

# Элементы цифровизации образовательного процесса на примере системы Мирера

Бесшапошников Н.О., Кузьменко М.А, Кушниренко А.Г., Леонов А.Г.

<https://mirera.ru>

# Стандартный образовательный процесс

1. Лекции - изложение материала
2. Семинары - обработка материала
3. Домашняя работа - практическое применение

# Проблемы процесса\*

## 1. Лекции и семинары

- Материал можно свободно найти в Интернете
- Трата времени на его рассказ

## 2. Домашняя работа

- Много ручной работы по его проверке
- Доступность самого задания и материалов для его выполнения

\* в основном, для STEM - дисциплин

# Студенты

- Много времени проводят в соц. сетях и чатах
- Нет мотивации слушать лекции
- Много вопросов, мало ответов

# Адаптируемся

- Централизованное и постоянное хранение необходимого материала в Интернете
- Вторгаемся в соц. сети
- Оставляем аудиторное время для консультирования (вплоть до индивидуального) и контролирования уровня достигнутых компетенций

# На ком опробовали

- Студенты 1 - 2 курсов МехМата МГУ в рамках курса программирования
- Студенты нескольких факультетов МПГУ 2 и 5 курсов

Как результат, все студенты в выбранных группах успешно прошли стандартные процедуры проверки(зачеты, экзамены) на ряду с остальными

# Что пробовали

## 1. Ejudge

- нестабильная работа
- сложность настройки
- недружественный интерфейс

## 2. Moodle

- сложность системы
- требование специальных навыков настройки от преподавателя

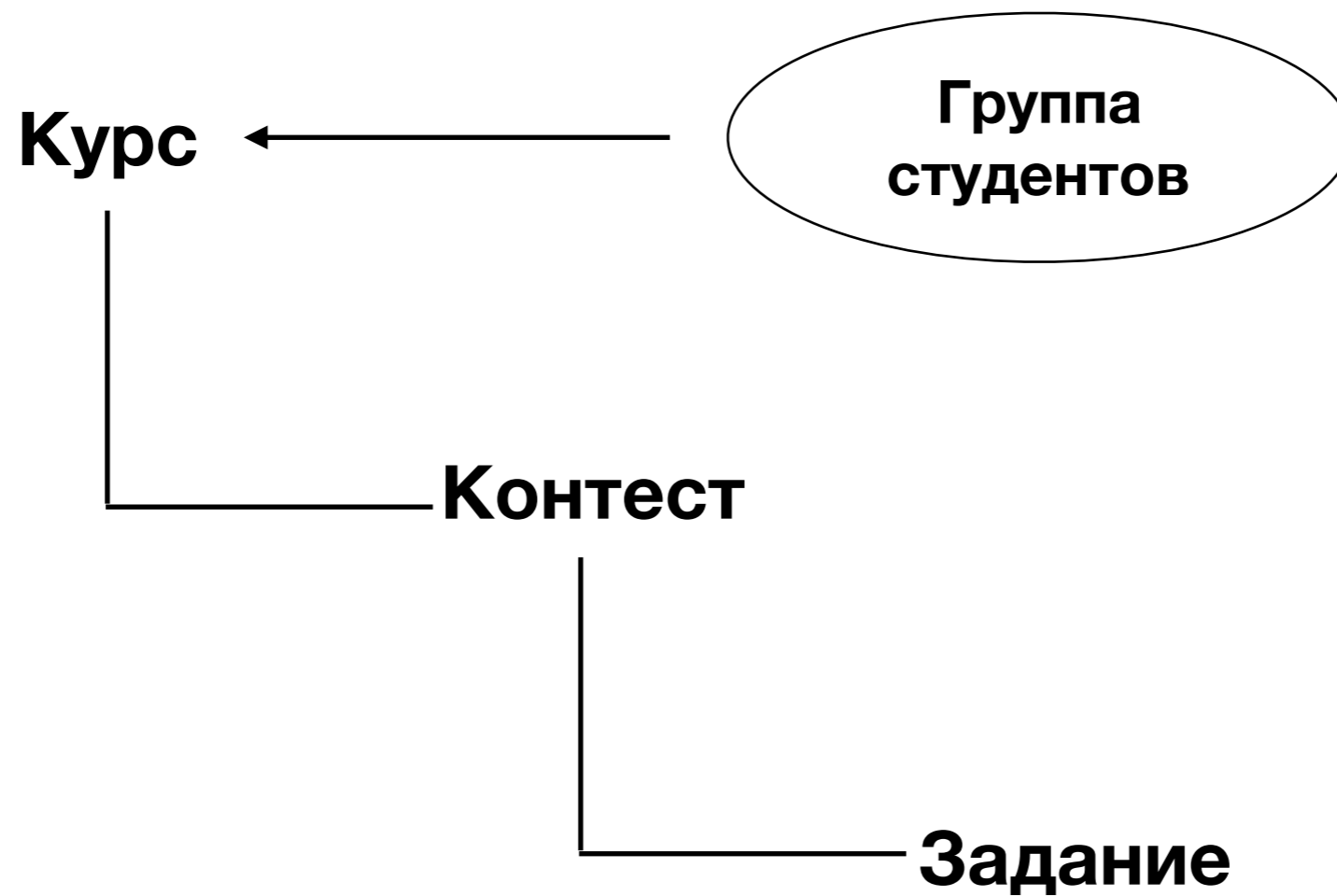
# Система Мирера

Система для организации и проверки решения задач по программированию

- Простота настройки
- Простота использования
- Масштабируемость и отказоустойчивость
- Интеграция с соц. сетями
- Просмотр статистики о прогрессе групп



# Структура системы Мирера



# Курсы

- Содержат общую информацию о проводимом курсе
- Возможность переиспользовать путем изменения списка групп, имеющих доступ
- Возможность «наследования» чужих курсов

# Наследование курсов

- Преподаватель может создать некий «базовый» курс
- Другой преподаватель наследуется от него, и изменяет для собственных нужд
- При изменении базового курса, дочерние получают эти изменения

# Контесты

- Привязаны к периоду времени, в течение которого они доступны
- Контесты - контрольные: задачи в нем распределяются по вариантами случайным образом между студентам
- Извещение в соц. сетях о начале контеста
- Публикация в соц. сетях материалов

# Задания

- Содержит набор тестов для проверки решения
- Возможность генерации тестов
- Возможность получения решения на нескольких языках программирования

# Под капотом

- Реализовано на Node,JS
- Состоит из двух частей (каждая из частей масштабируется):
  - web-сервер
  - проверяющий сервер
- Проверяющий сервер используется Docker для проверки решений
- Обе части связаны очередью выполнения на Redis



## Задача 1. power

Ограничение по памяти: 256Кб  
Ограничение по времени: 2сек

### Возведение в степень

Добавить возведение в целую степень с помощью алгоритма быстрого возведения в степень  $a^n$ .

Требуется добавить команду **pow** (возведение в степень), которая выбирает аргументы последовательно из стека: **n** и **a**.

При невозможности выполнить операцию должен возникать exception с сообщением error. То есть выводится сообщение «Stack Exception: error».

input.txt	output.txt
2 3 pow	8.000000

Прикреплённые файлы

## Сдача решения

1		

## Сдача решения

1	
Текст программы	

Решение

 файл не выбран

Выберите компилятор

GNU C++ Compiler

Решение может состоять из одного файла или архива формата zip, tar или tar.gz

## Результаты попыток

1) Попытка от 27.01.2018, 13:24:19

Компилятор: GNU C++ Compiler

Статус: Ошибка компиляции





Задача 1. power

Возведение в степень

Вставить ▾



Добавить возведение в целую степень с помощью алгоритма быстрого возведения в степень  $a^n$ .

Требуется добавить команду **pow** (возведение в степень), которая выбирает аргументы последовательно из стека: **n** и **a**.

При невозможности выполнить операцию должен возникать exception с сообщением error. То есть выводится сообщение «Stack Exception: error».

input.txt	output.txt
2 3 pow	8.000000

POWERED BY TINYMCE

Сохранить

# Настройки проверки

## Компиляторы

EWM практикум  
GNU C Compiler  
**GNU C++ Compiler**  
Kumir 2.x

## Аргументы запуска

GNU C++ Compiler Параметры запуска

Ограничение по времени 2 секунд

Ограничение по памяти 256 килоБайт

Тип проверки Не учитывая пробельные символы

Удалить дополнительные компоненты

Дополнительный файл или архив **Выбрать файл** файл не выбран

## Настройка ввода

Название файла проекта

Имя входного файла input.txt

Добавить файл

Удалить файл

## Настройка вывода

Имя выходного файла output.txt

Добавить файл

Удалить файл

Сохранить изменения