Московский государственный технический университет им. Н.Э. Баумана

Факультет «Информатика и системы управления»
Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Парадигмы и конструкции языков программирования»

Отчет по лабораторной работе №6

Выполнил:
студент группы ИУ5-35Б
Бронникова М. Е.
Подпись и дата:

Проверил:
Подпись и дата:

Москва, 2024 г

Задание:
Разработать бота, который будет создавать папки с карточками, по которым можно устраивать викторину
Выполнение на python:

```
import telebot
from telebot.types import InlineKeyboardMarkup, ReplyKeyboardMarkup, KeyboardButton, InlineKeyboardButton
import random
import time

#message аргумент, который передается автоматически декоратором
#декоратор - это  функции, которые принимают другую функцию в качестве аргумента, добавляют
# к ней некоторую дополнительную функциональность и возвращают функцию с изменённым поведением

token = "7322283656:AAEMQT40skWKrNKATIR5mMLBiqsM1Bgj3zY"
bot = telebot.TeleBot(token)

folders = {}  # {folder_name: [("card_name", "translation"), ...]} папки и их карточки
quiz_settings_dict = {}  # {folder_name: {"format": "multiple_choice", "question_type": "card_name", "time_limit": False, "card_count": "all"}}  настройки викторины для каждой папки
current_quiz = {}  # {user_id: {"folder": folder_name, "current_card_index": 0, "correct_answers": 0, "incorrect_answers": 0}} информациz о текущей викторине каждого пользователя

# мэйнннн
def main_menu():
    markup = ReplyKeyboardMarkup(resize_keyboard=True)  # клава для меню, авто размер у польз
    markup.add(KeyboardButton("Help"), KeyboardButton("Создать папку"))
    markup.add(KeyboardButton("Посмотреть текущие папки"), KeyboardButton("Викторина"))
    return markup

@bot.message_handler(commands=['start']) # соо для старта декоратор, вызывает функцию от команды
def start(message):
    bot.send_message(message.chat.id, #айди чата
                     "Привет, я бот, который поможет тебе создавать карточки\n\n-"
                     "Нужно выучить перевод слов?\n-Какие-то определения или формулы?\n"
                     "-А может даже и билет на экзамен...\n\nТогда тебе сюда",
```

```
                reply_markup = main_menu()) #клава с командами

# help инструкция по использованию бота
@bot.message_handler(func=lambda message: message.text.lower() == "help")
#конкретный тип coo + перевод в нижний регистр ((ко всем добавить))
def help_message(message):
    bot.send_message(message.chat.id,
                "help - посмотреть команды\n"
                "создать папку - создание папки с будущими карточками\n"
                "посмотреть текущие папки - показывает ранее созданные
папки\n"
                "викторина - проверяет знание карточек уже существующей
папки",
                reply_markup=main_menu())

# Создание папки
@bot.message_handler(func=lambda message: message.text.lower() == "создать
папку") # func return T/F по обработке
def create_folder(message): #обрабатываемая функция
    msg = bot.send_message(message.chat.id, "Введите название новой
папки:")  # Запрашиваем название папки
    bot.register_next_step_handler(msg, process_folder_creation)  # Регистрируем
обработчик для следующего шага

def process_folder_creation(message): #naming
    folder_name = message.text.strip()  #coo + удаление пробелов по краям
    if folder_name:
        folders[folder_name] = []  # папка с пустым списком карточек

        print(f"Folder created: {folders}")  # отладка

        bot.send_message(message.chat.id, f"Папка '{folder_name}' создана.",
reply_markup = main_menu())
    else:
        bot.send_message(message.chat.id, "Имя папки не может быть пустым.",
reply_markup = main_menu())

# текущие папки
@bot.message_handler(func=lambda message: message.text.lower() ==
"посмотреть текущие папки")
def list_folders(message):
    if not folders:
        bot.send_message(message.chat.id, "Нет созданных папок.", reply_markup
= main_menu())  # если папко нет
    else:
```

```
    markup = InlineKeyboardMarkup()  # набор кнопышей
    for folder_name in folders:
        markup.add(InlineKeyboardButton(folder_name,
callback_data=f"open_folder_{folder_name}"))  # кнопка для каждомй пакпи
        #call_back вернет значение на которое прожмем и будут дальнейшие
действия \\форматирование строки по факту
    bot.send_message(message.chat.id, "Выберите папку:", reply_markup =
markup)

# карточки в папке
@bot.callback_query_handler(func=lambda call:
call.data.startswith("open_folder_")) #декоратор для инлайн кнпк
def open_folder(call):  #чек на прожимание
    folder_name = call.data.split("_")[2]  # имя папки из callback_data [open,
folder, name]
    markup = InlineKeyboardMarkup()
    markup.add(InlineKeyboardButton("Добавить карточку",
callback_data=f"add_card_{folder_name}"))  # добавление карточки
    markup.add(InlineKeyboardButton("Посмотреть карточки",
callback_data=f"view_cards_{folder_name}"))  # просмотр карточек
    bot.edit_message_text(chat_id=call.message.chat.id,
message_id=call.message.message_id,
                          text=f"Папка: {folder_name}", reply_markup = markup)  #
Обновляем сообщение с кнопками кт уже

# добавление карточки
@bot.callback_query_handler(func=lambda call: call.data.startswith("add_card_"))
def add_card_prompt(call):
    folder_name = call.data.split("_")[2] # [add, card, name]
    msg = bot.send_message(call.message.chat.id, "Введите имя карточки:")
    bot.register_next_step_handler(msg, add_card, folder_name)  # to be
continued.....

def add_card(message, folder_name):
    card_name = message.text  # наминг
    msg = bot.send_message(message.chat.id, "Введите перевод карточки:")
    bot.register_next_step_handler(msg, save_card, card_name, folder_name)  # to
be continued 2.0 .....

def save_card(message, card_name, folder_name):
    translation = message.text  # Translation of the card

    # Check if the folder exists, if not, create it
    if folder_name not in folders:
        folders[folder_name] = []  # Create the folder with an empty list of cards
```

```
    # Add the card to the folder
    folders[folder_name].append((card_name, translation))
    bot.send_message(message.chat.id, f"Карточка '{card_name}' с переводом
'{translation}' добавлена.", reply_markup=main_menu())

# cards in folder
@bot.callback_query_handler(func=lambda call:
call.data.startswith("view_cards_"))
def view_cards(call):
    folder_name = call.data.split("_")[2]  # name folder

    if folder_name in folders and folders[folder_name]:
        cards = "\n".join(f"{i+1}. {card[0]} - {card[1]}" for i, card in
enumerate(folders[folder_name]))  # список карточек
        #существвование доступность
        bot.send_message(call.message.chat.id, f"Карточки в папке
'{folder_name}':\n{cards}", reply_markup = main_menu())
    else:
        bot.send_message(call.message.chat.id, f"В папке '{folder_name}' пока нет
карточек.", reply_markup = main_menu())

# викторина  !!!!!
# было в планах
@bot.message_handler(func=lambda message: message.text.lower() ==
"настройки викторины")
def quiz_settings(message):
    # Настройка формата викторины для всех папок
    for folder_name in folders:
        quiz_settings_dict[folder_name] = {"format": "written", "question_type":
"card_name", "time_limit": False, "card_count": "all"}

    bot.send_message(message.chat.id, "Викторина настроена на формат 'written'
для всех папок.", reply_markup=main_menu())


# викторина йес
@bot.message_handler(func=lambda message: message.text.lower() ==
"викторина")
def quiz_menu(message):
    if not folders:
        bot.send_message(message.chat.id, "Нет доступных папок для
викторины.", reply_markup = main_menu())  # rasstrel
    else:
        markup = InlineKeyboardMarkup()
```

```
        for folder_name in folders:
            markup.add(InlineKeyboardButton(folder_name,
callback_data=f"start_quiz_{folder_name}"))  # choose your fighter
        bot.send_message(message.chat.id, "Выберите папку для викторины:",
reply_markup = markup)

# fight
@bot.callback_query_handler(func=lambda call:
call.data.startswith("start_quiz_"))
def start_quiz(call):
    folder_name = call.data.split("_")[2]  # [start, quiz, name]
    if folders.get(folder_name): #try to значение по ключу get
        current_quiz[call.from_user.id] = { #была заготовка под настройки (())
            "folder": folder_name,
            "current_cards": random.sample(folders[folder_name],
len(folders[folder_name])),  # ees
            "score": []
        }
        bot.send_message(call.message.chat.id, f"Начинаем викторину по папке
'{folder_name}'.", reply_markup = main_menu())
        ask_question(call.message.chat.id, call.from_user.id)  #func fight
    else:
        bot.send_message(call.message.chat.id, f"Папка '{folder_name}' не
найдена.", reply_markup = main_menu())

# question
def ask_question(chat_id, user_id):
    quiz_data = current_quiz.get(user_id)
    if quiz_data and quiz_data["current_cards"]: #check остатки викторины
        current_card = quiz_data["current_cards"].pop(0)  # извлечение карточки
        quiz_data["current_card"] = current_card #заготовка на сложные штуки
заливаеи тек значение в словарь ключ кар кард
        bot.send_message(chat_id, f"Как переводится '{current_card[0]}'?",
reply_markup = main_menu())
        bot.register_next_step_handler_by_chat_id(chat_id, check_answer,
current_card[1], user_id)  # все сверяем
    else: #конец викторины
        show_results(chat_id, user_id)

# check
def check_answer(message, correct_answer, user_id):
    quiz_data = current_quiz.get(user_id) #information по викторине
пользователя
    if quiz_data:
        user_answer = message.text.strip() #answer == trans?
```

```
        if user_answer.lower() == correct_answer.lower():
            quiz_data["score"].append(True)
            bot.send_message(message.chat.id, "Правильно!")
        else:
            quiz_data["score"].append(False)
            bot.send_message(message.chat.id, f"Неправильно. Правильный ответ:
{correct_answer}")
        time.sleep(1)
        ask_question(message.chat.id, user_id)  # Спрашиваем следующий вопрос

# Показ результатов викторины
def show_results(chat_id, user_id):
    quiz_data = current_quiz.get(user_id)
    if quiz_data:
        correct_answers = sum(quiz_data["score"])
        total_questions = len(quiz_data["score"])
        bot.send_message(chat_id, f"Викторина завершена!\nПравильных ответов:
{correct_answers}/{total_questions}")
        del current_quiz[user_id]  # free memory
    else:
        bot.send_message(chat_id, "Викторина не была начата.",
reply_markup=main_menu())

# Запуск бота
bot.polling()
```

Результаты:


21:08
flashcards
бот
/start 10:50
Привет, я бот, который поможет тебе создавать карточки
-Нужно выучить перевод слов?
-Какие-то определения или формулы?
-А может даже и билет на экзамен...
Тогда тебе сюда 10:50
Help 10:50
help - посмотреть команды
создать папку - создание папки с будущими карточками
посмотреть текущие папки - показывает ранее созданные папки
викторина - проверяет знание карточек уже существующей папки 10:50
Создать папку 10:50
Введите название новой папки: 10:50
jgug 10:50
Папка 'jgug' создана. 10:50
Посмотреть текущие папки 10:50
Сообщение
Help
Создать папку
Посмотреть текущие папки
Викторина


21:08
flashcards
бот
Посмотреть текущие папки 10:50
Папка: jgug 10:50
Добавить карточку
Посмотреть карточки
В папке 'jgug' пока нет карточек. 10:50
Введите имя карточки: 10:50
jvjg 10:53
Введите перевод карточки: 10:53
fgjk 10:53
Карточка 'jvjg' с переводом 'fgjk' добавлена. 10:53
Карточки в папке 'jgug':
1. jvjg - fgjk 10:53
Введите имя карточки: 10:54
fugiyogo 10:54
Введите перевод карточки: 10:54
Сообщение
Help
Создать папку
Посмотреть текущие папки
Викторина


21:08
flashcards
бот
fgjk 10:54
Карточка 'fugiyogo' с переводом 'fgjk' добавлена. 10:54
Введите имя карточки: 10:54
1 10:54
Введите перевод карточки: 10:54
0 10:54
Карточка '1' с переводом '0' добавлена. 10:54
Посмотреть текущие папки 10:54
Папка: jgug 10:54
Добавить карточку
Посмотреть карточки
Карточки в папке 'jgug':
1. jvjg - fgjk
2. fugiyogo - fgjk
3. 1 - 0 10:54
Сообщение
Help
Создать папку
Посмотреть текущие папки
Викторина


21:08
flashcards
бот
Викторина 10:54
Выберите папку для викторины: 10:54
jgug
Начинаем викторину по папке 'jgug'. 10:54
Как переводится '1'? 10:54
0 10:55
Правильно! 10:55
Как переводится 'jvjg'? 10:55
лалв 10:55
Неправильно. Правильный ответ: fgjk 10:55
Как переводится 'fugiyogo'? 10:55
оалвл 10:55
Неправильно. Правильный ответ: fgjk 10:55
Викторина завершена!
Сообщение
Help
Создать папку
Посмотреть текущие папки
Викторина