**Московский государственный технический**

**университет им. Н.Э. Баумана**

Факультет «Радиотехнический»

Кафедра «Системы обработки информации и управления»

Курс «Парадигмы и конструкции языков программирования »

Домашнее задание

|  |  |
| --- | --- |
| Выполнил: |  |
| студент группы  РТ5-31Б |  |
| Данилова А.С. |  |
|  |  |

Москва, 2024 г

**Постановка задачи:**

Разработать телеграм-бот, в котором можно будет пройти тесты по математике, русскому языку и английскому языку. В конце бот будет выводить результат тестирования.

**Текст программы:**

import telebot  
from telebot import types  
  
API\_TOKEN = '7820479723:AAEuI6wgtNRx72h2-kfeIlnXJ21sRYYc5I4'  
bot = telebot.TeleBot(API\_TOKEN)  
  
# Состояния  
START, SUBJECT, MATH, RUSSIAN, ENGLISH, ENGLISH\_TOPIC, QUIZ, RESULT = range(8)  
  
# Вопросы  
math\_questions = [  
 ("2 + 2 =", "4"),  
 ("3 + 5 =", "8"),  
 ("10 + 15 =", "25"),  
 ("7 + 3 =", "10"),  
 ("5 + 9 =", "14"),  
 ("2 \* 2 =", "4"),  
 ("3 \* 5 =", "15"),  
 ("10 \* 15 =", "150"),  
 ("7 \* 3 =", "21"),  
 ("5 \* 9 =", "45"),  
]  
  
russian\_questions = [  
 ("Укажите предложения, в которых нужно поставить ОДНУ запятую. Запишите номера этих предложений.\n1)Плоды этого растения полезные и вкусные и обладают прекрасным ароматом.\n2)Стало нестерпимо душно и пришлось открыть все окна.\n3)Из окна были видны стволы вишен да кусочек аллеи.\n4)Изучение роста необычных кристаллов имеет и теоретическое и практическое и общенаучное значение.\n5)Древние испанские мастера при строительстве замков применяли либо каменную либо кирпичную кладку.", "25"),  
 ("Укажите предложения, в которых нужно поставить ОДНУ запятую. Запишите номера этих предложений.\n1)Для развития личности полезны и искусство и наука и жизненный опыт.\n2)Грубое или даже нелюбезное поведение людей может испортить настроение окружающим.\n3)Память накапливает добрый опыт и традиции и постоянно противостоит уничтожающей силе времени.\n4)Хорошие манеры и правильно выработанное поведение принесут человеку как хорошее настроение так и уважение окружающих.\n5)Белый теплоход уверенно рассекал невысокие волны и только по лёгкому дрожанию корпуса пассажиры догадывались о начале морской качки.", "45"),  
 ("Укажите предложения, в которых нужно поставить ОДНУ запятую. Запишите номера этих предложений.\n1)  Прогулка или разговор с другом были одинаково приятны для меня.\n2)Сумрак скрыл очертания лица и фигуры Ольги и набросил на неё как будто покрывало.\n3)На земле жилось нелегко и поэтому я очень полюбил бездонное небо.\n4)Ученье да труд всё перетрут.\n5)В колледже он с увлечением занимался как гуманитарными так и естественно-математическими дисциплинами.'?", "35"),  
 ("Укажите цифру(-ы), на месте которой(-ых) должна(-ы) стоять запятая(-ые).\nИздали (1) он увидел дом (2) непохожий на другие (3) построенный (4) каким-то итальянским архитектором.", "23"),  
 ("Укажите цифру(-ы), на месте которой(-ых) должна(-ы) стоять запятая(-ые).\nНад ещё не улёгшимся (1) после недавней бури (2) бескрайним морем (3) возвышалось небо (4) унизанное (5) ярко мерцавшими звёздами.", "4"),  
 ("Укажите цифру(-ы), на месте которой(-ых) должна(-ы) стоять запятая(-ые).\nГород (1) вдали сверкающий на солнце (2) синие леса (3) окаймляющие берега залива (4) казались мне особенно торжественными.", "1234"),  
 ("Укажите цифру(-ы), на месте которой(-ых) должна(-ы) стоять запятая(-ые).\nБольшой пруд (1) густо заросший кувшинками (2) располагался (3) в удалённой от дома (4) части старого парка.", "12"),  
 ("Укажите цифру(-ы), на месте которой(-ых) должна(-ы) стоять запятая(-ые).\nВладимир (1) махавший косой не переставая (2) резал траву (3) не выказывая (4) ни малейшего усилия.", "123"),  
 ("Укажите цифру(-ы), на месте которой(-ых) должна(-ы) стоять запятая(-ые).\nТуча (1) нависшая (2) над высокими вершинами тополей (3) уже сыпала (4) моросящим дождиком.", "13"),  
 ("Укажите цифру(-ы), на месте которой(-ых) должна(-ы) стоять запятая(-ые).\nПробравшись (1) через мокрый папоротник и какую-то (2) стелющуюся растительность (3) выбираемся на едва приметную тропинку.", "3"),  
]  
  
english\_questions = {  
 "fruits": [  
 ("Яблоко", ["Apple", "Banana", "Cherry", "Grape"]),  
 ("Банан", ["Banana", "Orange", "Peach", "Mango"]),  
 ("Виноград", ["Grape", "Apple", "Lemon", "Kiwi"]),  
 ("Апельсин", ["Orange", "Peach", "Plum", "Berry"]),  
 ("Персик", ["Peach", "Apple", "Banana", "Grape"]),  
 ],  
 "clothes": [  
 ("Рубашка", ["Shirt", "Pants", "Hat", "Shoes"]),  
 ("Штаны", ["Pants", "Shirt", "Belt", "Socks"]),  
 ("Шляпа", ["Hat", "Scarf", "Gloves", "Coat"]),  
 ("Обувь", ["Shoes", "Boots", "Sandals", "Slippers"]),  
 ("Платье", ["Dress", "Skirt", "Blouse", "Jacket"]),  
 ],  
 "animals": [  
 ("Собака", ["Dog", "Cat", "Mouse", "Bird"]),  
 ("Кошка", ["Cat", "Dog", "Fish", "Rabbit"]),  
 ("Птица", ["Bird", "Insect", "Mammal", "Reptile"]),  
 ("Лошадь", ["Horse", "Donkey", "Mule", "Zebra"]),  
 ("Корова", ["Cow", "Goat", "Sheep", "Pig"]),  
 ],  
}  
  
user\_data = {}  
  
@bot.message\_handler(commands=['start'])  
def start\_quiz(message):  
 user\_data[message.chat.id] = {'state': START, 'score': 0, 'question\_index': 0}  
 bot.send\_message(message.chat.id, "Добро пожаловать! Выберите предмет: /math, /russian, /english")  
  
@bot.message\_handler(commands=['math'])  
def math\_quiz(message):  
 user\_data[message.chat.id]['state'] = MATH  
 user\_data[message.chat.id]['score'] = 0  
 user\_data[message.chat.id]['question\_index'] = 0  
 ask\_math\_question(message)  
  
@bot.message\_handler(commands=['russian'])  
def russian\_quiz(message):  
 user\_data[message.chat.id]['state'] = RUSSIAN  
 user\_data[message.chat.id]['score'] = 0  
 user\_data[message.chat.id]['question\_index'] = 0  
 ask\_russian\_question(message)  
  
@bot.message\_handler(commands=['english'])  
def english\_quiz(message):  
 user\_data[message.chat.id]['state'] = ENGLISH  
 bot.send\_message(message.chat.id, "Выберите тему: /fruits, /clothes, /animals")  
  
@bot.message\_handler(commands=['fruits', 'clothes', 'animals'])  
def english\_topic(message):  
 topic = message.text[1:] # Убираем '/'  
 user\_data[message.chat.id]['state'] = ENGLISH\_TOPIC  
 user\_data[message.chat.id]['score'] = 0  
 user\_data[message.chat.id]['question\_index'] = 0  
 user\_data[message.chat.id]['topic'] = topic  
 ask\_english\_question(message)  
  
def ask\_math\_question(message):  
 user\_id = message.chat.id  
 index = user\_data[user\_id]['question\_index']  
 if index < len(math\_questions):  
 question, correct\_answer = math\_questions[index]  
 bot.send\_message(user\_id, question)  
 user\_data[user\_id]['correct\_answer'] = correct\_answer  
 else:  
 show\_result(message)  
  
def ask\_russian\_question(message):  
 user\_id = message.chat.id  
 index = user\_data[user\_id]['question\_index']  
 if index < len(russian\_questions):  
 question, correct\_answer = russian\_questions[index]  
 bot.send\_message(user\_id, question)  
 user\_data[user\_id]['correct\_answer'] = correct\_answer  
 else:  
 show\_result(message)  
  
def ask\_english\_question(message):  
 user\_id = message.chat.id  
 index = user\_data[user\_id]['question\_index']  
 topic = user\_data[user\_id]['topic']  
 if index < len(english\_questions[topic]):  
 question, options = english\_questions[topic][index]  
 correct\_answer = options[0] # Первый вариант - правильный  
 markup = types.ReplyKeyboardMarkup(one\_time\_keyboard=True)  
 for option in options:  
 markup.add(option)  
 bot.send\_message(user\_id, question, reply\_markup=markup)  
 user\_data[user\_id]['correct\_answer'] = correct\_answer  
 else:  
 show\_result(message)  
  
@bot.message\_handler(func=lambda message: user\_data.get(message.chat.id, {}).get('state') in [MATH, RUSSIAN, ENGLISH, ENGLISH\_TOPIC])  
def handle\_answer(message):  
 user\_id = message.chat.id  
 correct\_answer = user\_data[user\_id]['correct\_answer']  
 if message.text.lower() == correct\_answer.lower():  
 user\_data[user\_id]['score'] += 1  
 user\_data[user\_id]['question\_index'] += 1  
  
 if user\_data[user\_id]['state'] == MATH:  
 ask\_math\_question(message)  
 elif user\_data[user\_id]['state'] == RUSSIAN:  
 ask\_russian\_question(message)  
 elif user\_data[user\_id]['state'] in [ENGLISH, ENGLISH\_TOPIC]:  
 ask\_english\_question(message)  
  
def show\_result(message):  
 user\_id = message.chat.id  
 score = user\_data[user\_id]['score']  
 total\_questions = len(math\_questions) if user\_data[user\_id]['state'] == MATH else len(russian\_questions) if user\_data[user\_id]['state'] == RUSSIAN else len(english\_questions[user\_data[user\_id]['topic']])  
 bot.send\_message(user\_id, f"Твой результат: {score} из {total\_questions}!")  
 user\_data[user\_id]['state'] = START  
 user\_data[user\_id]['state'] = RESULT  
 bot.send\_message(user\_id, "Выберите предмет: /math, /russian, /english или /stop для завершения.")  
  
@bot.message\_handler(commands=['stop'])  
def stop\_quiz(message):  
 bot.send\_message(message.chat.id, "Спасибо за участие! До свидания!")  
  
  
bot.polling()

**Экранные формы с примерами выполнения программы**





