

# Разработка системы проверки упражнений для образовательной платформы

Алексей Кладов

группа 545

руководитель Луцив Д. В.

рецензент Вяххи Н. И.

Санкт-Петербургский государственный университет

4 июня 2014 г.

- Udacity
- Coursera
- edX

Много студентов  $\Rightarrow$

- Масштабирование лекций.
- Масштабирование упражнений.

## Статус

- 2013 год
- 23000 студентов
- 2 курса на Coursera
- курсы CS Center

## Технологии

- Linux
- Python 3, Django, Celery, Codejail
- Django **REST** framework
- CoffeeScript, Ember

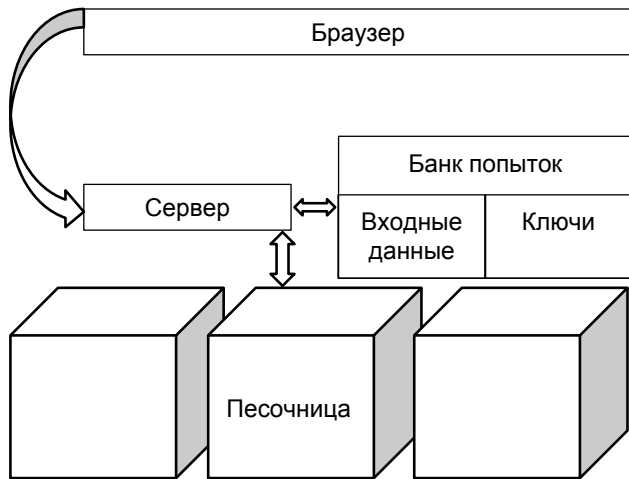
# Постановка задачи

Реализовать расширяемую систему для создания и проверки упражнений для образовательной платформы Stepic.

## Подзадачи

- Разработать фреймворк для расширения набора типов упражнений сторонними разработчиками.
- Реализовать с помощью фреймворка часто встречающиеся типы упражнений.
- Обеспечить масштабируемое и изолированное исполнение потенциально небезопасного кода упражнений.

# Фреймворк

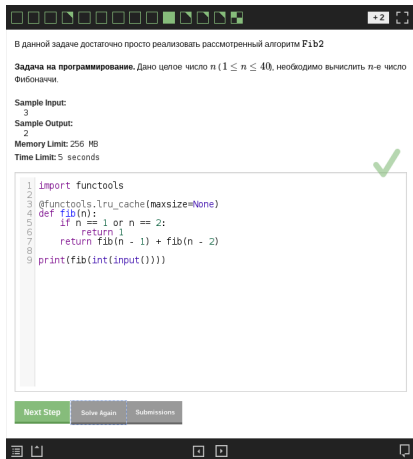


Архитектура решения

# Примеры типов упражнений

- Choice
- Code
- Dataset
- Free Answer
- Math
- Number
- Sorting
- String
- **Admin**

Используются в текущих курсах!



The screenshot shows a coding problem interface. At the top, there's a toolbar with icons for undo, redo, and other editing functions. Below the toolbar, the problem description is in Russian: "В данной задаче достаточно просто реализовать рассмотренный алгоритм Fib2". The task is to calculate the n-th Fibonacci number for  $1 \leq n \leq 40$ . It includes sample input (3) and output (2), memory limit (256 MB), and time limit (5 seconds). A green checkmark is visible on the right side of the code editor. The code editor contains the following Python code:

```
1 import functools
2
3 @functools.lru_cache(maxsize=None)
4 def fib(n):
5     if n == 1 or n == 2:
6         return 1
7     return fib(n - 1) + fib(n - 2)
8
9 print(fib(int(input())))
```

At the bottom, there are three buttons: "Next Step", "Solve Again", and "Submissions".

Пример упражнения (Code)

# Изоляция и масштабирование

- Расширение Codejail (мультиязычность, сообщения об ошибках, коммуникация...)
- Создание профилей Apparmor.
- Масштабируемость при помощи Celery.
- Планы: управление конфигурациями.

# Результаты

- ✓ Разработан фреймворк для создания новых типов упражнений.
- ✓ С помощью фреймворка реализовано 9 типов упражнений, которые успешно использованы в крупных курсах. Один тип упражнения был разработан сторонним разработчиком.
- ✓ На основе Celery и Codejail создана система масштабируемого и безопасного исполнения кода.