C Koans итерация 2

Разработчики:

- Афанасьев Назар (0310, магистр)
- Бухарин Максим (2382)
- Кочуров Александр (2382)
- Муравин Егор (2382)
- Федоров Михаил (2382)
- Чепасов Дмитрий (2382)

Заказчик: Заславский Марк Маркович

Постановка цели итерации

Сроки итерации: 27.02.2025 - 26.03.2025

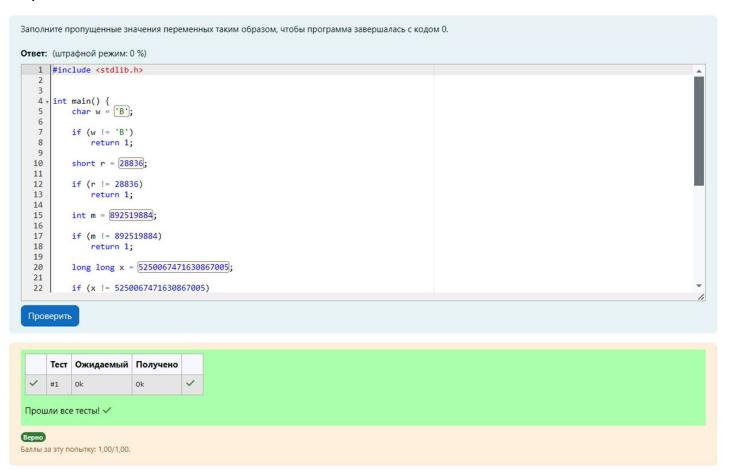
Главная цель итерации — подготовить версию 1 (частично работоспособная версия)

План на итерацию

- Написать метод для генерации условия задания
- Написать метод для генерации шаблона кода в задании
- Написать метод для генерации темплейта для Coderunner
- Объединить в класс методы для генерации заданий
- Обернуть решение в Docker контейнер

Пример готовой задачи

Правильный ответ



Пример готовой задачи

Неправильный ответ



Обёртка в Docker контейнер и вывод опций программы

```
PS D:\> docker images
REPOSITORY
                  TAG
                            IMAGE ID
                                           CREATED
                                                         SIZE
koans_generator latest
                            54be16d314d4
                                           3 hours ago
                                                         48.8MB
PS D:\> docker run --rm koans_generator -h
usage: main.py [-h] [--method METHOD [METHOD ...]] [--name NAME [NAME ...]]
               [--seed SEED]
options:
 -h, --help
                       show this help message and exit
  --method METHOD [METHOD ...]
                        methods available: code_tmp - get_code_template, cond_task - get_condition_task, tmp_coderunner - get_template_coderunner
 --name NAME [NAME ...]
                        name tasks
  --seed SEED
                        seed generation number
```

Пример работы программы (фрагмент)

```
PS D:\> docker run -rm koans_generator -method cond_task code_tmp tmp_coderunner -name basic_task --seed 1235
Condition, name: basic_task:
Заполните пропущенные значения переменных таким образом, чтобы программа завершалась с кодом 0.
Coderunner, name: basic_task:
#include <stdlib.h>
int main() {
    char w = {[1]};
    if (w != 'B')
        return 1;
    short r = {[2]};
    if (r != 28836)
        return 1;
    int m = \{[3]\};
    if (m != 892519884)
        return 1;
    long long x = \{[4]\};
    if (x != 5250067471630867005)
        return 1;
    unsigned int g = \{[5]\};
    if (g > 2554311362)
        return 1;
    double n = {[6]};
    if (abs(n - 98.478) >= 0.000001)
        return 1;
    double o = {[7]};
    if (abs(o - (-45.735)) >= 0.000001)
        return 1;
    return 0;
Code, name: basic_task:
import subprocess, sys, re
student_answer = {{ STUDENT_ANSWER }}
template = {str(ind + 1): s for ind, s in enumerate(student_answer)}
code = """
#include <stdlib.h>
int main() {
   char w = {[1]};
    if (w != 'B')
        return 1;
    short r = {[2]};
    if (r != 28836)
        return 1;
```

Результаты

- Была подготовлена версия 1
- Были написан функционал по генерации условий, темплейтов Coderunner и шаблонов кода для задачки Basics (Базовые конструкции и структуры)
- Решение было обёрнуто в Docker контейнер
- Была написана инструкция по сборке и запуску

Планы на следующую итерацию

- Реализовать тему Pointers (Указатели)
- Реализовать тему Functions (Функции)
- Реализовать тему Arrays (Массивы)
- Реализовать тему Strings (Строки)
- Повысить читабельность результатов вывода программы и добавить возможность вывода в файл