

**Московский государственный технический  
университет им. Н.Э. Баумана**

**Факультет «Радиотехнический»  
Кафедра «Системы обработки информации и управления»**

**Курс «Парадигмы и конструкции языков программирования »**

**Лабораторная работа №5-6**

**Выполнил:**

**студент группы  
РТ5-31Б**

**Данилова А.С.**

**Москва, 2024 г**

## Постановка задачи

В данной работе объединено две лабораторных работы. Задачи каждой лабораторной:

1. Разработайте простого бота для Telegram. Бот должен использовать функциональность создания кнопок.
2. Разработайте бота для Telegram. Бот должен реализовывать конечный автомат из трех состояний.

## Текст программы

```
import telebot
from telebot import types

API_TOKEN = '7716906729:AAHzMuP4y3vFlElmfCn9LX_zG5UUUedovZc'
bot = telebot.TeleBot(API_TOKEN)

# Состояния
START, QUIZ, RESULT = range(3)

# Вопросы и ответы
questions = [
    ("Какое самое крупное наземное животное?", "Слон"),
    ("Какое животное известно как король джунглей?", "Лев"),
    ("Какое самое быстрое наземное животное?", "Гепард"),
    ("Какое животное известно своей способностью менять цвет?", "Хамелеон"),
    ("Какое самое крупное млекопитающее в океане?", "Кит"),
    ("Какое животное известно своей мудростью?", "Сова"),
    ("Какое единственное млекопитающее способно по-настоящему летать?",
     "Летучая мышь"),
    ("Какое животное известно своей длинной шеей?", "Жираф"),
    ("Какой самый крупный вид медведей?", "Полярный медведь"),
    ("Какое животное известно своими полосами?", "Тигр"),
]

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 на викторину о животных! Type /quiz to start.")

@bot.message_handler(commands=['quiz'])
def quiz(message):
    user_data[message.chat.id]['state'] = QUIZ
    ask_question(message)

def ask_question(message):
    user_id = message.chat.id
    index = user_data[user_id]['question_index']
    if index < len(questions):
        question, _ = questions[index]
        markup = types.ReplyKeyboardMarkup(one_time_keyboard=True)
        markup.add('Next')
        bot.send_message(user_id, question, reply_markup=markup)
    else:
```

```

        show_result(message)

@bot.message_handler(func=lambda message: user_data.get(message.chat.id,
{}).get('state') == QUIZ)
def handle_answer(message):
    user_id = message.chat.id
    index = user_data[user_id]['question_index']
    if index < len(questions):
        _, correct_answer = questions[index]
        if message.text.lower() == correct_answer.lower():
            user_data[user_id]['score'] += 1
            user_data[user_id]['question_index'] += 1
            ask_question(message)

def show_result(message):
    user_id = message.chat.id
    score = user_data[user_id]['score']
    bot.send_message(user_id, f"Твой результат {score} из {len(questions)}!
Type /start to play again.")
    user_data[user_id]['state'] = RESULT

bot.polling()

```

## Экранные формы с примерами выполнения программы



