Федеральное государственное бюджетное учреждение высшего образование

«Чувашский государственный университет имени И. Н. Ульянова»

Факультет «Информатики и Вычислительной техники»

Кафедра «Вычислительной техники»

Отчет по лабораторной работе №4 по дисциплине

Кроссплатформенные средства разработки программного обеспечения

На тему

«Работа с JSON‐файлами»

Вариант 6

Выполнил студент группы ИВТ 42-16

Константинов К.С.

Проверил преподаватель кафедры ВТ

Галибин С.В.

**Цель работы:**

Целью третьей лабораторной работы является ознакомление с методами работы с JSON‐файлами, а так же получение навыков создания пользовательских клавиатур для viber‐бота.

**Задание**:

В четвертой лабораторной работе необходимо реализовать чат‐бота для мессенджера viber, который предназначен для заучивания иностранных слов. К чат‐боту предъявляются ледующие требования:

1. Чат‐бот состоит из двух основных «экранов». Первый экран выводится в начале общения пользователя с чат‐ботом и сообщает о назначении этого чат‐бота. Данный экран должен содержать как минимум одну кнопку, позволяющую начать раунд вопросов и ответов. При нажатии на кнопку происходит переход ко второму окну.
2. На втором «экране» чат‐бот задает вопрос и предлагает выбрать один из вариантов ответов. Ответ выбирается с помощью кнопок клавиатуры. Необходимо, чтобы было не менее четырех возможных вариантов ответа. Также клавиатура содержит кнопку, позволяющую запросить вариант использования данного слова в предложении.
3. Правильный ответ всегда размещается в случайной позиции, которая меняется при каждом использовании данного слова. Неправильные ответы также выбираются случайным образом.
4. При нажатии на кнопку, запрашивающую пример, чат‐бот должен выводить разные примеры из доступного списка и вновь ожидать действий пользователя.
5. При выборе варианта ответа необходимо сообщить пользователю правильно ли он ответил на вопрос, после чего перейти к следующему вопросу.
6. За один раунд задается 10 вопросов, после чего пользователю сообщается результат – количество правильных и неправильных ответов. Затем чат бот переходит к первому экрану, и пользователь может начать сначала.
7. Список слов для заучивания приводится в JSON‐файле, поставляемым вместе с данными методическими указаниями.
8. Чат бот должен «различать» всех своих пользователей, то есть каждый пользователь может отгадывать разные слова, запрашивать разные примеры и так далее...
9. Web‐приложение должно быть размещено в сети, например, на сайте https://www.pythonanywhere.com

**Текст программы:**

**Main.py**

from flask import Flask, request, Response

from viberbot import Api

from viberbot.api.bot\_configuration import BotConfiguration

from viber\_bot import viber\_bot

app = Flask(\_\_name\_\_)

viber = Api(BotConfiguration(

name='Bilinhual',

avatar='https://dl-media.viber.com/1/share/2/long/vibes/icon/image/0x0/f09b'

'/fc477e80d5306023ccf92a07170886fee98bba96aca04959700207a62cc6f09b.jpg',

auth\_token='4ac9ce371e67d30e-9dc412b2061eb971-fb15ec32858c9525'

))

bot = viber\_bot()

@app.route('/', methods=['POST'])

def incoming():

viber\_request = viber.parse\_request(request.get\_data())

bot.set\_request(viber\_request)

messages = bot.get\_response()

if messages is not None:

viber.send\_messages(bot.get\_user().get\_id(), messages)

return Response(status=200)

if \_\_name\_\_ == "\_\_main\_\_":

app.run(host='localhost', port=80, debug=True)

**User.py**

from game\_round import game\_round

class user:

def clear(self):

self.\_\_game\_point = 0

self.\_\_round.clear()

def get\_id(self):

return self.\_\_id

def get\_name(self):

return self.\_\_name

def next\_round(self):

return self.\_\_round.next\_round()

def get\_current\_word(self):

return self.\_\_round.get\_current\_word()

def get\_current\_keyboard(self):

return self.\_\_round.get\_current\_keyboard()

def get\_score(self):

return self.\_\_game\_point

def set\_score(self, game\_point):

self.\_\_game\_point += game\_point

def \_\_init\_\_(self, user\_id, name):

self.\_\_id = user\_id

self.\_\_name = name

self.\_\_round = game\_round()

self.\_\_game\_point = 0

self.\_\_current\_word = None

**viber\_bot.py**

from viberbot.api.messages import PictureMessage, KeyboardMessage

from viberbot.api.messages.text\_message import TextMessage

from viberbot.api.viber\_requests import ViberMessageRequest

from viberbot.api.viber\_requests import ViberSubscribedRequest

import random

# Бот

import json

from User import user

KeysStart = json.load(open('/home/konstantinovk/viberbot/start\_keyboard.json', encoding='utf-8'))

class viber\_bot:

\_\_unknown\_messages\_collection = ['Не понимаю о чем ты говоришь', 'Данная информация не может быть мною распознана',

'Критическая ошибка в программе. Соощение ошибки: "Убить всех человеков"']

\_\_response\_message = list()

\_\_users = {}

\_\_current\_user = None

# Конструктор класса

def \_\_init\_\_(self):

pass

# получение ботом запроса

def set\_request(self, viber\_request):

if isinstance(viber\_request, ViberSubscribedRequest):

self.\_\_response\_message = self.\_\_hello\_\_message\_\_(viber\_request.user.name)

self.\_\_append\_\_user\_\_(viber\_request.user)

elif isinstance(viber\_request, ViberMessageRequest):

message = viber\_request.message

self.\_\_append\_\_user\_\_(viber\_request.sender)

print(viber\_request.message)

if isinstance(message, TextMessage):

word = viber\_request.message.text

if word[0] == '/':

word = word.replace('/', '')

if word.split(' ')[0].lower() == 'help':

self.\_\_response\_message = self.\_\_help\_\_message\_\_()

elif word.split(' ')[0].lower() == 'start':

self.\_\_response\_message = self.\_\_new\_\_word\_\_message\_\_()

elif word.split(' ')[0].lower() == 'example':

self.\_\_response\_message = self.\_\_example\_message\_\_()

else:

self.\_\_response\_message = self.\_\_unknown\_\_message\_\_()

elif int(word) in [1, 2, 3, 4]:

self.\_\_response\_message = self.\_\_answer\_message\_\_(word.split(' ')[0].lower())

else:

self.\_\_response\_message = self.\_\_unknown\_\_message\_\_()

elif isinstance(message, PictureMessage):

self.\_\_response\_message = TextMessage(text='Очень красиво, ' + viber\_request.sender.name)

else:

self.\_\_response\_message = None

# ответ бота

def get\_response(self):

return self.\_\_response\_message

def get\_user(self):

return self.\_\_current\_user

def \_\_answer\_message\_\_(self, word):

if self.\_\_current\_user.get\_current\_keyboard()['Buttons'][int(word) - 1]['Text'] == '<font color=\"#494E67\" size=\"10\"><b>' + self.\_\_current\_user.get\_current\_word()['translation'] + '</b></font>':

self.\_\_current\_user.set\_score(1)

message = [TextMessage(text='И это правильный ответ! Вы получаете плюс один балл (smiley)')]

return message + self.\_\_new\_\_word\_\_message\_\_()

else:

self.\_\_current\_user.set\_score(0)

message = [TextMessage(text='К сожалению вы ошиблись! Баллов не будет, но вы держитесь(ugh)')]

return message + self.\_\_new\_\_word\_\_message\_\_()

def \_\_example\_message\_\_(self):

# Если примеры кончились

if len(self.\_\_current\_user.get\_current\_word()['examples']) == 0:

return [TextMessage(

text='Примеры кончились (sad)'),

KeyboardMessage(keyboard=self.\_\_current\_user.get\_current\_keyboard())]

# Выбираем случайный пример

message\_text = random.choice(self.\_\_current\_user.get\_current\_word()['examples'])

# Удаляем полученный пример

self.\_\_current\_user.get\_current\_word()['examples'].remove(message\_text)

# Возвращаем сообщение с примеров

return [TextMessage(

text=message\_text),

KeyboardMessage(keyboard=self.\_\_current\_user.get\_current\_keyboard())]

def \_\_hello\_\_message\_\_(self, sender\_name):

messages = [PictureMessage(text="Поздравляю тебя с подпиской, " + sender\_name,

media='https://pngimage.net/wp-content/uploads/2018/06/%D1%87%D0%B5%D0%BB%D0%BE%D0'

'%B2%D0%B5%D1%87%D0%BA%D0%B8-png-%D0%B4%D0%B5%D0%BD%D1%8C%D0%B3%D0%B8-2.png'),

TextMessage(

text='Я создан для того, чтобы помогать людям в изуение новых слов на английском языке')]

return messages + self.\_\_help\_\_message\_\_()

def \_\_help\_\_message\_\_(self):

return [TextMessage(text='Чтобы начать изучение - напиши /start'),

KeyboardMessage(keyboard=KeysStart)]

# получение ответа на неизвестную команду

def \_\_unknown\_\_message\_\_(self):

keyboard = self.\_\_current\_user.get\_current\_keyboard()

if keyboard is None:

keyboard = KeysStart

return [

TextMessage(text=(

self.\_\_unknown\_messages\_collection[random.randint(0, len(self.\_\_unknown\_messages\_collection) - 1)])),

KeyboardMessage(keyboard=keyboard)]

def \_\_append\_\_user\_\_(self, raw\_user):

if self.\_\_users.get(raw\_user.id) is None:

self.\_\_users[raw\_user.id] = user(raw\_user.id, raw\_user.name)

self.\_\_current\_user = self.\_\_users[raw\_user.id]

def \_\_new\_\_word\_\_message\_\_(self):

if not self.\_\_current\_user.next\_round():

message = [TextMessage(text='Раунд закончился. Ваш результат: ' + str(self.\_\_current\_user.get\_score()) + ' из 10'),

TextMessage(text='Спасибо за игру! (like)')] + self.\_\_help\_\_message\_\_()

self.\_\_current\_user.clear()

return message

else:

return [TextMessage(text='Ваше слово: ' + self.\_\_current\_user.get\_current\_word()['word']),

TextMessage(text='Вариатны перевода представлены на клавиатуре'),

TextMessage(text='Удачи!(moa)'),

KeyboardMessage(keyboard=self.\_\_current\_user.get\_current\_keyboard())]

**game\_round.py**

import json

import random

english\_words = json.load(open('/home/konstantinovk/viberbot/english\_words.json', encoding='utf-8'))

class game\_round:

\_\_KeysTask = json.load(open('/home/konstantinovk/viberbot/word\_keyboard.json', encoding='utf-8'))

\_\_current\_word = dict()

def \_\_init\_\_(self):

self.\_\_words = list()

def next\_round(self):

if len(self.\_\_words) == 10:

return False

# Выбираем правильный ответ

new\_word = self.\_\_get\_new\_word()

# Добавляем основное слово к истории слов

self.\_\_words.append(new\_word['word'])

# Неправильные ответы

wrong\_answers = self.\_\_get\_wrong\_answers(new\_word['word'])

for i in range(0, len(wrong\_answers)):

new\_word['wrong\_answer' + str(i)] = wrong\_answers[i]['translation']

# Сохраняем результаты

self.\_\_current\_word = new\_word

self.\_\_gen\_keyboard()

return True

def get\_current\_keyboard(self):

return self.\_\_KeysTask

def get\_current\_word(self):

return self.\_\_current\_word

def \_\_gen\_keyboard(self):

shuffled = list(range(1, 5))

random.shuffle(shuffled)

self.\_\_KeysTask['Buttons'][shuffled[0] - 1]['Text'] = '<font color=\"#494E67\" size=\"10\"><b>' + self.\_\_current\_word['translation'] + '</b></font>'

self.\_\_KeysTask['Buttons'][shuffled[1] - 1]['Text'] = '<font color=\"#494E67\" size=\"10\"><b>' + self.\_\_current\_word['wrong\_answer0'] + '</b></font>'

self.\_\_KeysTask['Buttons'][shuffled[2] - 1]['Text'] = '<font color=\"#494E67\" size=\"10\"><b>' + self.\_\_current\_word['wrong\_answer1'] + '</b></font>'

self.\_\_KeysTask['Buttons'][shuffled[3] - 1]['Text'] = '<font color=\"#494E67\" size=\"10\"><b>' + self.\_\_current\_word['wrong\_answer2'] + '</b></font>'

return self.\_\_KeysTask

def clear(self):

self.\_\_words.clear()

def \_\_get\_new\_word(self):

word = random.choice(english\_words)

while word['word'] in self.\_\_words:

word = random.choice(english\_words)

return word

def \_\_get\_wrong\_answers(self, right\_word):

wrong\_answers = dict()

wrong\_answers\_history = list()

for i in range(0, 3):

word = random.choice(english\_words)

while word['word'] in wrong\_answers\_history or word['word'] == right\_word:

word = random.choice(english\_words)

wrong\_answers[i] = word

wrong\_answers\_history.append(word['word'])

return wrong\_answers