

# Проект “Простейшие задачи по Программированию”

Выполнили: Нагибин И.С. 0310, Ефремов А.Д. 2304, Затримайлов Д.Р. 2304, Ишутин О.В. 2304,  
Ламашовский Д.В. 2303, Николаева Т.Д. 2304

## План на третью итерацию

- Добавление простых режимов для гибкой настройки задач, исправление логики условий и поддержка пользовательских вариантов;
- Упрощение сборки и запуска проекта через Docker-контейнер;
- Внедрение метрик для оценки процента комментариев в ответах пользователей;
- Реализация механизма задания seed-значений для воспроизводимости результатов тестов;
- Унифицировать текстовые формулировки условий для всех вариантов;
- В задаче со строковыми операциями реализовать генерацию примеров входных/выходных данных;
- Обновление и дополнение описаний новых функций, модулей и режимов работы (CommentMetric).

# Результаты

- Добавлены простые режимы для гибкой настройки задач, исправлена логика условий и реализована поддержка пользовательских вариантов.
- Упрощена сборка и запуск проекта через Docker-контейнеры.
- Внедрены метрики для оценки процента комментариев в ответах студентов.
- Реализован механизм задания seed-значений для воспроизводимости результатов тестов.
- Унифицированы текстовые формулировки условий для всех вариантов задач.
- В задачах со строковыми операциями реализована генерация примеров входных/выходных данных.
- Обновлено и дополнены описания новых функций, модулей и режимов работы , включая CommentMetric.

Демо результатов работы ([Демо](#))

# Методы решения, технологии

- Python для создания классов обертки и интеграции модуля генерации задач
- Moodle для интеграции и управления контентом
- CodeRunner для интеграции модуля генерации задач и создания динамических заданий по программированию
- Twig для шаблонизации и динамической генерации заданий в CodeRunner
- Moodle XML для удобного импорта/экспорта созданных заданий в Moodle
- Python скрипт для “бандлинга” созданной библиотеки и созданию файлов заданий в формате Moodle XML
- Pytest для организации функционального тестирования
- GitHub Actions для автоматизации тестирования и сборки
- Docker для контейнеризации сборки задач

# Планы на следующую итерацию

- Исправление правок заказчика.
- Приведение документации к единообразному виду.
- Улучшить тестирование кода через использование различных входных параметров для классов задач.
- Дополнить вывод процента комментариев новыми метриками (например, использование памяти, среднее время исполнения).