САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО

Лабораторная работа №5
по Дисциплине "Технологии программирования"

Выполнил: Юсюмбели В. И.
Плохотнюк В. С.
Толстухин М.С.
Группа: Р3411
Преподаватель: Оголюк А. А.

Санкт-Петербург
27 ноября 2019 г.

```
"""
Вход: файл guess.txt содержащий имена для угадывания

(например из http://www.biographyonline.net/people/famous-100.html можно взять имена)


Написать игру "Угадай по фото"

3 уровня сложности:
1) используются имена только 1-10
2) имена 1-50
3) имена 1-100

- из используемых имен случайно выбрать одно
- запустить поиск картинок в Google по выбранному
- получить ~30-50 первых ссылок на найденные по имени изображения
- выбрать случайно картинку и показать ее пользователю для угадывания
  (можно выбрать из выпадающего списка вариантов имен)
- после выбора сказать Правильно или Нет

п.с. сделать серверную часть, т.е. клиент играет в обычном браузере обращаясь к веб-серверу.

п.с. для поиска картинок желательно эмулировать обычный пользовательский запрос к Google
или можно использовать и Google image search API
https://ajax.googleapis.com/ajax/services/search/images? или др. варианты
НО в случае API нужно предусмотреть существующие ограничения по кол-ву запросов
т.е. кешировать информацию на случай исчерпания кол-ва разрешенных (бесплатных)
запросов или другим образом обходить ограничение. Т.е. игра не должна прерываться после N
↪  запросов (ограничение API)


п.с. желательно "сбалансировать" параметры поиска (например искать только лица,
использовать только первые 1-30 найденных и т.п.)
для минимизации того что найденная картинка не соответствует имени

"""

import urllib.request
from urllib.request import Request, urlopen
from lxml import html
import sys


def read_names(filename):
    f = open(filename, "r")

    fl = f.readlines()

    names = []
    for name in fl:
        names.append(name)

    return names


from http.server import BaseHTTPRequestHandler, HTTPServer
import json


class HTTPServerQuiz(BaseHTTPRequestHandler):
    # GET
    def do_GET(self):
        names = read_names(sys.argv[1])
```

```
63
64          import random
65
66          variants = []
67          for i in range(0, 3):
68              variant = names[random.randint(0, 99)]
69              variants.append(variant[:variant.find('\n')])
70
71          answer = variants[random.randint(0, 2)]
72          photo = self.get_photo(answer)[0]
73
74          print(answer + ' - ' + photo)
75
76          response = {
77              'photo': photo,
78              'answer': answer,
79              'variants': variants
80          }
81
82          self.send_response(200)
83          self.send_header('Content-type', 'application/json')
84          self.end_headers()
85          self.wfile.write(bytes(json.dumps(response), encoding='utf-8'))
86
87      def get_photo(self, name):
88          url = "https://www.google.ru/search?site=&tbm=isch&source=hp&biw=1600&bih=1600&q=" +
            ↪  name.replace(" ", "%20")
89
90          req = Request(
91              url,
92              headers={'User-Agent': 'Mozilla/5.0'})
93          html_content = urlopen(req).read()
94
95          tree = html.fromstring(html_content)
96
97          import random
98          return tree.xpath('(.//*[@target="_blank"]/img)[' + str(random.randint(1, 10)) +
            ↪  ']/@src')
99
100
101 def name_foto_quiz():
102     server_address = ('127.0.0.1', 8081)
103     httpd = HTTPServer(server_address, HTTPServerQuiz)
104
105     httpd.serve_forever()
106
107
108 if __name__ == '__main__':
109     name_foto_quiz()
```