МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

### федеральное государственное автономное образовательное учреждение высшего образования

«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ»

(ГУАП)

КАФЕДРА ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ И СЕТЕЙ

КУРСОВАЯ РАБОТА

ЗАЩИЩЕНА С ОЦЕНКОЙ

РУКОВОДИТЕЛЬ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| доцент, канд. техн. наук |  |  |  | Л.Н. Бариков |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

|  |
| --- |
| ПОЯСНИТЕЛЬНАЯ ЗАПИСКА К КУРСОВОЙ РАБОТЕ |
| ПРОГРАММА **Бот для мессенджера «Telegram»** ОП 44.4243.23 ПЗ |
| по дисциплине: Основы программирования |
|  |
|  |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ ГР. | 4243 |  |  |  | В.С. Суковатицин |
|  |  |  | подпись, дата |  | инициалы, фамилия |

Санкт-Петербург

2023

**СОДЕРЖАНИЕ**

ВВЕДЕНИЕ..…………………………………………………………………………………3

1 Постановка задачи………………………………………………………………….4

2 Спецификация………………………………………………………………………6

3 Текст программы………………………………………………………………….10

3.1 Текст программы……………………………………………………….10

3.2 Результат тестирования……………………………………………….16

4 Описание программы…………………………………………………………….25

4.1 Общие сведения………………………………………………………..27

4.2 Функциональное назначение………………………………………….27

4.3 Описание логической структуры……………………………………….27

4.4 Спецификация функции……………………………………………….27

4.5 Используемые технические средства…………………………………30

4.6 Вызов и загрузка……………………………………………………….30

5 Описание применения…………………………………………………………….32

5.1 Назначение программы……………………………………………..34

5.2 Условие применения…………………………………………………..34

5.3 Описание задачи…………………………………………………………34

5.4 Характеристики занимаемой памяти………………………………….34

ЗАКЛЮЧЕНИЕ……………………………………………………………………………..35

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ……………………………………...36

**ВВЕДЕНИЕ**

Данная работа предназначена для закрепления учебного материала, изученного по курсу «Основы программирования».

Программа представляет собой реализованного бота для мессенджера «Telegram» с подключённой к нему базой данных. Управление ботом с базой данных простое, интерфейс интуитивно понятен.

Цель курсовой работы – выработать навыки использования алгоритмов для решения конкретных задач, углубления знаний по программированию на языке Python.

Кроме этого, курсовая работа предназначена для приобретения навыков оформления сопроводительной документации к разработанному программному средству.

**1 Постановка задачи**

Используя технологию структурного программирования, разработать программу бот для мессенджера «Telegram».

Программа должна иметь удобный интерфейс, быстро работать.

Бот предназначен для упрощения хранения шуток и анекдотов в текстовом или звуковом формате и для развлечения пользователя.

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

### федеральное государственное автономное образовательное учреждение высшего образования

«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ»

(ГУАП)

КАФЕДРА ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ И СЕТЕЙ

|  |
| --- |
| ПРОГРАММА |
| Бот для мессенджера «Telegram» |
| Спецификация |
| 44.4243.23-01 |
|  |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ ГР. | 4243 |  |  |  | В.С. Суковатицин |
|  |  |  | подпись, дата |  | инициалы, фамилия |

Санкт-Петербург

2023

**2.1 Спецификация**

|  |  |  |
| --- | --- | --- |
| Обозначение | Наименование | Примечание |
| 44.4243.23-12 | Документация  Программа  Бот для мессенджера «Telegram»  Текст программы |  |
| 44.4243.23-13 | Документация  Программа  Бот для мессенджера «Telegram»  Описание программы |  |
| 44.4243.23-31 | Документация  Программа  Бот для мессенджера «Telegram»  Описание применения |  |

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

### федеральное государственное автономное образовательное учреждение высшего образования

«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ»

(ГУАП)

КАФЕДРА ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ И СЕТЕЙ

|  |
| --- |
| ПРОГРАММА |
| Бот для мессенджера «Telegram» |
| Текст программы |
| 44.4243.23-12 |
|  |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ ГР. | 4243 |  |  |  | В.С. Суковатицин |
|  |  |  | подпись, дата |  | инициалы, фамилия |

Санкт-Петербург

2023

Аннотация

В документе приводится текст программы «бот для мессенджера «Telegram» », написанной на языке Python, а так же результат тестирования.

**СОДЕРЖАНИЕ**

3.1 Текст программы

3.2 Результат тестирования

**3.1 Текст программы**

from telebot import types  
import telebot  
import sqlite3  
name = None  
current\_category = None  
token = 'токен не стоит выкладывать на гитхаб'  
bot=telebot.TeleBot(token)  
password = "let me in"  
@bot.message\_handler(commands=['start'])  
def greetings(message):  
  
 conn = sqlite3.connect('anecdotes.sql')  
 cur = conn.cursor()  
  
 cur.execute('CREATE TABLE IF NOT EXISTS anecdotes (id int auto\_increment primary key, category varchar, jokename varchar , joke varchar)')  
 conn.commit()  
 cur.close()  
 conn.close()  
  
 markup= types.ReplyKeyboardMarkup(resize\_keyboard=True)  
 item1 = types.KeyboardButton("1- добавить шутку ➕")  
 item2 = types.KeyboardButton("2- удалить шутку ❌" )  
 item3 = types.KeyboardButton("3- вывести случайную шутку 🎲")  
 item4 = types.KeyboardButton("4- просмотреть все шутки 👁")  
 item5 = types.KeyboardButton("5- прослушать озвученные шутки 🗣")  
 markup.add(item1,item2,item3,item4,item5)  
 bot.send\_message(message.chat.id, "Привет, я бот, который хранит и рассказывает шутки ",reply\_markup=markup)  
@bot.message\_handler(commands=['randomjoke'])  
def randomjoke(message):  
 conn = sqlite3.connect('anecdotes.sql')  
 cur = conn.cursor()  
  
 cur.execute('SELECT \* FROM anecdotes ORDER BY RANDOM() LIMIT 1')  
 jokenames = cur.fetchall()  
 info = ''  
 for el in jokenames:  
 info += f'Категория {el[1]}, заголовок: {el[2]}, шутка: {el[3]}\n'  
 cur.close()  
 conn.close()  
 bot.send\_message(message.chat.id, "случайная шутка:")  
 bot.send\_message(message.chat.id, info)  
  
@bot.message\_handler(commands=['help'])  
def help(message):  
 bot.send\_message(message.chat.id,"для того чтобы перезапустить меню пропишите в чате команду /start")  
 bot.send\_message(message.chat.id,"через кнопки меню в чате вы можете добавлять или удалять шутки,"  
 " но для доступа к этой функции вам нужно ввести пароль"  
 "\n ещё вы можете без пароля просмотреть все уже имеющиеся в базе шутки с помощью"  
 " команды /show\_jokes или кнопки 4- просмотреть все шутки 👁,\n "  
 "также вы можете прослушать уже озвученные, нажав на кнопку 5- прослушать озвученные шутки 🗣")  
@bot.message\_handler(commands=["show\_jokes"])  
def show\_jokes(message):  
 conn = sqlite3.connect('anecdotes.sql')  
 cur = conn.cursor()  
  
 cur.execute('SELECT \* FROM anecdotes')  
 jokes = cur.fetchall()  
  
 info = ''  
 category=''  
 viewed\_categories=[]  
 for el in jokes:  
 category=el[1]  
 if category not in viewed\_categories:  
 info=''  
 cur.execute(f"SELECT \* FROM anecdotes WHERE category='{category}';")  
 jokes\_category=cur.fetchall()  
 for elem in jokes\_category:  
 info += f'заголовок: {elem[2]}\n шутка: {elem[3]}\n'  
 markup = telebot.types.InlineKeyboardMarkup()  
 markup.add(telebot.types.InlineKeyboardButton(f'категория: {el[1]}', callback\_data='category.anecdotes'))  
 bot.send\_message(message.chat.id, info, reply\_markup=markup)  
 viewed\_categories.append(category)  
  
 cur.close()  
 conn.close()  
  
@bot.message\_handler(content\_types=["text"])  
def bot\_message(message):  
 if message.chat.type == 'private':  
 if message.text == "1- добавить шутку ➕":  
 bot.send\_message(message.chat.id,"введите пароль")  
 bot.register\_next\_step\_handler(message,password\_check\_add\_joke)  
 elif message.text == "2- удалить шутку ❌":  
 bot.send\_message(message.chat.id, "введите пароль")  
 bot.register\_next\_step\_handler(message, password\_check\_delete\_joke)  
 elif message.text == "3- вывести случайную шутку 🎲":  
 markup = types.ReplyKeyboardMarkup(resize\_keyboard=True)  
 back = types.KeyboardButton("назад 🔙")  
 item1 = types.KeyboardButton("ещё одна случайная шутка ")  
 markup.add(item1,back)  
 conn = sqlite3.connect('anecdotes.sql')  
 cur = conn.cursor()  
  
 cur.execute('SELECT \* FROM anecdotes ORDER BY RANDOM() LIMIT 1')  
 jokenames = cur.fetchall()  
 info = ''  
 for el in jokenames:  
 info += f'Категория {el[1]}, название: {el[2]}, сама шутка: {el[3]}\n'  
 cur.close()  
 conn.close()  
 bot.send\_message(message.chat.id,"случайная шутка:",reply\_markup=markup)  
 bot.send\_message(message.chat.id, info)  
  
 elif message.text== "4- просмотреть все шутки 👁":  
 markup = telebot.types.InlineKeyboardMarkup()  
 markup.add(telebot.types.InlineKeyboardButton('список шуток', callback\_data='anecdotes'))  
 bot.send\_message(message.chat.id, "шутки", reply\_markup=markup)  
  
 elif message.text=="5- прослушать озвученные шутки 🗣":  
  
 markup = types.ReplyKeyboardMarkup(resize\_keyboard=True)  
 item1=types.KeyboardButton("🗣анекдот про зайца и медведя🐻🐇")  
 item2=types.KeyboardButton("🗣заколдовал колдун мужика")  
 item4 = types.KeyboardButton("🗣анекдот про ушлого студента")  
 item5 = types.KeyboardButton("🗣генерал-захотел-сапоги-из-крокодила")  
 item6 = types.KeyboardButton("🗣что будете делать если вам миллион дадут?")  
 back = types.KeyboardButton("назад 🔙")  
 markup.add(item1,item2,item4,item5,item6,back)  
 bot.send\_message(message.chat.id, "список озвученных шуток:", reply\_markup=markup)  
 elif message.text == "🗣анекдот про зайца и медведя🐻🐇":  
 voice = open("C:\\Users\\sukov\\OneDrive\\Desktop\\лабы и практические\\3сем\\курсач\\pjoect\\test.ogg",  
 'rb')  
 bot.send\_voice(message.chat.id, voice)  
 voice.close()  
 elif message.text =="🗣заколдовал колдун мужика":  
 voice = open("C:\\Users\\sukov\\OneDrive\\Desktop\\лабы и практические\\3сем\\курсач\\pjoect\\koldun.ogg",  
 'rb')  
 bot.send\_voice(message.chat.id, voice)  
 voice.close()  
 elif message.text =="🗣анекдот про ушлого студента":  
 voice = open("C:\\Users\\sukov\\OneDrive\\Desktop\\лабы и практические\\3сем\\курсач\\pjoect\\анекдот про ушлого студента.ogg",  
 'rb')  
 bot.send\_voice(message.chat.id, voice)  
 voice.close()  
 elif message.text =="🗣генерал-захотел-сапоги-из-крокодила":  
 voice = open("C:\\Users\\sukov\\OneDrive\\Desktop\\лабы и практические\\3сем\\курсач\\pjoect\\генерал-захотел-сапоги-из-крокодила.ogg",  
 'rb')  
 bot.send\_voice(message.chat.id, voice)  
 voice.close()  
 elif message.text =="🗣что будете делать если вам миллион дадут?":  
 voice = open("C:\\Users\\sukov\\OneDrive\\Desktop\\лабы и практические\\3сем\\курсач\\pjoect\\если-вам-миллион-дадут.ogg",  
 'rb')  
 bot.send\_voice(message.chat.id, voice)  
 voice.close()  
 elif message.text=="ещё одна случайная шутка":  
  
 conn = sqlite3.connect('anecdotes.sql')  
 cur = conn.cursor()  
  
 cur.execute('SELECT \* FROM anecdotes ORDER BY RANDOM() LIMIT 1')  
 jokenames = cur.fetchall()  
 info = ''  
 for el in jokenames:  
 info += f'Категория {el[1]}, название: {el[2]}, сама шутка: {el[3]}\n'  
 cur.close()  
 conn.close()  
 bot.send\_message(message.chat.id, info)  
  
 markup = types.ReplyKeyboardMarkup(resize\_keyboard=True)  
 back = types.KeyboardButton("назад 🔙")  
 item1 = types.KeyboardButton("ещё одна случайная шутка ")  
 markup.add(item1,back)  
  
 elif message.text == "назад 🔙":  
 markup = types.ReplyKeyboardMarkup(resize\_keyboard=True)  
 item1 = types.KeyboardButton("1- добавить шутку ➕")  
 item2 = types.KeyboardButton("2- удалить шутку ❌")  
 item3 = types.KeyboardButton("3- вывести случайную шутку 🎲")  
 item4 = types.KeyboardButton("4- просмотреть все шутки 👁")  
 item5 = types.KeyboardButton("5- прослушать озвученные шутки 🗣")  
 markup.add(item1, item2, item3, item4, item5)  
 bot.send\_message(message.chat.id, "назад 🔙",reply\_markup=markup)  
  
password\_checker = False  
  
def password\_check\_add\_joke(message):  
 global password\_checker  
 pswrd = message.text.strip()  
 if pswrd == password:  
 password\_checker = True  
 bot.send\_message(message.chat.id, "пароль верный, добро пожаловать")  
 bot.send\_message(message.chat.id, "Введите категорию шутки")  
 bot.register\_next\_step\_handler(message, joke\_category)  
 else:  
 bot.send\_message(message.chat.id, "пароль неверный.")  
  
def joke\_category(message):  
 global current\_category  
 current\_category = message.text.strip()  
 bot.send\_message(message.chat.id, "введите название шутки")  
 bot.register\_next\_step\_handler(message,joke\_name)  
  
def joke\_name(message):  
 global name  
 name = message.text.strip()  
 bot.send\_message(message.chat.id, "отправьте шутку одним текстовым сообщением")  
 bot.register\_next\_step\_handler(message, joke\_text)  
  
def joke\_text(message):  
 joke = message.text.strip()  
  
 conn = sqlite3.connect('anecdotes.sql')  
 cur = conn.cursor()  
 sql='INSERT INTO anecdotes (category, jokename, joke) VALUES (%s, %s, %s)'  
 cur.execute("INSERT INTO anecdotes VALUES(?,?,?,?)",(None,current\_category,name,joke))  
 conn.commit()  
 cur.close()  
 conn.close()  
  
 markup = telebot.types.InlineKeyboardMarkup()  
 markup.add(telebot.types.InlineKeyboardButton('список шуток', callback\_data='anecdotes'))  
 bot.send\_message(message.chat.id, "ваша шутка успешно добавлена!",reply\_markup=markup)  
  
def password\_check\_delete\_joke(message):  
 global password\_checker  
 pswrd = message.text.strip()  
 if pswrd == password:  
 password\_checker = True  
 bot.send\_message(message.chat.id, "пароль верный, добро пожаловать")  
 bot.send\_message(message.chat.id, "Введите заголовок шутки, которую хотите удалить")  
 bot.register\_next\_step\_handler(message, delete\_joke)  
 else:  
 bot.send\_message(message.chat.id, "пароль неверный.")  
  
def delete\_joke(message):  
 deleting\_name = message.text.strip()  
  
 conn = sqlite3.connect('anecdotes.sql')  
 cur = conn.cursor()  
  
 cur.execute(f"DELETE FROM anecdotes WHERE jokename='{deleting\_name}';")  
 conn.commit()  
 cur.close()  
 conn.close()  
  
 markup = telebot.types.InlineKeyboardMarkup()  
 markup.add(telebot.types.InlineKeyboardButton('список шуток', callback\_data='anecdotes'))  
 bot.send\_message(message.chat.id, f"Шутка с заголовком '{deleting\_name}' удалена", reply\_markup=markup)  
 markup = types.ReplyKeyboardMarkup(resize\_keyboard=True)  
 back = types.KeyboardButton("назад 🔙")  
 markup.add(back)  
  
  
@bot.callback\_query\_handler(func= lambda call: call.data.startswith("anecdotes"))  
def callback(call):  
  
 conn = sqlite3.connect('anecdotes.sql')  
 cur = conn.cursor()  
  
 cur.execute('SELECT \* FROM anecdotes')  
 jokes = cur.fetchall()  
  
 info = ''  
 category=''  
 viewed\_categories=[]  
 for el in jokes:  
 category=el[1]  
 if category not in viewed\_categories:  
 info=''  
 cur.execute(f"SELECT \* FROM anecdotes WHERE category='{category}';")  
 jokes\_category=cur.fetchall()  
 for elem in jokes\_category:  
 info += f'заголовок: {elem[2]}\n шутка: {elem[3]}\n'  
 markup = telebot.types.InlineKeyboardMarkup()  
 markup.add(telebot.types.InlineKeyboardButton(f'категория: {el[1]}', callback\_data='category.anecdotes'))  
 bot.send\_message(call.message.chat.id, info, reply\_markup=markup)  
 viewed\_categories.append(category)  
  
 cur.close()  
 conn.close()  
  
if \_\_name\_\_=='\_\_main\_\_':  
 bot.infinity\_polling()

**3.2 Результаты тестирования**

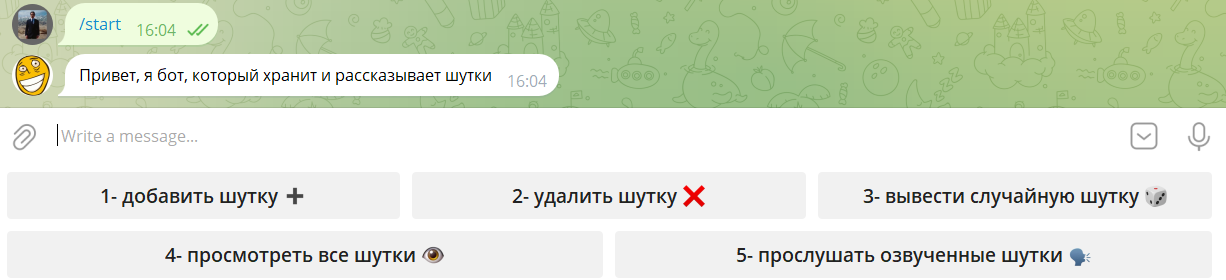
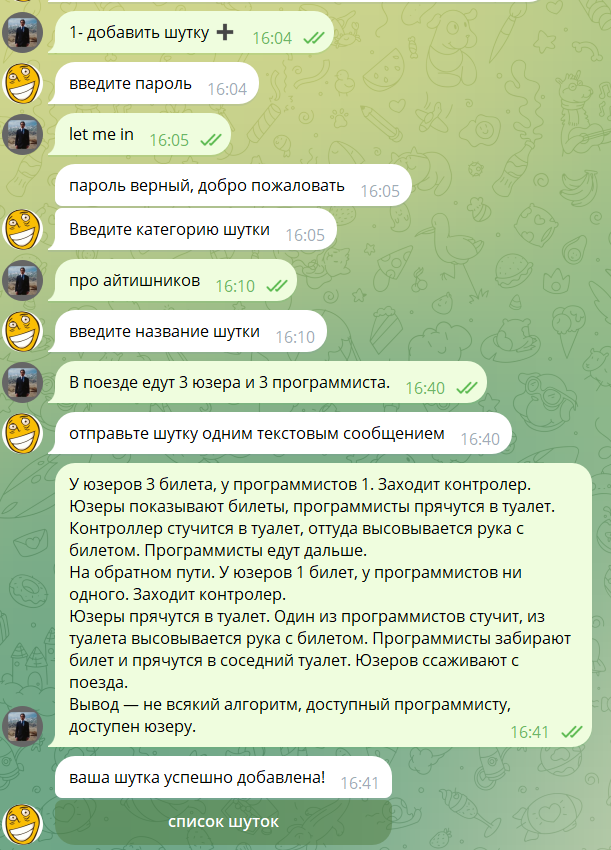
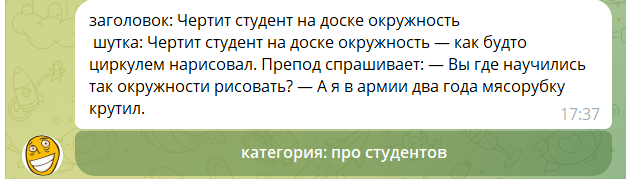


Рисунок 1 – запуск бота

  
Рисунок 2 – добавление шутки с помощью кнопки 1



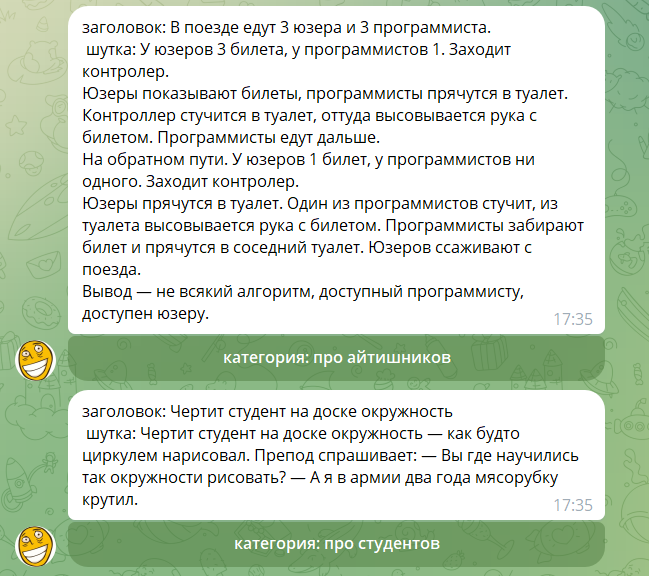


Рисунок 3 – результаты до и после добавления новой шутки

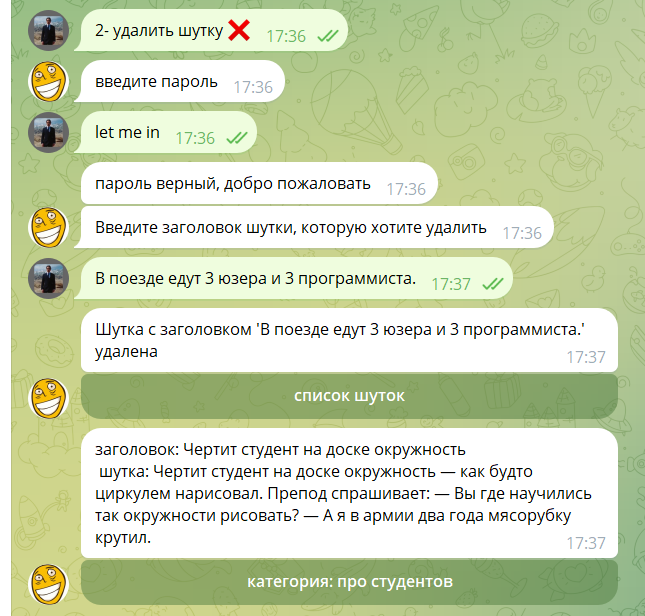
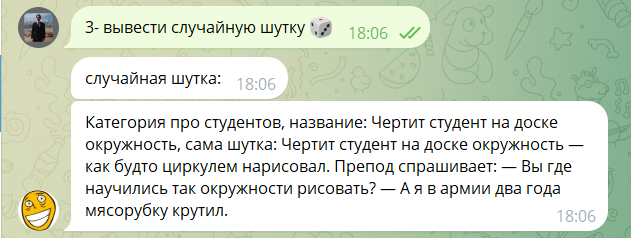


Рисунок 4 – удаление шутки с помощью кнопки 2



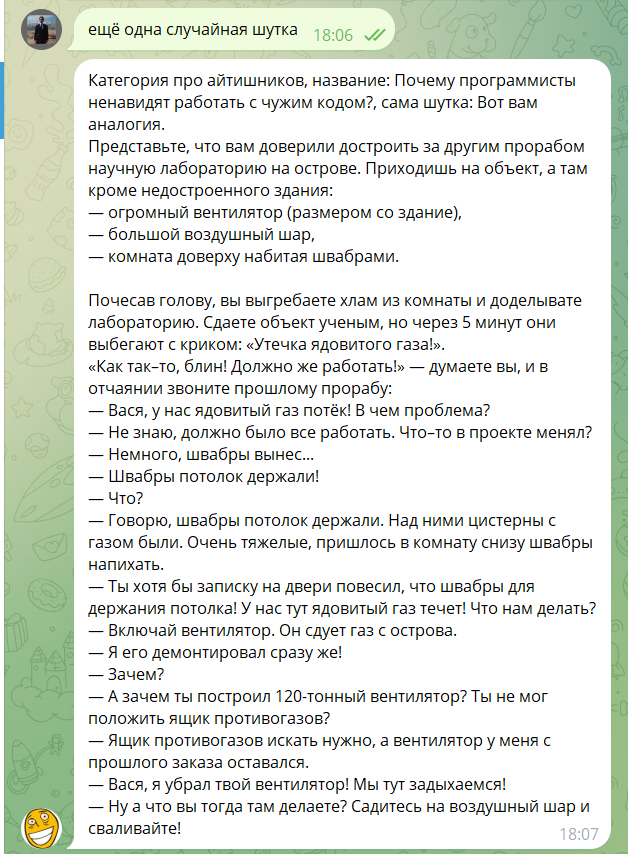
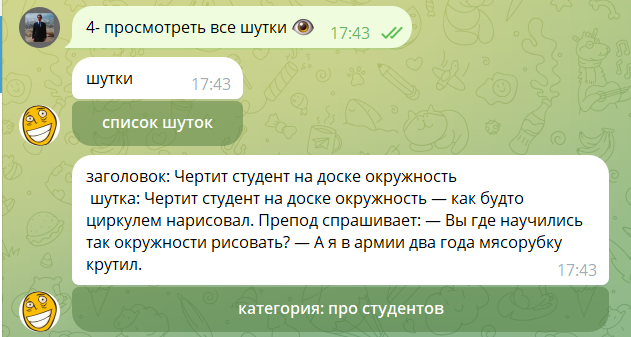


Рисунок 5 – вывод случайной шутки из базы



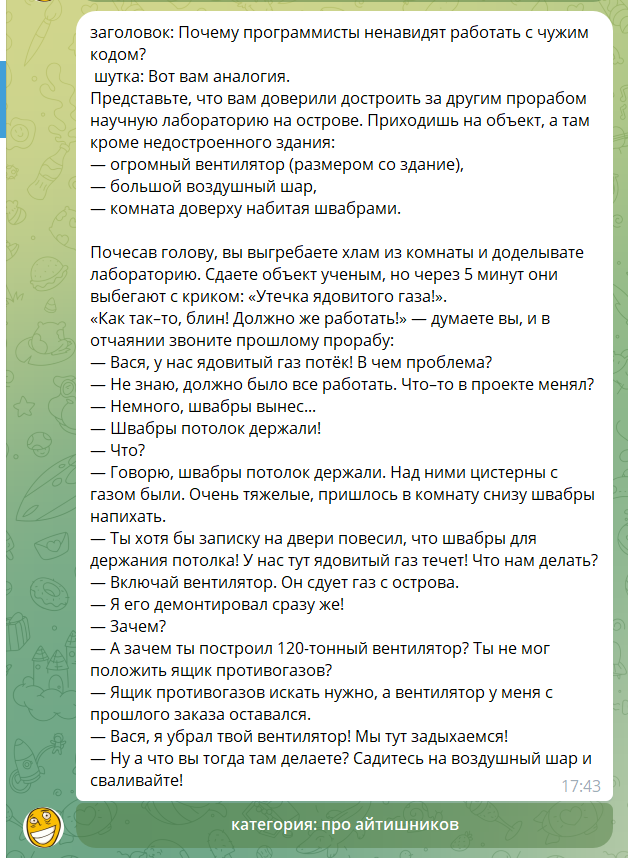


Рисунок 6 – просмотр всего содержимого базы данных с помощью кнопки «список шуток»

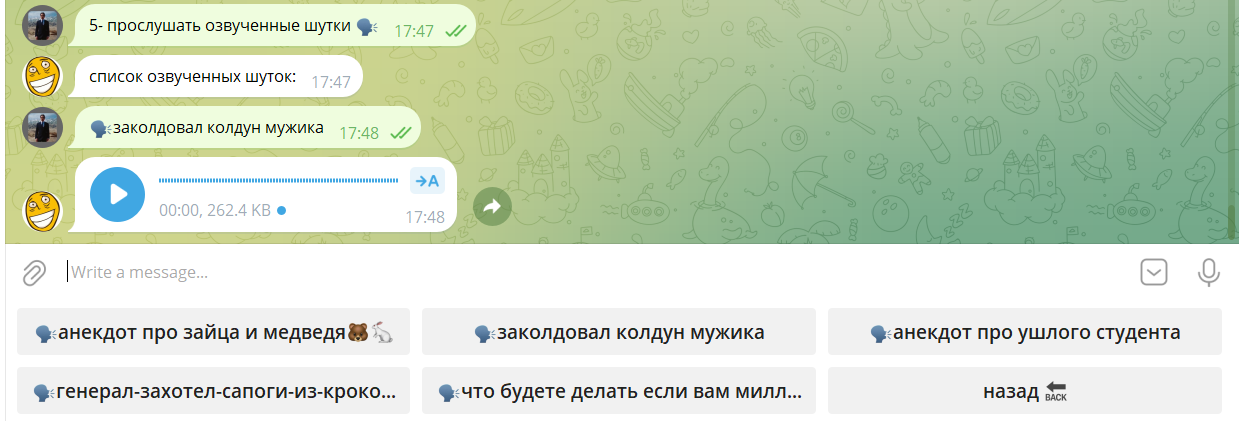


Рисунок 7 – прослушивание озвученных анекдотов с помощью кнопки 5, которая вызывает меню с выбором озвученных анекдотов

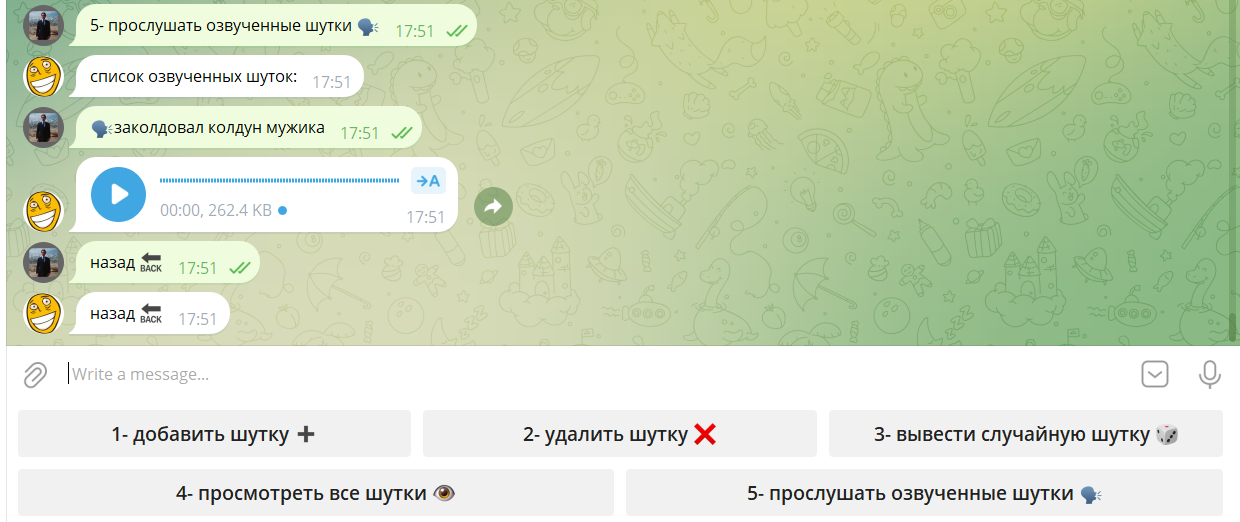
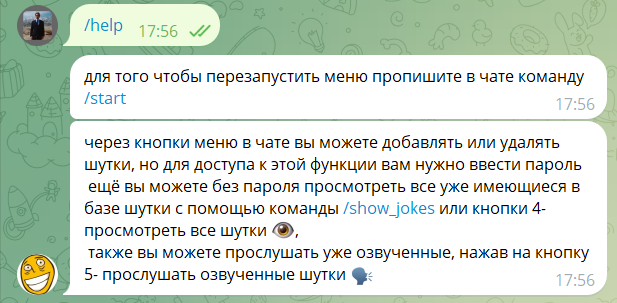


Рисунок 8 – возврат в главное меню с помощью кнопки «назад»

  
Рисунок 9 – команда /help

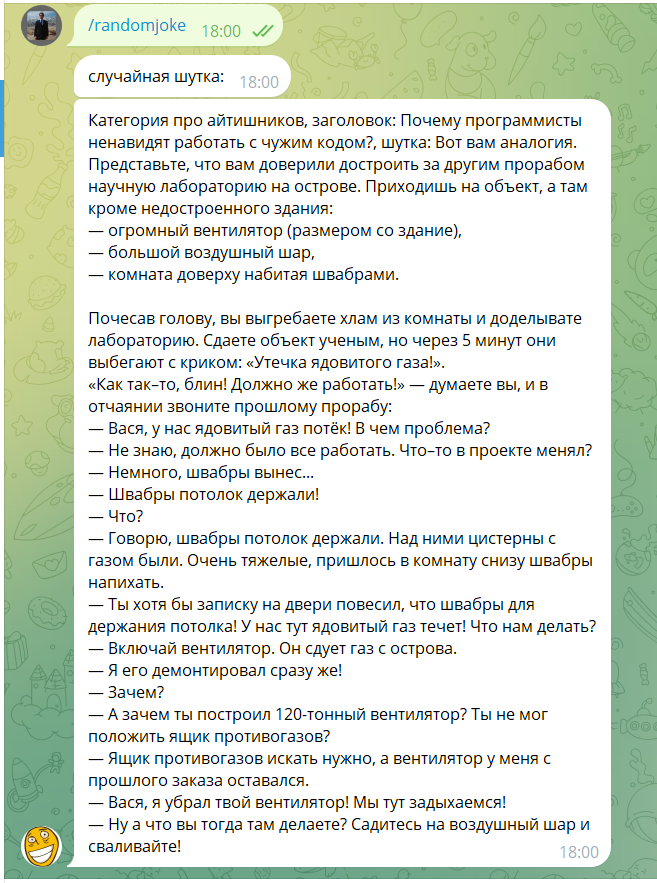
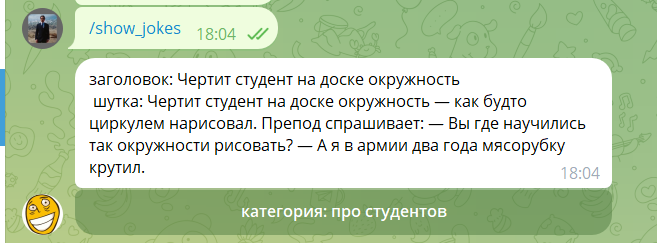


Рисунок 10 – команда /randomjoke



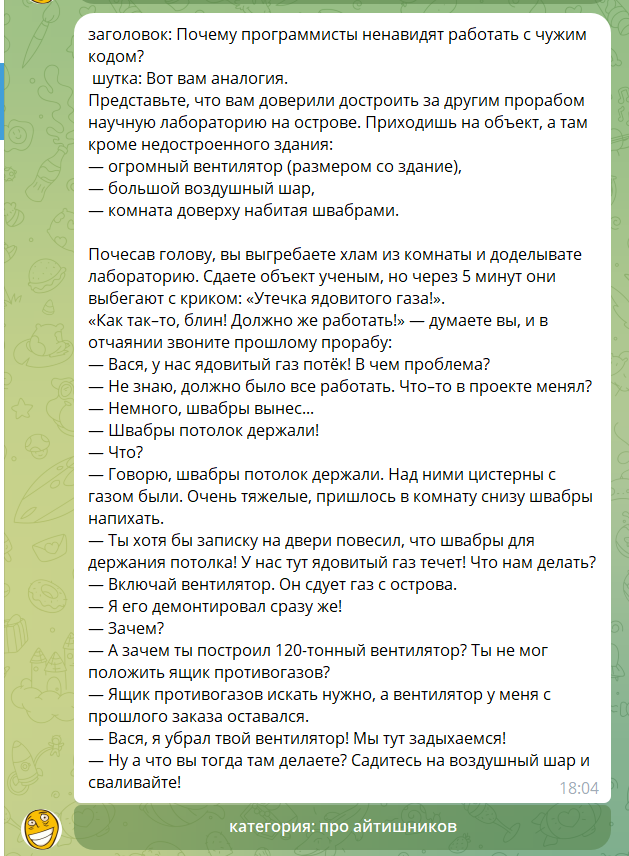


Рисунок 11 – команда /show\_jokes

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

### федеральное государственное автономное образовательное учреждение высшего образования

«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ»

(ГУАП)

КАФЕДРА ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ И СЕТЕЙ

|  |
| --- |
| ПРОГРАММА |
| Бот для мессенджера «Telegram» |
| Описание программы |
| 44.4243.23-12 |
|  |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ ГР. | 4243 |  |  |  | В.С. Суковатицин |
|  |  |  | подпись, дата |  | инициалы, фамилия |

Санкт-Петербург

2023

Аннотация

В документе указаны общие сведения о программе «Бот для мессенджера «Telegram»». Приведено общее описание функционирования программы. Описаны исходные и выходные параметры и логика работы программы, даны сведения об используемых технических средствах и запуске программы.

**СОДЕРЖАНИЕ**

4.1 Общие сведения

4.2 Функциональное назначение

4.3 Описание логической структуры

4.4 Спецификация функции

4.5 Используемые технические средства

4.6 Вызов и загрузка

**4.1 Общие сведения**

Программа написана на Python с использованием сторонних библиотек и состоит из меню программы и набора функций. Она требует использования клавиатуры, так как только с помощью клавиатуры пользователь может вносить изменения в базу данных. Для функционирования данной программы необходимо иметь подключение к интернету и использовать мессенджер «Telegram».

**4.2 Функциональное назначение**

С помощью данной программы пользователь может создавать базу данных, добавлять в неё новые или удалять уже существующие в ней данные, искать по этим данным.

**4.3 Описание логической структуры**

Программа написана на языке Python с использованием сторонних библиотек. При открытии программы запускается бот в мессенджере «Telegram». При вводе команды /start выводится приветствие и появляется меню с кнопками. Для продолжения работы пользователю необходимо выбрать соответствующее действие, просто нажав на кнопку или отправить в чат сообщение с точно таким же текстом, который написан на кнопке.

Варианты выбора следующие:

1. Добавить шутку
2. Удалить шутку
3. Вывести случайную шутку
4. Просмотреть все шутки
5. Прослушать озвученные шутки

**4.4 Спецификация** **основных функций**

1)Функция greetings

Входные данные: команда

Назначение: Вывод приветствия и вывод главного меню

2)Функция randomjoke

Входные данные: команда

Назначение: вывод случайной шутки из базы

3)Функция help

Входные данные: команда

Назначение: вывод подсказки по использованию программы

4)Функция show\_jokes

Входные данные: команда

Назначение: вывод всего содержимого базы данных

5) Функция bot\_message

Входные данные: текст из сообщения в чате

Назначение:

Вывод и функционирование главного меню

6) Функция password\_check\_add\_joke

Входные данные: пароль из сообщения в чате

Назначение: проверка пароля для допуска к добавлению шуток в базу данных

7) Функция joke\_category

Входные данные: категория шутки из сообщения в чате

Назначение: ввод категории для добавляемой шутки и вызов функции для ввода заголовка шутки

8) Функция joke\_name

Входные данные: заголовок шутки из сообщения в чате

Назначение: ввод заголовка шутки и вызов функции ввода текста шутки

9) Функция joke\_text

Входные данные: текст шутки из сообщения в чате

Назначение: ввод текста шутки, добавление в базу данных шутки и создание кнопки «список шуток», которая отвечает за вывод всей базы шуток

10) Функция password\_check\_delete\_joke

Входные данные: пароль из сообщения в чате

Назначение: проверка пароля для допуска к удалению из базы данных

11) Функция delete\_joke

Входные данные: заголовок удаляемой шутки из сообщения в чате

Назначение: удаление шутки с введённым заголовком из базы данных, создание кнопки «список шуток», которая отвечает за вывод всей базы шуток

12) Функция callback

Входные данные: тип данных из кнопки

Назначение: вывод всей базы шуток с разбиением на категории

**4.5 Используемые технические средства**

Для нормального функционирования программы необходимо наличие ПК, на котором установлен язык Python и библиотека pyTelegramBotAPI, подключение к интернету и доступ к мессенджеру «Telegram»

**4.6 Вызов и загрузка**

Вызов и загрузка программы осуществляется в следующей последовательности: Запуск программы на компьютере, создание файла базы данных sqlite по команде /start, вывод меню программы. Дальнейшие действия определяются соответствующими элементами управления.

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

### федеральное государственное автономное образовательное учреждение высшего образования

«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ»

(ГУАП)

КАФЕДРА ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ И СЕТЕЙ

|  |
| --- |
| ПРОГРАММА |
| Бот для мессенджера «Telegram» |
| Описание применения |
| 44.4243.23-12 |
|  |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ ГР. | 4243 |  |  |  | В.С. Суковатицин |
|  |  |  | подпись, дата |  | инициалы, фамилия |

Санкт-Петербург

2023

Аннотация

В документе приводится назначение программы, её основные характеристики и область применения. Также приводятся общие характеристики входной и выходной информации.

**СОДЕРЖАНИЕ**

5.1 Назначение программы

5.2 Условие применения

5.3 Описание задачи

5.4 Характеристика

**5.1 Назначение программы**

Программа «Бот для мессенджера «Telegram» » предназначена для упрощения хранения шуток и анекдотов в текстовом или звуковом формате и для развлечения пользователя.

**5.2 Условия применения**

Программа написана на языке Python. Для нормального функционирования программы необходимо наличие ПК, на котором установлен язык Python и библиотека pyTelegramBotAPI, подключение к интернету и доступ к мессенджеру «Telegram»

**5.3 Описание задачи**

Используя технологию структурного программирования, разработать программу бот для мессенджера «Telegram».

Программа должна иметь удобный интерфейс, быстро работать.

Бот предназначен для упрощения хранения шуток и анекдотов в текстовом или звуковом формате и для развлечения пользователя.

* 1. **Характеристики занимаемой памяти**

main.py – 13.7кб

Папка с проектом в среде разработки вместе со всеми необходимыми файлами (в том числе с файлом main.py) - 17.3мб

**ЗАКЛЮЧЕНИЕ**

В ходе выполнения курсовой работы была разработана программа «Бот для мессенджера «Telegram»», в процессе работы были ознакомительно изучены базы данных, библиотека pyTelegramBotAPI и углублены знания по языку python.

**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1 ГОСТ 19.201-78 Техническое задание. Требования к содержанию и оформлению.

2 ГОСТ 19.202-78 Спецификация. Требования к содержанию и оформлению.

3 ГОСТ 19.401-78 Текст программы. Требования к содержанию и оформлению.

4 ГОСТ 19.402-78 Описание программы. Требования к содержанию и оформлению.

5 ГОСТ 19.404-79 Пояснительная записка. Требования к содержанию и оформлению.

6 ГОСТ 19.502-78 Описание применения. Требования к содержанию и оформлению.

7 Бариков Л.Н. Документирование программного обеспечения. [Текст]: Методические указания к выполнению курсовой работы / Л.Н. Бариков. – СПб.: ГУАП, 2013. – 24 с.