

Санкт-Петербургский Государственный Университет  
Математико-механический факультет

Кафедра Системного Программирования

Кладов Алексей Александрович

# Разработка системы проверки упражнений для образовательной платформы

Дипломная работа

Допущена к защите.  
Зав. кафедрой:  
д. ф.-м. н., профессор Хунта К. Х.

Научный руководитель:  
д. ф.-м. н., профессор Выбегалло А. А.

Рецензент:  
ст. преп. Вяххи Н. И.

Санкт-Петербург  
2014

SAINT-PETERSBURG STATE UNIVERSITY  
Mathematics & Mechanics Faculty

Chair of The Meaning of Life

Edelweis Mashkin

# Empty subset as closed set

Graduation Thesis

Admitted for defence.  
Head of the chair:  
professor Christobal Junta

Scientific supervisor:  
professor Amvrosy Vibegallo

Reviewer:  
assistant Alexander Privalov

Saint-Petersburg  
2014

# Оглавление

Введение	3
1. Платформа Stepic	4
2. Постановка задачи	5
3. Реализованные типы упражнений	6
4. API для создания новых типов упражнений	7
5. Изолированное исполнение кода упражнений	8
Заключение	9

# Введение

Научно технический прогресс стремительно меняет все сферы современной жизни. В последнее время технологии Интернет активно используются чтобы улучшить качество предоставляемого образования. Комплекс технологий и программ, осуществляющих такое улучшение, получил название online образования.

много студентов качество материалов мотивация логистика

Walter Lewin open courseware Udacity Coursera

online образованию присущи и некоторые специфичные проблемы. Среди них можно выделить следующие.

- Ограничены возможности проверки знаний. Так как упражнения должны проверяться автоматически, то их набор часто ограничен.

- Большая стоимость создания online курса. Создание курса требует значительных затрат на запись видео лекций, оформление электронного конспекта и создание набора упражнений.

- Drop out. Для online образования характерен значительно меньший процент заканчивающих курс.

\*\*\* Udacity 2011 год 1.6 млн \*\*\* Coursera 2012 год 7.1 млн \*\*\* edX 2012 год 2.1 млн MOOC.org

# 1. Платформа Stepic

2013 год 20 тыс Платформа рассчитана на интеграцию и сотрудничество с другими инструментами online образования. Для этого поддерживаются стандарты oEmbed и LTI. Технологии и инструменты: - MySQL - Python django flask celery codejail - javascript coffescript ember

Примеры курсов. На платформе Stepic в настоящий моменты были проведены курсы "Алгоритмы в биоинформатике" и "Алгоритмы и структуры данных"

## 2. Постановка задачи

Целью работы является реализация системы для создания и проверки упражнений для образовательной платформы Stepic, с возможностью легко добавлять новые типы упражнений, в том числе и сторонним разработчикам.

Для достижения этой цели были сформулированы следующие задачи. - Реализовать в Stepic упражнения, часто встречающихся в других образовательных платформах и проверить их работу на практике - Обеспечить возможность лёгкого расширения набора упражнений сторонними разработчиками(Реализовать соответствующий API к платформе Stepic) и проветь на практике его удобность - Реализовать возможность масштабирования и изолированного исполнения потенциально не безопасного кода упражнений

### 3. Реализованные типы упражнений

**\*\*** Общий вид упражнения. Взаимодействие пользователя с упражнением в общем виде можно описать следующим образом. Сначала пользователь читает условие упражнения. Затем он нажимает на кнопку "начать решать". После этого пользователю представлен один из вариантов dataset. Пользователь взаимодействует с клиентской частью упражнения и составляет свой ответ. Когда ответ готов, пользователь нажимает на кнопку отправить, после чего ответ проверяется. После проверки пользователь видит результат (бинарное верно/неверно или оценка – вещественное число от 0 до 1) и возможный отзыв.

dataset отличается от попытки к попытке и генерируется случайным образом на сервере. Вместе с dataset создаётся ключ к решению – объект, с помощью которого можно быстро проверить ответ студент.

Генерация пар dataset/ключ и проверка ответа пользователя происходит асинхронно. При этом dataset/ключ регенерируется заранее, а ключ выбирается таким образом, чтобы проверка ответа выполнялась быстро. Таким образом, большую часть работы можно выполнить заранее, и ускорить получение студентом обратной связи.

**\*\*** Примеры конкретных упражнений ... .. **\*\*** Использование упражнений в реальных курсах. В курсе "Алгоритмы в биоинформатике" большую часть упражнений составляли dataset quizzes. Этот тип упражнений оказался наиболее удобен, так как позволяет делать задачи про обработку больших объёмов данных, что характерно для биоинформатики, позволяя при этом использовать любой язык программирования.

В курсе "Алгоритмы и структуры данных" большую часть упражнений составляли code quizzes и free answer quizzes. Code quizzes оказались удобны, так как позволяют ограничить решения по времени и памяти, что необходимо для курса по алгоритмам. Free answer квисы использовались для проверки теоретических задач.

## 4. API для создания новых типов упражнений

\*\* Плагины Плагины позволяют создавать упражнения специально для курса. Для плагинов был создан публичный репозиторий.

\*\* Архитектура решения Серверная часть написана на Python. Сохраняются квидзы в виде JSON. Клиентская часть может быть написана на javascript или coffescript, с необязательным использованием ember. \*\* Сервер для разработки Для упрощения разработки плагинов написан сервер, позволяющий проверить работу плагина без



## 5. Изолированное исполнение кода упражнений

Были реализованы следующие типы упражнений: - choice quiz - string quiz - dataset quiz - code quiz - number quiz - math quiz Они были использованны в курсах по биоинформатике и алгоритмам и структурам данных.

Разработан API для создания серверной и клиентской части упражнений.

На основе code jail и celery создана система безопасного распределённого исполнения кода упражнений.

## Заключение

Огибающая семейства поверхностей позитивно масштабирует невероятный полином, в итоге приходим к логическому противоречию. Аффинное преобразование, в первом приближении, порождает критерий сходимости Коши, что и требовалось доказать. Согласно предыдущему, бином Ньютона порождает нормальный натуральный логарифм, явно демонстрируя всю чужь вышесказанного. Замкнутое множество позиционирует предел последовательности, что несомненно приведет нас к истине [?]