Данный документ предназначен для объяснения принципа работы модуля обработки языка PHP на платформе Django учебного комплекса CAPPA. Версия PHP используется 7.4.6 в качестве тестов и пробного варианта написания скриптов для последующей обработки использовался PyCharm Community Edition. (Выполнения языка и тестирования производились через консоль)

Основной принцип работы модуля лежит в обработке языка PHP на сервере через консоль системы (Windows или Linux). В данном объяснении примера работы

была использована ОС Windows. Для того, чтобы наша система могла обработать PHP, его сначала нужно установить на компьютер разработчика, а также прописать его в PATH. После этого можно протестировать работу с помощью команды

php example.php

Где example.php наш файл с каким-либо скриптом PHP, к примеру:

echo "Hello World";

Данный скрипт выведет нам в консоль сообщение Hello World.

Это будет работать если мы введем в консоль команду напрямую, если же мы хотим, чтобы это заработало на учебном комплексе мы должны использовать

субпроцессы. С помощью субпроцессов мы можем в фоновом режиме открыть консоль ввести наш код и получить его результат. В данном случае код - это не сам скрипт, а вызов ранее

генерируемого документа со случайным названием, в котором находится код пользователя.

\*Отрывок из модуля с субпроцессом\*

p1 = subprocess.Popen(['php', tmp.file\_php\_dir],

stdin=subprocess.PIPE,

stdout=subprocess.PIPE,

stderr=subprocess.PIPE,

cwd=settings.TMP\_DIR)

Так выглядит работа с субпроцессом в tmp.file\_php\_dir мы получаем путь до нашего файла, перед этим передав php, чтобы консоль "поняла", что мы хотим обработать язык PHP. Нам нужны значения которые мы получим в stdout и stderr это то, что мы хотим получить, а именно вывод и если возникнут ошибки, то и их тоже.

Название файла генерируется командой:

filename = uuid.uuid4()

После чего добавляем его в путь (предварительно добавив ему название расширения - php) и при вызове в субпроцессе мы получим сам путь до файла.

\*Команды для добавления сгенерированного названия файла в путь\*

self.filename\_php = '%s.php' % filename

self.file\_php\_dir = os.path.join(settings.TMP\_DIR, self.filename\_php)

Нам нужно сделать запись в файл кода нашего пользователя, командами:

file = open(self.file\_php\_dir, "wb")

input\_tag = f'<?php {input} ?>'

content\_tag = f'<?php {content} ?>'

file.write(bytes(input\_tag, 'utf-8'))

file.write(bytes(content\_tag, 'utf-8'))

file.close()

Открываем в режиме бинарной записи, делаем запись в бинарном виде предварительно добавив теги для корректной работы языка после чего закрываем файл.

Также не забываем удалить файл после работы с ним, а также убить наш субпроцесс.

p1.kill()

tmp.remove\_file\_php()

В качестве входных данных в интерфейсе учебного комплекса в модуле PHP было решено использовать - переменные с определенными значениями, в связи с особенностями языка, в тестах аналогично используются переменные.

Это базовая часть работы модуля, остальные его компоненты базируется на данном коде и могут дорабатываться всевозможными способами, например, отлов ошибок или обработка кода на нескольких тестах.