

# Задания по курсу Python

## Задание 4

Д.В. Иртегов

26 апреля 2018 г.

Задачи необходимо сдать до 26 мая. Решения необходимо сдавать путем отправки pull request в каталог problems-4 репозитория

<https://github.com/dmitry-irtegov/NSUPython2018>.

Датой сдачи задания считается дата отправки первого pull request. Если запрос не принят из-за моих замечаний, у вас есть неделя на их исправление.

Если запрос принят, задание считается засчитанным. Если запрос не принят, в комментарии вы можете узнать, почему.

В одном запросе следует отправлять не более одного решения. Если решение состоит из нескольких файлов, в запрос должны быть включены они все. Все запросы одного студента должны отправляться в каталог с именем, соответствующим его учетной записи. Например, для задачи 3 из группы задач 2, сдаваемой студентом v-purkin, рекомендуемое имя файла problems-3/v-purkin/task3.py.

**Задача 1.** Существует шуточный «закон Философии» для статей Википедии: если переходить по первой нормальной ссылке в статье, то рано или поздно мы придем на статью о философии. Ваша задача заключается в том, чтобы проверить этот закон.

Для этого нужно написать программу, которая получает на вход ссылку на статью Википедии, а потом циклически переходит по первой нормальной ссылке и повторяет эту операцию (до тех пор, пока не будет достигнута статья о философии, или ссылки не зайдутся). Нормальной ссылкой будем называть ссылку, которая находится в основном содержании статьи, не в инфобоксах и не в служебных блоках, написана синим цветом (красный соответствует несуществующей статье), не курсивом, не является сноской и не находится в скобках. Обратите внимание, что для проверки нормальности не обязательно разбирать таблицы стилей и проверять цвет и т.п., достаточно сделать, чтобы программа работала для текущей верстки википедии (напри-

мер, можно использовать атрибут `class` у тегов).

Для удобства проверки сделайте, чтобы последовательность переходов выводилась в стандартный поток вывода.

**Внимание!** Чтобы не создавать большую нагрузку на сервер Википедии, сделайте так, чтобы ваша программа делала не более 2-х запросов в секунду. Для этого воспользуйтесь функцией `sleep` из модуля `time`.

**Задача 2.** Вам необходимо написать программу с графическим интерфейсом для моделирования игры «Жизнь» (Conway's Game of Life). Игра должна загружать начальную конфигурацию из некоторого файла и отображать на экране текущее состояние игры и панель управления моделированием. Должна быть доступна возможность запуска и паузы моделирования, изменения скорости моделирования (как минимум три скорости: 1x, 2x, 4x). Также должна присутствовать возможность редактирования текущего поля в режиме паузы (самостоятельно выберите удобный способ это делать).

Для файлов сохранения рекомендуется использовать форматы [Life 1.06](#) или [RLE](#).

Разрешенные для использования библиотеки: Tkinter, PyQt, pygame, Kivy, bitarray.

**Задача 3.** С использованием фреймворка Flask реализуйте набор веб-страниц, обеспечивающих регистрацию пользователей на сайте. Неаутентифицированный пользователь должен иметь возможность выбрать уникальное имя и задать для него пароль. Необходимо обеспечить возможность использования русских букв (а лучше, произвольных алфавитных символов Unicode) в именах. После регистрации, пользователь должен иметь возможность заходить на сайт по паролю и видеть имена тех пользователей, которые зарегистрировались после него, но не тех, кто зарегистрировался раньше. Неаутентифицированные пользователи не могут видеть никаких имен. Для хранения данных следует использовать стандартную библиотеку `dbm` или `sqlite`. При первом запуске приложение должно самостоятельно инициализировать файлы БД в текущем каталоге.

Не следует реализовать слишком сложные механизмы аутентификации. Предполагайте, что в production версии приложения доступ к вашим страницам будет происходить через `https`, поэтому пароли и куки защищены от перехвата. Но пароли следует хранить в форме хэшей.

Обратите внимание, что пользователь может ввести в составе имени HTML-теги, код на JavaScript или что-то похожее на них. Это не следует запрещать, но при показе таких имен необходимо проследить, чтобы они показывались на веб-странице буквально, а не интерпретировались как HTML-код.