем студентам, которые недобросовестно оценивают назначенные им работы, потому что чересчур завышенные баллы, которые поставил недобросовестный грейдер могут повредить тому обучающемуся, которого оценивали, так как он получает большую поддержку тех грейдеров, которые плохо оценивают.

В большинстве систем коллегиального оценивания при вычислении итоговой консолидированной оценки студента используется среднее арифметическое от всех оценок, которые поставили ему поставили другие студенты. Большое преимущество алгоритма PeerRank в том, что с помощью него обучающиеся, которые добросовестно подходят к процессу проверки работ получают оценку выше, чем просто среднее арифметическое по всем выставленным им оценкам. Это так называемый «кредит доверия». Данный подход стимулирует обучающихся проверять работы честно. В случае, если грейдер ставит всем чересчур высокие или чересчур низкие оценки, алгоритм занижает оценку этого студента. Тоби Уолш в своей статье приводит результаты тестирования PeerRank на синтетических данных. Из них четко видно, что квадратичная погрешность алгоритма PeerRank примерно в 2 и более раз ниже, чем у метода вычисления оценок с помощью среднего арифметического.

Данный алгоритм показывает наилучшие результаты, если каждого обучающегося оценят не менее 10 других обучающихся. Скорее всего, если применять данный алгоритм на курсе, на который подписано несколько тысяч человек, то это ограничение не будет критичным. Так же для наилучшей работоспособности метода необходимо, чтобы большинство студентов добросовестно подходили к процессу проверки работ. Этот недостаток как раз предлагают решать за счет начисления «кредита доверия».

### Основные термины

*–* оценка i-го обучающегося на n-ой итерации

A – матрица оценок

– оценка, которую поставил обучающийся под номером j обучающемуся под номером i

m–количество обучающихся

α,β; 0< α+β ≤1 – вспомогательные параметры нужные в формуле.

### Алгоритм

*На вход подаются матрица А и количество обучающихся m, альфа, бета и эпсилон*

***for*** *i:=0* ***to*** *количество\_обучающихся-1*

***begin***

*сумма:=0*

***for*** *j:=0* ***to*** *m-1*

*сумма:=сумма+матрица\_оценок[i,j]*

*итоговые\_оценки[i]:=1/количество\_обучающихся \* сумма*

***end***

***repeat***

*предыдущие\_итоговые\_оценки:=итоговые\_оценки*

***for*** *i:=0* ***to*** *количество\_обучающихся-1*

***begin***

*первая\_часть\_выражения:=(1-альфа-бета)\*предущие\_итоговые\_оценки[i]*

*сумма:=0*

*сумма\_итоговых\_оценок:=0*

***for*** *j:=0* ***to*** *количество\_обучающихся-1*

***begin*** *сумма:=сумма+предудущие\_итоговые\_оценки[j]\*матрица\_оценок[i,j] суммма\_итоговых\_оценок:=сумма\_итоговых\_оценок+предыдущие\_итоговые\_оценки[j]*

***end***

*вторая\_часть\_выражения:=альфа/сумма\_итоговых\_оценок\*сумма*

*сумма:=0*

***for*** *j:=0* ***to*** *количество\_обучающихся-1*

***begin***

*сумма:=сумма+1-|матрица\_оценок[j,i]-предыдущие\_итоговые\_оценки[j]|*

***end***

*третья\_часть\_выражения:=бета/количество\_обучающихся\*сумма*

*итоговые\_оценки[i]:=первая\_часть\_выражения+вторая\_часть\_выражения+третья\_часть\_выражения*

***end***

*максимум:=0*

***for*** *i:=0* ***to*** *количество\_обучающихся-1*

***begin***

***if*** *|итоговые\_оценки[i]-предыдущие\_итоговые\_оценки[i]|>максимум* ***then***

*максимум:=|итоговые\_оценки[i]-предыдущие\_итоговые\_оценки[i]|*

***end***

***until*** *максимум < эпсилон*

### 1.7.4 Преимущества и недостатки

Преимущества PeerRank:

1. Более точное вычисление оценки, по сравнению с обычными методами вычисления консолидированной оценки, где просто вычисляется среднее между баллами, которые выставили оценщики за работу;
2. Обеспечение стимулов для студентов, для того чтобы они стремились оценивать лучше и подходили к этому процессу более ответственно. Если студент выставляет слишком заниженные баллы за работы других студентов, то его собственная оценка так же страдает от этого;
3. Использование абсолютной шкалы оценивания. Это означает, что оценка не зависит от лучшей работы в группе;
4. Данный алгоритм очень просто реализовать;
5. Метод PeerRank позволяет определить недобросовестных обучающихся, которые ставят чересчур заниженные оценки или чересчур завышенные [].

Недостатки PeerRank:

1. Относительно большое количество студентов, которые должны оценить каждую работу других студентов. Для нормальной работы алгоритма требуется не менее 5 оценок для каждой работы. Рекомендуемое количество рецензий на каждого студента равняется 10[]. В целом же метод не зависит от количества оценок, поставленных каждому студенту;
2. Необходимо, чтобы большинство студентов добросовестно подходило к процессу проверки работ других обучающихся. Впрочем, этот недостаток относится и ко всему коллегиальному оцениванию в целом.

## Bayesian Peer Grading

Bayesian Peer Grading (Байесовский метод коллегиального оценивания) – алгоритм, который совершенствует порядковое коллегиальное оценивание. Он был предложен Картником Раманом, профессором из Корнеллского Университета, который изучает машинное обучение, онлайн-обучение, Data Mining и тд.[] Данный алгоритм уже имеет готовую программную реализацию. Протестировать и посмотреть исходный код данной программы можно на веб-сайте <peergrading.com>.

Данный алгоритм использует попарные сравнения работ для упорядочивания работ по рангу. Обучающимся даются несколько пар работ других обучающихся. Они должны в каждой паре выбрать какая работа лучше. На основе этих сравнений работы упорядочиваются, а оценка вычисляется в перцентилях. То есть оценка отражает лишь положение работы в группе относительно других работ, но не отражает реальную оценку этой работы. Для каждой работы вычисляется вероятность того, какую позицию в рейтинге она будет занимать, средний ранг, медиану и энтропию.

На рисунке 1.3 изображен пример из статьи[], на котором показаны распределения вероятностей принадлежности к тому или иному рангу для 4 работ. Как видно из графиков, первая работа является лучшей из всех четырех, и из-за небольшой энтропии существует большая уверенность в том, что первая работа лучше всех остальных. У третьей работы очень большая энтропия, поэтому мы не можем с уверенностью сказать, что ее ранг равен самому высокому столбцу на графике. Для этого инструктору курса необходимо провести дополнительную проверку работы, чтобы выяснить позицию третьей работы в рейтинге более точно.

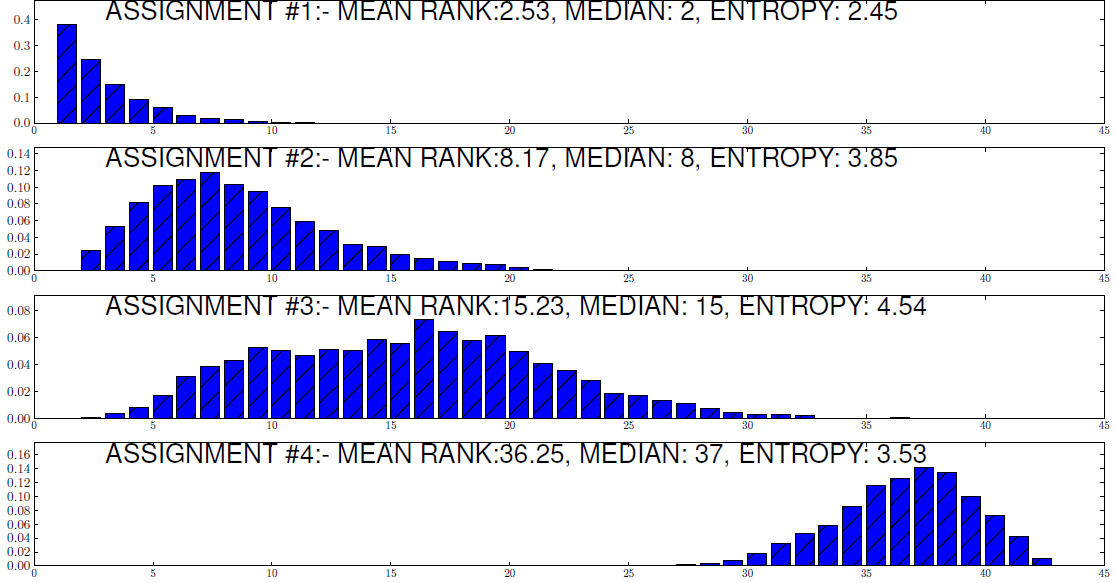


Рисунок 1.3 - Пример распределения рангов для 4 работ

Данный алгоритм совершенствует уже существующие системы, которые вычисляют ранг на основе комбинации порядковой (ранговой) информации из нескольких различных источников. Аналогичный подход применяется в системах голосования («Voting Systems for Social Choice»)[], а так же применяется в системах агрегирования результатов поиска таких, как Rank Fusion и Metasearch. Такие системы используют расширения таких классических методов агрегации ранга, как модель Брэдли-Терри и модель Маллоус.

Данный алгоритм расширяет классическую модель Маллоус. В отличие от приведенных выше систем, в порядковом коллегиальном оценивании существует своя специфика агрегации рангов, имеющая фундаментальные отличия, которые, в свою очередь, не позволяют использовать уже существующие методы агрегации ранга.

Первое отличие заключается в том, что задача существующих систем агрегации поисковых результатов или систем голосования заключается в том, чтобы правильно определить ранги самых верхних элементов в рейтинге. В порядковом коллегиальном оценивании необходимо определять ранги для всей совокупности элементов (работ), а не только для лучших работ, так как итоговую оценку должны получить все обучающиеся, а не только самые успешные.

Второе отличие заключается в том, что итоговая оценка у многих обучающихся может получиться с очень высокой энтропией. Такая ситуация получается из-за того, что не все грейдеры проверяют и оценивают назначенные им работы, поэтому, наверняка, оценить работы всех обучающихся не получится или попарных сравнений с участием каких-то работ будет настолько мало, что точный ранг вычислить не представляется возможным, и из этого следует, что у вычисленного ранга данной работы будет очень высокая энтропия. Информация об энтропии так же должна быть видна инструкторам, для того чтобы они могли провести дополнительную проверку работы с целью выяснить реальное положение работы в рейтинге.

Основная сложность данного алгоритма заключается в том, чтобы по вычислить ранг работы относительно всего множества работ на основе попарных сравнений её с небольшим подмножеством работ.

Преимущества Bayesian Peer Grading:

1. Порядковый подход к коллегиальному оцениванию более прост для понимания. Люди намного лучше справляются с попарным оцениванием работ, чем с выставлением количественной оценкой за работу, потому что для этого нужны необходимые навыки оценивания;
2. Данный алгоритм уже имеет программную реализацию и исходный код программы имеется в открытом доступе. То есть данный алгоритм существует не только в качестве теоретической математической модели;
3. Один из наиболее подробно разобранных алгоритмов, которые совершенствует порядковый подход к коллегиальному оцениванию.

Недостатки Bayesian Peer Grading:

1. Сложность данного алгоритма. Математическая модель, изложенная в статье Картника Рамана[], очень сложна для понимания человеку, не имеющего достаточной математической подготовки и знания математической статистики на очень высоком уровне. Так же автор статьи не очень много внимания уделяет тому, чтобы объяснять, назначения той или иной математической формулы, использованной в статье. В силу этой причины еще больше усиливается непонимание того, как работает данный алгоритм;
2. Для того чтобы данный алгоритм вычислял ранг достаточно точно, необходимо, чтобы для каждой работы имелось достаточно большое количество попарных сравнений данной работы с работами других обучающихся;
3. Оценка вычисляется относительно других работ в группе. Во всех МООС-платформах оценка имеет количественное выражение в баллах. В конце концов положение работы в рейтинге (ее ранг) при экспорте оценок в МООС-платформу потребуется преобразовать в итоговую оценку в баллах.

## Сравнение алгоритмов PeerRank и Bayesian Peer Grading

В таблице 1.6 приведено сравнение алгоритма PeerRank, использующего количественный подход к коллегиальному оцениванию, и алгоритма Bayesian Peer Grading, использующего порядковый подход к коллегиальному оцениванию.

Таблица 1.6 - Сравнение PeerRank и Bayesian Peer Grading

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Алгоритм | Относительная шкала оценивания | Возможность использования рубрик | Подходит для оценивания объемных работ | Предполагает стимулы для грейдеров оценивать корректно |
| PeerRank | Нет | Да | Да | Да |
| Bayesian PG | Да | Нет | Нет | Нет |

## Эффективные методы организации процесса коллегиального оценивания

В данном разделе приведены рекомендации о том, как наиболее рационально и эффективно спроектировать, а также реализовать систему коллегиального оценивания.

Требования к системе коллегиального оценивания:

1. Требование ко времени. Оптимальное обратная связь должна быть обеспечена в начале процесса оценки, с тем чтобы дать обучающимся возможность среагировать и улучшить свою работу[];
2. Требование к анонимности. Важным аспектом коллегиальной оценки является обеспечение анонимности обратной связи. Таким образом, рецензенты могут обеспечить критическую обратную связь и оценивание без учета факторов, например, дружба или личные антипатии. Так же система коллегиального оценивания должна поддерживать процесс двойного слепого оценивания, когда не только грейдер не знает личности автора работы, но и автор работы не знает, кто проверил его работу;
3. Требование к оценке. Коллегиальная оценка должна быть только частью итоговой оценки в целях обеспечения достоверности результатов оценки;
4. Требование к количеству грейдеров. Результаты оценки могут быть более точными и достоверными, когда обучающиеся получают обратную связь от нескольких рецензентов, а не от одного. Таким образом, студенты имеют возможность получить разностороннюю обратную связь;
5. Требования к инструктору. Инструктор все еще должен принимать активное участие в процессе коллегиальной оценки, создавая рубрики для задания и придумывая критерии оценки задания, обеспечивая примеры решения и проверки результатов коллегиального оценивания. Он также может помочь в развитии навыков рецензирования у обучающихся;
6. Требование к циклическому процессу оценивания. Наличие нескольких итераций обратной связи позволяет достичь лучшего результата обучения, поскольку обучающиеся могут многократно размышлять над назначенной им работой;
7. Требования к пользовательскому интерфейсу[]. Интерфейс должен быть простым, понятным, и достаточно простым в использовании, а также должен требовать минимальное количество данных, которые вводит пользователь. Всё должно быть достаточно наглядно и интерактивно. При разработке системы коллегиального оценивания необходимо уделить очень большое внимание графическому интерфейсу пользователя;
8. Требования к наличию рубрик. Необходимо предоставить обучающимся рубрики для каждого задания, которые включают в себя описания каждого критерия оценивания. Это необходимо для достижения справедливой и последовательной обратной связи для всех участников курса;
9. Требования к интеграции с МООС-платформой. Коллегиальным оцениванием должно быть легко управлять. Модуль должен быть интегрирован в платформу с функциями для активации и деактивации.
10. Требования к масштабируемости. Фундаментальное различие между MOOC-платформами и традиционным классом есть масштаб учащихся. Поэтому необходимо учитывать нагрузку учащихся на систему коллегиального оценивания при ее реализации.
11. Требования к срокам. У обучающихся должны быть обозначены два вида сроков. Срок сдачи работы – дата, до которой нужно предоставить работу для оценивания, и срок проверки назначенных работ – дата, до которой нужно проверить все назначенные работы других обучающихся.

## Выводы

Несмотря на некоторые недостатки алгоритм PeerRank существенно совершенствует процесс вычисления консолидированных оценок обучающихся при использовании количественного подхода к коллегиальному оцениванию. Погрешность при вычислении оценок при помощи данного алгоритма в 2 и более раз меньше, чем, когда консолидированная оценка вычисляются просто как среднее арифметическое от всех оценок, выставленных обучающемуся.

Существенным недостатком алгоритма PeerRank является то, что необходимо, чтобы работу каждого обучающегося оценили не менее 5 грейдеров. Это означает, что всем студентам помимо выполнения своих заданий, нужно будет проверить еще не менее 5 работ других студентов. Это является существенной нагрузкой для студентов в том случае, если ответ на данное задание является объемным и требует достаточно большого количества времени для проверки. Большая нагрузка на обучающихся может вызвать у них нежелание проверять назначенные им работы, и тогда многие студенты останутся не оцененными. Описанных выше проблем можно избежать, если, например, делить одно большое задание на несколько подзаданий для распределения нагрузки на обучающихся.

Несмотря на то, что алгоритм Bayesian Peer Grading является достаточно мощным улучшением порядкового подхода к коллегиальному оцениванию, у него все еще остается ряд недостатков, которые являются недостатками порядкового коллегиального оценивания в целом.

Главным недостатком Bayesian Peer Grading является то, что данный алгоритм вычисляет итоговую оценку обучающегося относительно лучшей работы в группе. Этот недостаток ставит под сомнение объективность вычисления оценки при помощи данного алгоритма. Также в МООС-платформах, как правило, используется шкала оценивания, выраженная в баллах.

Еще достаточно существенным недостатком является то, что при использовании Bayesian Peer Grading нельзя использовать рубрики, которые сильно упрощают процесс оценивания.

Так же Bayesian Peer Grading не получится применить к заданиям, в которых требуется достаточно объемный ответ (например, эссе на 150-200 слов). Так как проводить попарное сравнение работ будет очень сложно.

Учитывая все вышесказанное, можно с уверенностью сказать, что алгоритм PeerRank является достаточно мощным улучшением количественного подхода к коллегиальному оцениванию. Так же при сравнении алгоритмов Bayesian Peer Grading и PeerRank можно сделать вывод о том, что PeerRank является наиболее подходящим алгоритмом вычисления итоговых консолидированных оценок обучающихся, на основе тех оценок, которые им поставили грейдеры. Количественный подход в целом является наиболее подходящим для коллегиального оценивания на МООС-платформах, несмотря на необходимость навыков оценивания у студентов. Поэтому для системы коллегиального оценивания, наиболее подходящим алгоритмом для вычисления итоговой консолидированной оценки обучающегося будет алгоритм PeerRank.

# 2 Постановка задачи

В данном разделе описываются исходные данные с указанием типов, которые подаются на вход системе, ограничения на исходные данные, результирующие данные, которые получаются на выходе из системы, а также связь исходных данных и результатов.

## 2.1 Исходные данные

На вход системе коллегиального оценивания приходит:

1. От инструктора:
   1. Assign: структура {Id: int; Name: string; CourseId: int; Descript: string; MaxScore: int; CreatedDate: datetime; EndDate: datetime}. //Информация о задании
   2. MGrad: string //Метод оценивания (перечисление)
   3. GradDate: datetime //Сроки оценивания задания (дата)
   4. Rubr: структура {Descript: string; ListLevel: List<Level>} //Рубрики содержат инструкции по тому, как оценивать работу
   5. k: int //количество обучающихся, которые должны выполнить задание Assign
2. От обучающегося:
   1. Feedback: file|string //Ответ на задание в виде файла или текста,
   2. StudentId: int //Информация об обучающемся (его идентификатор),
   3. AssighId: int //Идентификатор задания. Все эти данные система может получить из БД платформы Moodle.
3. От грейдера:
   1. GraderId: int //Информация о грейдере (его идентификатор)
   2. Rate: int //Оценка, выраженная (баллы)
   3. Rem: string // Комментарий к выполненной обучающимся работе (строка)
   4. M: int //количество проверенных им работ

## 2.2 Ограничения на исходные данные

Ограничениями, которые вводятся на исходные данные, принимаемые системой коллегиального оценивания, являются:

1. Ограничения на срок оценивания задания. Дата, до наступления которой грейдерам необходимо оценить назначенные им работы, должна быть не менее, чем на неделю позже срока окончания выполнения работы.
2. Ограничения на рубрики. Рубрик должно быть не менее двух. Рубрика должна содержать текстовое описание показателя, на соответствие которому грейдер проверяет работу обучающегося. В рубрике должно быть не менее двух уровней. Уровень – это степень соответствия работы конкретному показателю с описанием и указанием баллов, которые ставятся за этот показатель, если работа соответствует этому уровню.
3. GradDate > Assign.EndDate
4. Все значения типа int должны быть больше либо равны нулю.
5. Feedback != null

## 2.3 Результаты

Результатами работы системы коллегиального оценивания будут:

1. SumGrad: {табл: [1..k]: {Grade: int; id:int}} //Консолидированные оценки всех обучающихся с указанием информации об обучающемся (его идентификатор) и идентификатора задания.
2. n: int //количество рецензий, которые были написаны грейдерами на задание Assign
3. R: {таблица с 4 полями [1..n]: {GraderId: int; Rate:int; Rem: string; M:int}} //Список оценок и комментариев к ним, которые отправили грейдеры. Помимо этого, должна быть указана информация о каждом грейдере (его идентификатор).

## 2.4 Связь исходных данных и результатов

# 3 Проектирование информационной системы

## 3.1 Функциональные требования к системе

Функциональными требованиями системы коллегиального оценивания являются:

1. Настройка параметров оценивания задания (количественное, порядковое оценивание или рубрики).
2. Загрузка работ обучающихся, которые они отправили в качестве ответа на задание с использованием коллегиального оценивания. В качестве ответа на задание принимается файл или текстовая информация.
3. Назначение отправленных обучающимися работ грейдерам для дальнейшего оценивания.
4. Оценивание грейдером назначенной ему работы.
5. Вычисление итоговых консолидированных оценок за выполненное задание для каждого обучающегося. Вычисление оценки происходит с учетом качества проделанной работы по оцениванию других обучающихся.
6. Экспорт итоговых оценок в МООС-платформу.

## 3.2 Этап анализа

Этап анализа предполагает подробное исследование бизнес-процессов и информации, необходимой для их выполнения (сущностей, их атрибутов и связей). На этом этапе создается информационная модель, а на следующем за ним этапе проектирования — модель данных.

### 3.2.1 Описание прецедентов

Прецеденты описывают сценарии взаимодействия внешних объектов по отношению к системе (акторов) и самой системы[книга чуприной]. Описывать прецеденты необходимо, для того чтобы понять каким образом система будет взаимодействовать со своим окружением (в том числе и с конечными пользователями), какая функциональность требуется от информационной системы и какие будут результаты взаимодействия акторов и системы.

Прецеденты, описывающие взаимодействие акторов с системой коллегиального оценивания:

Главные:

1. Обучающийся отправил ответ на задание.
2. Распределение работ обучающихся.
3. Оценивание грейдером работы.
4. Вычисление итоговых консолидированных оценок всех обучающихся.
5. Экспорт консолидированных оценок за конкретное задание в МООС-платформу.

Второстепенные:

1. Настройка параметров оценивания задания.

На рисунке 3.1 изображено графическое представление взаимодействия системы и акторов в виде UML-диаграммы.

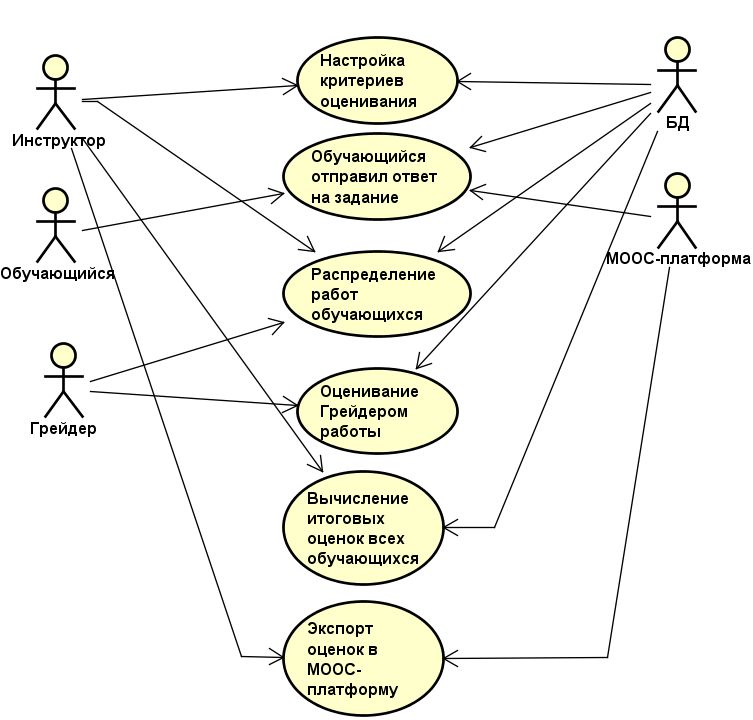


Рисунок 3.0.1 - Диаграмма прецедентов

Одним из основных прецедентов является «Оценивание грейдером работы».

Описание прецедента:

*Название*: Оценивание грейдером работы

*Акторы*: Грейдер, БД, МООС-платформа, обучающийся

*Описание*: Грейдеру назначили работу другого обучающегося для оценивания.

Грейдер изучает критерии оценки данного задания. Грейдер прочитывает работу обучающегося. Грейдер при помощи критериев оценки выставляет справедливую на его взгляд оценку данной работе.

*Триггер*: Обучающийся решил задание и отправил на проверку.

*Основной поток:*

Таблица 3.1. Таблица, отображающая основной поток прецедента

|  |  |
| --- | --- |
| Действия акторов | Отклик системы |
| 1. Грейдер изучает критерии оценивания. Грейдер прочитывает работу обучающегося. Грейдер выставляет оценку и подтверждает оценку. | 1. Система загружает оценку грейдера и комментарий к работе в БД. Если грейдер проверил все работы, которые прислали ему для оценивания, то он получает свое вознаграждение в виде баллов или дополнительных дней на задание, иначе Е1. |

*Альтернативные потоки:*

E1: Грейдер не оценивает данную работу по каким-либо причинам. Прецедент прекращается.

### 3.2.2 Описание бизнес-процесса

В отличии от прецедента, который описывает решение некоторой частной задачи пользователя, бизнес-процесс ориентирован на достижение глобальных целей компании, включает в себя несколько прецедентов и позволяет понять контекст возникновения прецедентов.

Для описания бизнес-процессов используются построение диаграмм активностей. На рисунке 3.2 изображена диаграмма активностей, описанная на языке UML.

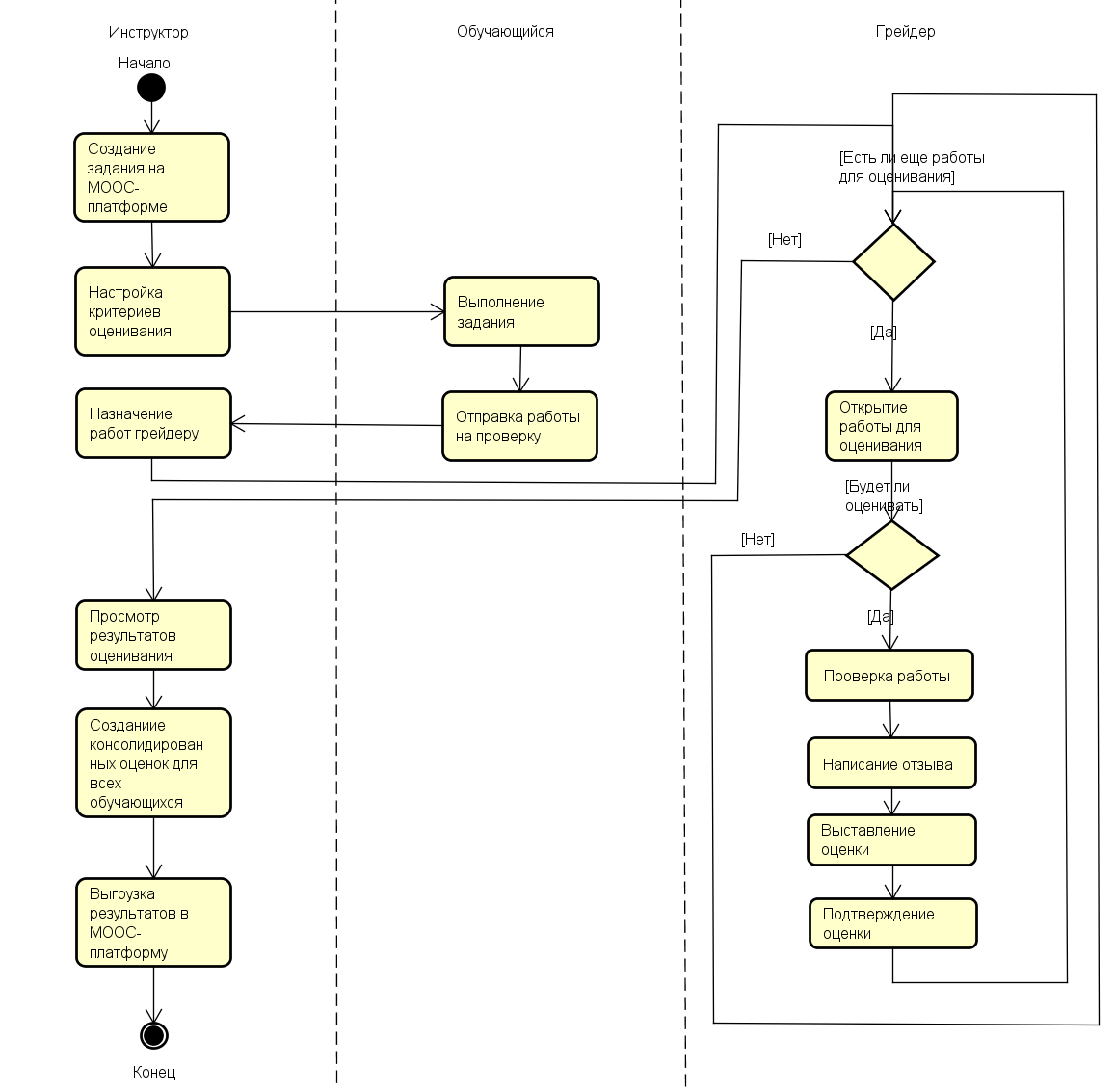


Рисунок 3.0.2 - Диаграмма активностей

### 3.2.3 Концептуальная модель предметной области

Концептуальное моделирование необходимо для построения модели, отражающей основные понятия предметной области, их структуру и взаимосвязи между ними. Результатом концептуального моделирования является диаграмма понятий. Диаграмма понятий является частным случаем диаграммы классов.

На рисунке 3.3 изображена диаграмма понятий, описанная на языке UML.

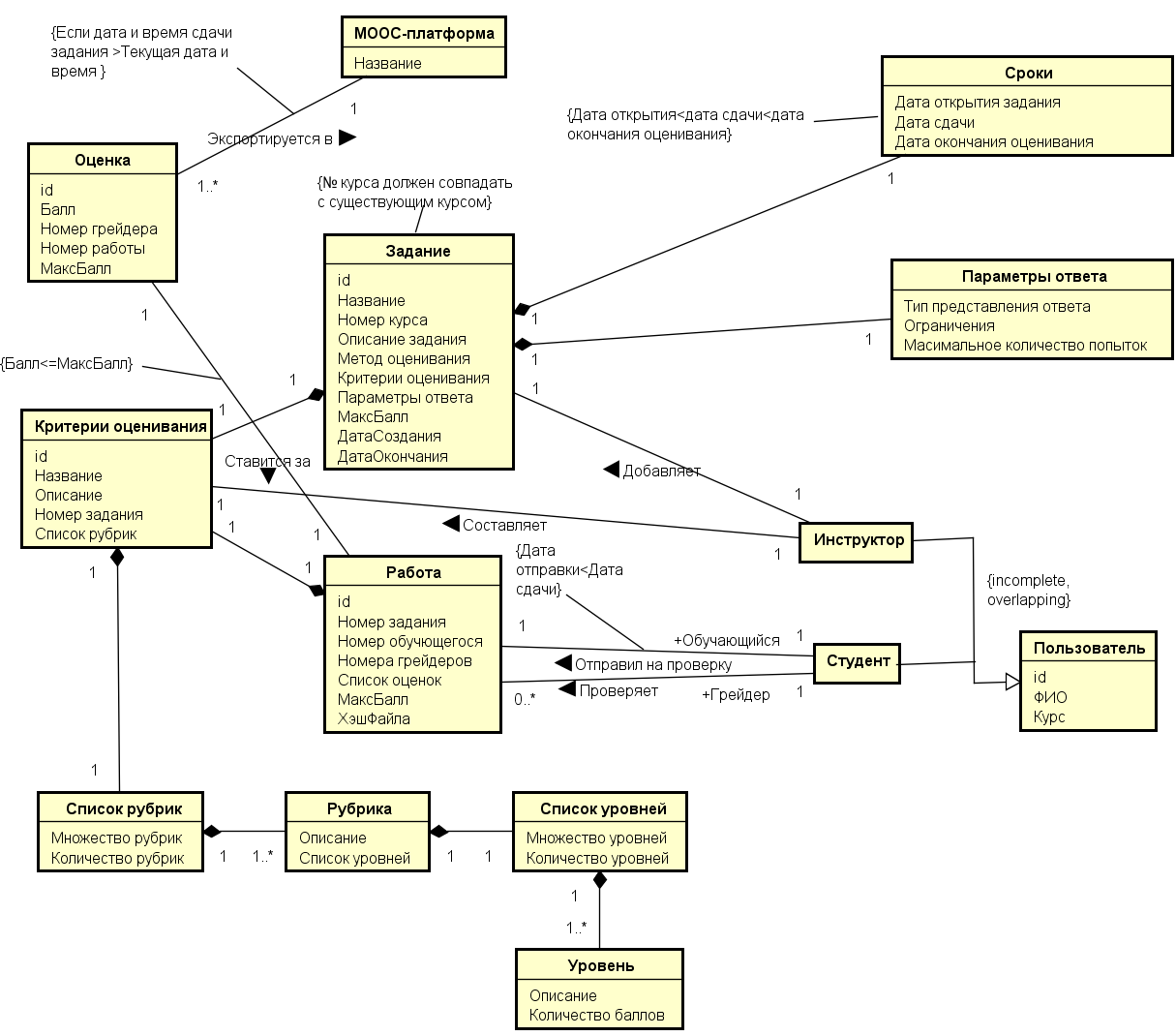


Рисунок 3.0.3 - Диаграмма понятий

### 3.2.4 Моделирование поведения системы

Модель поведения системы необходима для того, чтобы определить системные операции и системные сообщения. При моделировании поведения системы также описывается взаимодействие акторов и системы. Модель поведения завершает этап анализа и строится на основе диаграммы прецедентов.

Список системных операций:

1. Настройка параметров оценивания задания (количественное, порядковое оценивание или рубрики).
2. Обучающийся отправил ответ на задание.
3. Грейдер оценил работу обучающегося.
4. Вычисление итоговых консолидированных оценок для всех обучающихся.
5. Экспорт консолидированных оценок за конкретное задание в МООС-платформу.

Одной из самых основных системных операций является операция «Грейдер оценил работу обучающегося».

Описание системной операции «Грейдер оценил работу обучающегося»:

*Имя*: Грейдер оценил работу обучающегося.

*Обязанности*: Она сохраняет оценку, которую выставил грейдер обучающемуся, и в случае, если грейдер проверил все назначенные ему работы, начисляет вознаграждение.

*Ссылки*: Прецедент «Оценивание грейдером работы».

*Примечание*: Использовать самый быстрый способ доступа к БД.

*Исключения*: Отсутствуют.

*Предусловия*: У грейдера есть не оцененные назначенные работы для проверки.

*Постусловия*:

1. Для подтвержденной грейдером оценки создается объект :Оценка(создание экземпляра).
2. Атрибут :Оценка.idРаботы принял значение «idРаботы» (модификация атрибута).
3. Атрибут :Оценка.idГрейдера принял значение «idГрейдера» (модификация атрибута).
4. Атрибут :Оценка.Балл принял значение «Оценка» (модификация атрибута).
5. Объект :Оценка добавляется в объект :СписокОценок (формирование связи).
6. Для вознаграждения грейдеру создается объект :Вознаграждение (создание экземпляра).
7. Атрибут :Вознаграждение.idГрейдера принимает значение «idГрейдера» (модификация атрибута).

На изображена диаграмма последовательностей для системной операции «Грейдер оценил работу обучающегося».

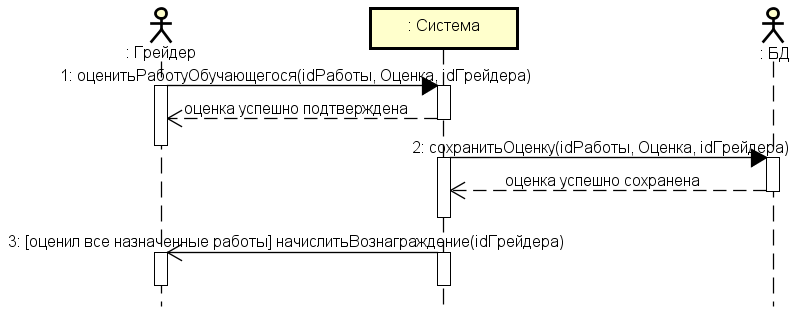


Рисунок 3.0.4 - Диаграмма последовательностей для системной операции "Грейдер оценил работу обучающегося"

## 3.3. Этап проектирования системы

Этап проектирование информационной системы – следующий этап после анализа бизнес-процессов и предметной области. На данном этапе разрабатывается логическое решение поставленной задачи.

Цель этапа проектирования: ответить на вопрос «как добиться требуемой функциональности системы?».

Основные задачи этапа:

1. Построение диаграмм взаимодействия.
2. Построение диаграмм классов.

### 3.3.1 Проектирование поведения системы

При моделировании поведения системы мы получаем описания системных операций, которые содержат постусловия. Но в постусловиях не отражена информация о том, какие программные объекты ответственны за выполнение системной операции, каким образом эти объекты достигают требуемого состояния информационной системы.

На этапе проектирования необходимо построить диаграммы взаимодействия. Они используются для раскрытия содержания системных операций, а также иллюстрируют процесс обмена сообщения между объектами системы.

На рисунке 3.5 изображена диаграмма сотрудничества для системной операции «Грейдер оценил работу обучающегося».

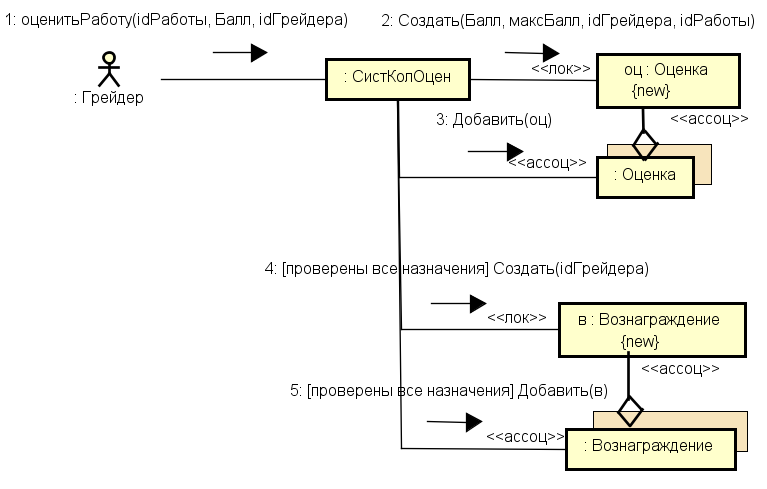


Рисунок 3.0.5 - Диаграмма сотрудничества для системной операции "Грейдер оценил работу обучающегося"

### 3.3.2 Проектирование статической структуры системы

При проектировании статической структуры системы необходимо построить диаграмму классов, которая отображает связи между классами системы, а также атрибуты и методы каждого класса.

На рисунке 3.6 изображена UML-диаграмма классов для системы коллегиального оценивания.

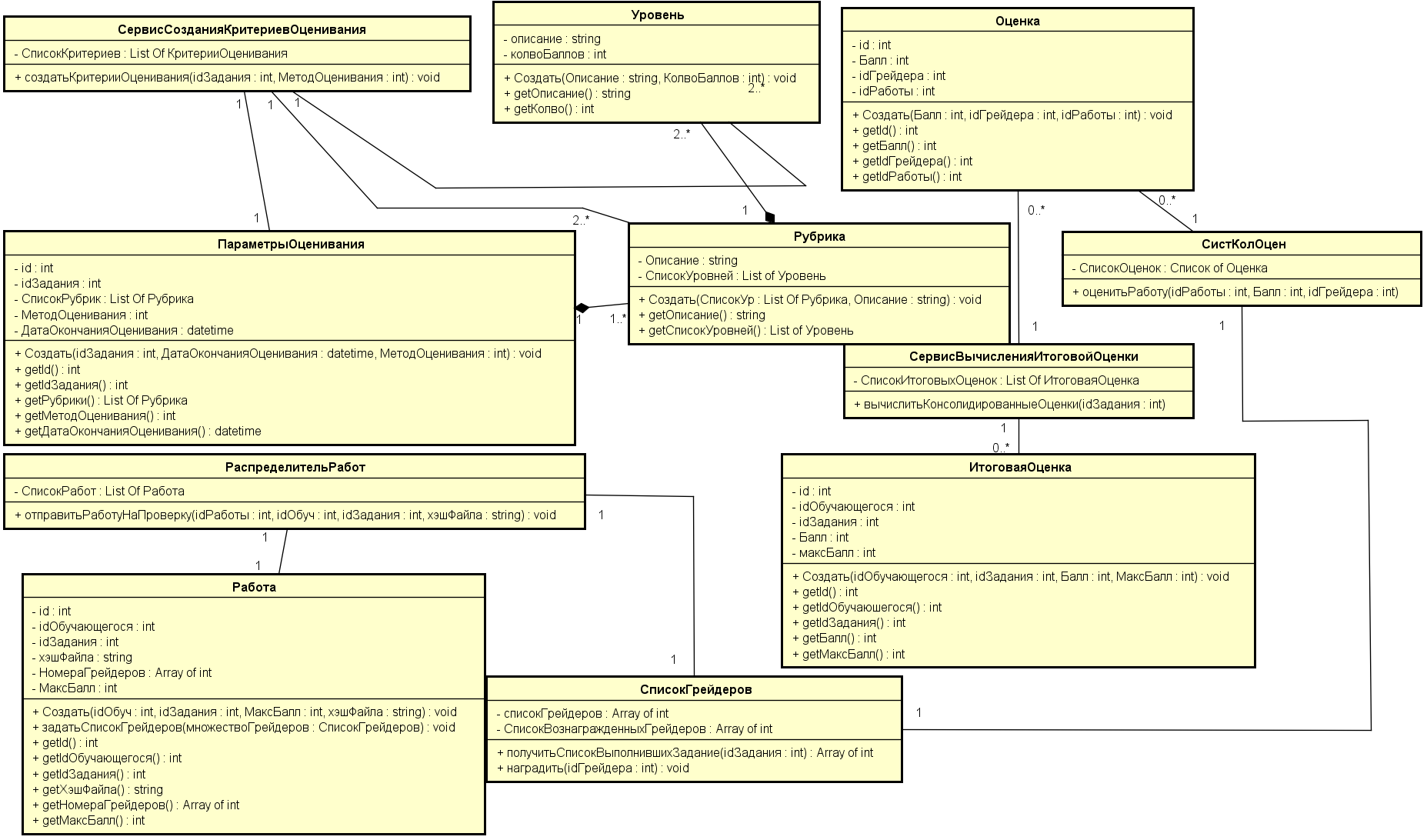


Рисунок 3.6 - Диаграмма классов

## 3.4 Этап реализации

После этапов анализа и проектирования у разработчика информационной системы есть понимание того, что надо сделать и того, как это сделать. Модель реализации завершает описание системы и акцентирует внимание на ее физической структуре.

Для построения модели реализации в UML используются:

1. Диаграмма компонентов.
2. Диаграмма развёртывания.

Так же перед построением модели реализации необходимо уже выбрать программные и аппаратные средства для реализации информационной системы.

### 3.4.1 Диаграмма компонентов

Диаграмма компонентов – граф компонентов и зависимостей между ними. Данный вид диаграмм необходим для отображения зависимостей между программными модулями, включая компоненты исходного и выполняемого кода.

ПЕРЕДЕЛАТЬ

На рис. 3.7 изображена диаграмма компонентов информационной системы.

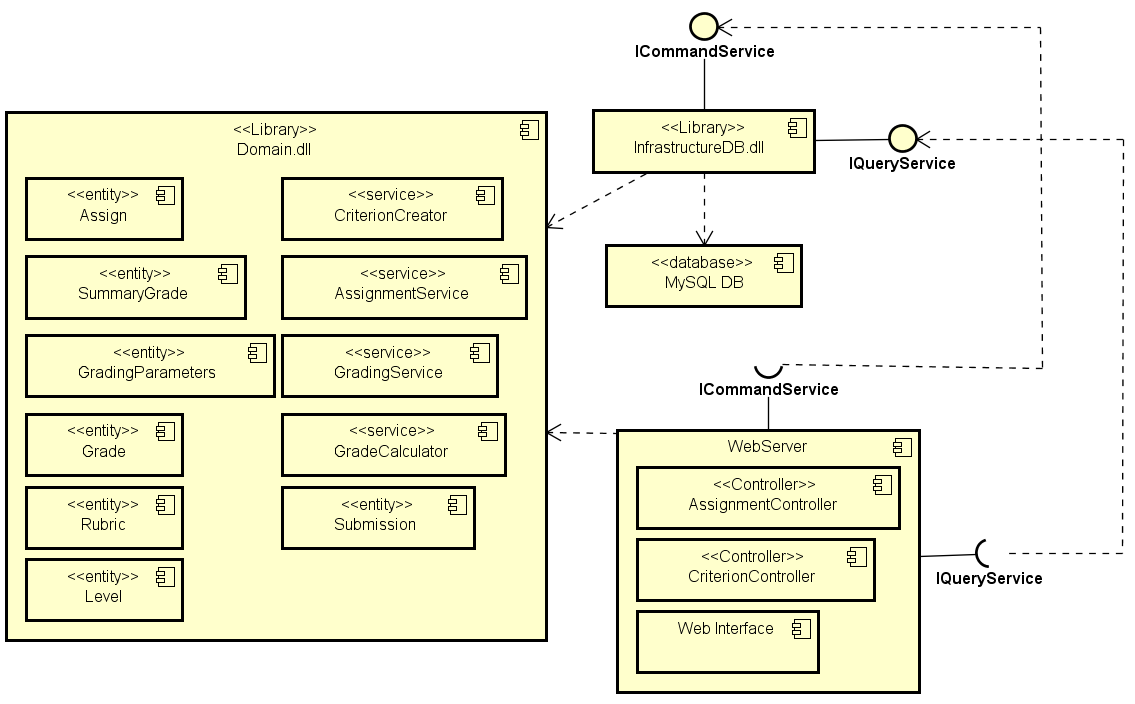


Рис. 3.7. Диаграмма компонентов

### 3.4.2 Диаграмма развертывания

Модель развертывания показывает описание состава, характеристик и топологии аппаратных средств, распределение компонентов между ними.

На рис. 3.8 изображена диаграмма развертывания информационной системы.

Рис. 3.8. Диаграмма развертывания

# 4 Тестирование PeerRank

Тестирование алгоритма PeerRank, который вычисляет итоговые консолидированные оценки для всех обучающихся, происходило на двух случаях:

1. Случай с «плохими» и «хорошими» студентами;
2. Случай, который более приближен к реальным данным, где один недобросовестный студент занижает оценки студентам, которые ответственно подходят к процессу оценивания.

В каждом случае на вход подавалась одна матрица оценок, где оценки варьируются от 0 до 1, затем параметры алгоритма α и β варьировались. При чем α+β≤1. На выходе программа выдавала документ Excel с 15 листами, на каждом из которых приводились: параметры α и β, матрица оценок, итоговые оценки студентов.

Рассмотрим случай с «плохими» и «хорошими» студентами. В этом случае «плохой» студент под номером 1 ставит максимальную оценку всем студентам, а «хорошие» студенты ставят максимальную оценку «хорошим» студентам и ставят 0 «плохому».

На рисунке 4.1 изображен скриншот листа документа Excel, где хранятся результаты работы алгоритма PeerRank с параметрами α=0,9, β=0,1. В i-ой строке находятся оценки, которые поставили другие студенты i-ому студенту. В j-ом столбце находятся оценки, которые поставил j-й студент другим студентам.

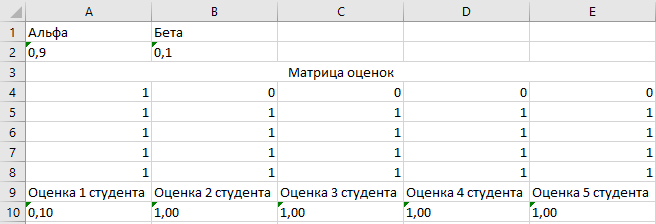


Рисунок 4.1 - Скриншот одного листа документа Excel для первого случая

Из скриншота видно, что «хороших» студентов алгоритм поощрил, а «плохому» студенту он поставил неудовлетворительную оценку, которая была бы еще меньше, но в алгоритме предусмотрено, что каждый студент должен еще оценить сам себя, но эти оценки можно не брать в расчет.

Такой результат PeerRank показывает далеко не всегда. Такая картина получается, если разница между параметрами α и β достаточно существенная.

На рисунке 4.2 изображен скриншот листа Excel, на котором изображены результирующие данные PeerRank при одинаковых параметрах алгоритма. Из скриншота видно, что алгоритм чересчур сильно поощрил «плохого» студента и незначительно снизил оценки «хороших» студентов.

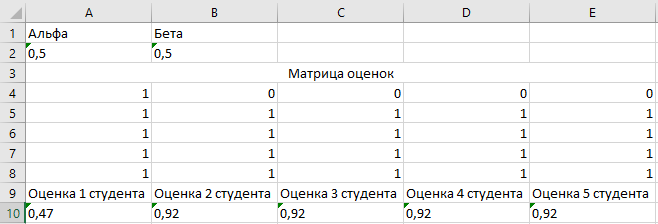


Рисунок 4.2 - Скриншот листа документа Excel для первого случая с одинаковыми параметрами альфа и бета

Рассмотрим второй случай, который более приближен к реальным данным, где один недобросовестный студент занижает оценки студентам, которые ответственно подходят к процессу оценивания.

На рисунке 4.3 изображен скриншот для второго случая с различными параметрами α и β. Из этого скриншота видно, что первому студенту алгоритм PeerRank вычислил итоговую оценку ниже средней, а других студентов алгоритм поощрил и поставил оценку выше среднего.

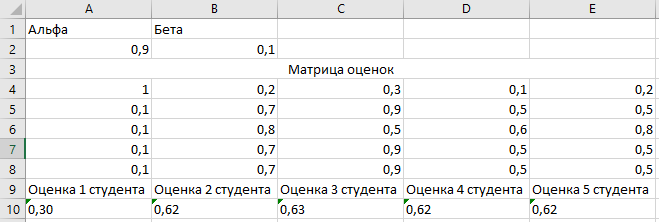


Рисунок 4.3 - Скриншот листа документа Excel для второго случая с различными параметрами альфа и бета

На рисунке 4.4 изображен скриншот листа Excel с результатами работы PeerRank для второго случая с одинаковыми параметрами α и β. В этом случае получилась немного другая ситуация, в отличии от первого случая. PeerRank значительно повысил оценку для добросовестных студентов и незначительно повысил для недобросовестного, но эта оценка все еще неудовлетворительная.

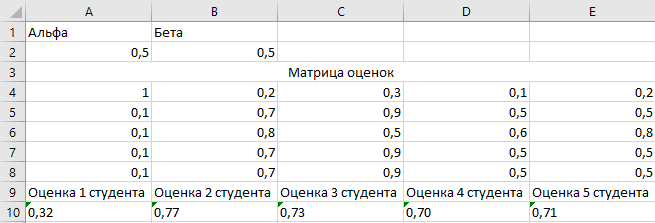


Рисунок 4.4 - Скриншот листа документа Excel для второго случая с одинаковыми параметрами альфа и бета

Из рассмотренных выше двух случаев можно сделать вывод о том, что результаты работы алгоритма сильно зависят от входных параметров α и β. И при выборе существенно отличающихся α и β PeerRank показывает результаты существенно лучше, чем при вычислении итоговой оценки при помощи среднего арифметического от всех оценок.

# Заключение

Подводя итог проделанной работы, можно говорить о следующих результатах:

1. Проанализированы существующие инструменты для коллегиального оценивания на МООС-платформах. Выявлено, что возможность добавления заданий с использованием коллегиального оценивания присутствует далеко не во всех платформах;
2. Проанализированы инструменты для коллегиального оценивания в LMS Moodle. Выявлено, что существующего инструментария платформы не хватает для полноценного коллегиального оценивания;
3. Рассмотрены существующие подходы к коллегиальному оцениванию. Выявлено, что наиболее подходящим для системы коллегиального оценивания является количественный подход;
4. Рассмотрены существующие алгоритмы вычисления итоговых консолидированных оценок для обучающихся, использующие различные подходы к коллегиальному оцениванию;
5. Выбран наиболее подходящий алгоритм вычисления итоговой оценки PeerRank;
6. Выработаны и обоснованы требования к системе коллегиального оценивания;
7. Разработан проект информационной системы, позволяющий внедрить количественное коллегиальное оценивание в LMS Moodle;
8. Протестирован выбранный алгоритм вычисления итоговых консолидированных оценок PeerRank;
9. Проанализированы полученные результаты тестирования алгоритма PeerRank.

# Список литературы