**Московский государственный технический**

**университет им. Н.Э. Баумана**

Факультет «Информатика и системы управления»

Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Базовые компоненты интернет-технологий»

Отчет по лабораторной работе №5-6

|  |  |  |
| --- | --- | --- |
| Выполнил: |  | Проверил: |
| студент группы ИУ5-31Б |  | преподаватель каф. ИУ5 |
| Коваленко  Алексей Викторович |  | Гапанюк Юрий Евгеньевич |
|  |  |  |
| Подпись:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Дата:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  | Подпись:\_\_\_\_\_\_\_\_\_\_\_\_\_  Дата:\_\_\_\_\_\_\_\_\_\_\_\_\_ |
|  |  |  |
|  |  |  |

Москва, 2021 г.

**Лабораторная работа №5-6**

**Описание задания**

### Задание:

1. Разработайте простого бота для Discord.
2. Разработайте бота для Discord. Бот должен реализовывать конечный автомат из трех состояний.

**Текст программы**

Файл bot.py

import discord  
import sqlite3  
import time  
from discord.ext import commands  
from setting import settings  
  
intents = discord.Intents(messages=True, guilds=True, members = True)  
intents.reactions = True  
  
TOKEN = settings['token']  
client = commands.Bot(command\_prefix='&', intents=intents)  
client.remove\_command('help')  
  
connection = sqlite3.connect('oracle.db')  
cursor = connection.cursor()  
  
  
  
@client.event  
async def on\_ready():  
  
 cursor.execute("""CREATE TABLE IF NOT EXISTS socrating (  
 id INT,  
 rating INT  
 )""")  
 connection.commit()  
  
 cursor.execute("""CREATE TABLE IF NOT EXISTS warnings (  
 user\_id INT,  
 admin\_id INT,  
 descr TEXT,  
 amount INT,  
 warntype TEXT,  
 id INT  
 )""")  
  
 cursor.execute("""CREATE TABLE IF NOT EXISTS modcase (  
 case\_id INT,  
 user\_id INT,  
 admin\_id INT,  
 descr TEXT  
 )""")  
  
 connection.commit()  
  
 for guild in client.guilds:  
 for member in guild.members:  
 if cursor.execute(f"SELECT id FROM socrating WHERE id = {member.id}").fetchone() is None:  
 cursor.execute(f"INSERT INTO socrating VALUES({member.id},30)")  
 else:  
 pass  
 connection.commit()  
 print('Join done')  
 weekly()  
  
  
@client.event  
async def on\_memeber\_join(member):  
 if cursor.execute(f"SELECT id FROM socrating WHERE id = {member.id}").fetchone() is None:  
 cursor.execute(f"INSERT INTO socrating VALUES({member.id},30,'{member}')")  
 connection.commit()  
 else:  
 pass  
  
  
@client.command(aliases = ['rating'])  
async def \_\_rating(ctx, member: discord.Member=None):  
 if member == None:  
 await ctx.send(embed=discord.Embed(  
 description = f"""Social rating of \*\*{ctx.author}\*\* is \*\*{cursor.execute(f"SELECT rating FROM socrating WHERE id = {ctx.author.id}").fetchone()[0]}\*\* points"""  
 ))  
 else:  
 await ctx.send(embed=discord.Embed(  
 description = f"""Social rating of \*\*{member}\*\* is \*\*{cursor.execute(f"SELECT rating FROM socrating WHERE id = {member.id}").fetchone()[0]}\*\* points"""  
 ))  
  
  
@client.command(aliases = ['addrating','add'])  
@commands.has\_any\_role(\*settings['admins'])  
async def \_\_addrating(ctx,member: discord.Member = None, amount : int=None):  
 print('add triggered')  
 if member==None:  
 await ctx.send(embed=discord.Embed(  
 description = "Please enter user, for who you want to manage rating"  
 ))  
 else:  
 if amount==None:  
 await ctx.send(embed=discord.Embed(  
 description = "Please enter amount of rating, which you want to add"  
 ))  
 else:  
 cursor.execute("UPDATE socrating SET rating = rating +{} WHERE id = {}".format(amount,member.id))  
 connection.commit()  
  
 await ctx.message.add\_reaction('✅')  
 await notification(member,ctx)  
 await cap\_check(member)  
  
  
  
  
@client.command(aliases = ['decreaserating','decrease','minus'])  
@commands.has\_any\_role(\*settings['admins'])  
async def \_\_decrating(ctx,member: discord.Member = None, amount : int=None):  
 if member==None:  
 await ctx.send(embed=discord.Embed(  
 description = "Please enter user, for who you want to manage rating"  
 ))  
 else:  
 if amount==None:  
 await ctx.send(embed=discord.Embed(  
 description = "Please enter amount of rating, which you want to decrease"  
 ))  
 else:  
 cursor.execute("UPDATE socrating SET rating = rating - {} WHERE id = {}".format(amount,member.id))  
 connection.commit()  
 await ctx.message.add\_reaction('✅')  
 await notification(member,ctx)  
 await cap\_check(member)  
  
  
  
@client.event  
async def on\_command\_error(ctx, error):  
 if isinstance(error, commands.MissingAnyRole):  
 await ctx.send("\*bonk\* You have no permissions here")  
  
  
async def weekly():  
 sleep(604800)  
 for guild in client.guilds:  
 for member in guild.members:  
 cursor.execute("UPDATE socrating SET rating = rating + 5 WHERE id = {}".format(member.id))  
 cap\_check(member)  
 notification(member)  
 s  
async def cap\_check(member): #our current cap is 100  
 amount= cursor.execute("SELECT rating FROM socrating WHERE id = {}".format(member.id)).fetchone()[0]  
 if amount>100:  
 print('deacrease done')  
 cursor.execute("UPDATE socrating SET rating = 100 WHERE id = {}".format(member.id))  
 connection.commit()  
  
  
async def notification(member,ctx):  
 amount= cursor.execute("SELECT rating FROM socrating WHERE id = {}".format(member.id)).fetchone()[0]  
 if amount<25:  
 print('notif done')  
 logs = client.get\_channel(settings['logs'])  
 await logs.send(embed=discord.Embed(  
 description = f"""\*\*{member}\*\* reached low level of reputation. Current points is \*\*{cursor.execute(f"SELECT rating FROM socrating WHERE id = {ctx.author.id}").fetchone()[0]}\*\*."""  
 ))  
  
@client.command(aliases = ['warn'])  
@commands.has\_any\_role(\*settings['admins'])  
async def \_\_warn(ctx,member: discord.Member = None, amount : int=None, description: str =None):  
 if member==None:  
 await ctx.send(embed=discord.Embed(  
 description = "Please enter user, for who you want to warn"  
 ))  
 else:  
 if amount==None:  
 await ctx.send(embed=discord.Embed(  
 description = "Please enter amount of rating, which you want to decrease"  
 ))  
 else:  
 cursor.execute("UPDATE socrating SET rating = rating - {} WHERE id = {}".format(amount,member.id))  
 connection.commit()  
 if amount <10:  
 warntype = 'lightwarning'  
 elif amount >= 10 and amount <25:  
 warntype = 'warning'  
 else:  
 warntype = 'hardwarning'  
 warn\_id=cursor.execute(f"SELECT MAX(id) FROM warnings").fetchone()[0]  
 print(warn\_id)  
 cursor.execute(f"INSERT INTO warnings VALUES({member.id},{ctx.author.id},'{description}',{amount},'{warntype}',{warn\_id+1})")  
 connection.commit()  
 settings['warn']=warn\_id  
 connection.commit()  
 await ctx.message.add\_reaction('✅')  
 await notification(member,ctx)  
 await cap\_check(member)  
  
  
@client.command(aliases = ['unwarn'])  
@commands.has\_any\_role(\*settings['admins'])  
async def \_\_unwarn(ctx,case\_id:int=None):  
 if case\_id==None:  
 await ctx.send(embed=discord.Embed(  
 description = "Please enter id of warning, which you want to delete"  
 ))  
 else:  
 amount = cursor.execute(f"SELECT amount FROM warnings WHERE id = {case\_id}").fetchone()[0]  
 member = cursor.execute(f"SELECT user\_id FROM warnings WHERE id = {case\_id}").fetchone()[0]  
 cursor.execute("UPDATE socrating SET rating = rating + {} WHERE id = {}".format(amount,member))  
 cursor.execute(f"DELETE FROM warnings WHERE id = {case\_id}")  
 connection.commit()  
 await ctx.message.add\_reaction('✅')  
 await notification(member,ctx)  
 await cap\_check(member)  
  
@client.command(aliases = ['warnings'])  
async def \_\_warning(ctx,member: discord.Member = None):  
 if member==None:  
 await ctx.send(embed=discord.Embed(  
 description = "Please enter user"  
 ))  
 else:  
 amount = cursor.execute(f"SELECT COUNT(\*) FROM warnings WHERE user\_id = {member.id}").fetchone()[0]  
 res = cursor.execute(f"SELECT \* FROM warnings WHERE user\_id = {member.id}").fetchmany(amount)  
 desc = f"Warnings of {member} \n"  
 for i in range (0,amount):  
 desc += f"Case {res[i][5]} of {res[i][4]}: User - {client.get\_user(res[i][0])}, Admin - {client.get\_user(res[i][1])},\nDescription:{res[i][2]}, Deacreased for {res[i][3]}\n\n"  
 await ctx.send(  
 embed=discord.Embed(  
 description = desc  
 ))  
  
  
  
  
  
#future undates 1  
@client.command(aliases = ['help'])  
async def \_\_1warning(ctx,member: discord.Member = None, amount : int=None):  
 pass  
@client.command(aliases = ['admins'])  
async def \_\_2warning(ctx,member: discord.Member = None, amount : int=None):  
 pass  
@client.command(aliases = ['mods'])  
async def \_\_3warning(ctx,member: discord.Member = None, amount : int=None):  
 pass  
  
#future undates2  
@client.command(aliases = ['mute'])  
@commands.has\_any\_role(\*settings['admins'])  
async def \_\_4warning(ctx,member: discord.Member = None, amount : int=None):  
 pass  
@client.command(aliases = ['unmute'])  
@commands.has\_any\_role(\*settings['admins'])  
async def \_\_5warning(ctx,member: discord.Member = None, amount : int=None):  
 pass  
  
#future undates3  
@client.command(aliases = ['ban'])  
@commands.has\_any\_role(\*settings['admins'])  
async def \_\_6warning(ctx,member: discord.Member = None, amount : int=None):  
 pass  
@client.command(aliases = ['unban'])  
@commands.has\_any\_role(\*settings['admins'])  
async def \_\_7warning(ctx,member: discord.Member = None, amount : int=None):  
 pass  
@client.command(aliases = ['kick'])  
@commands.has\_any\_role(\*settings['admins'])  
async def \_\_8warning(ctx,member: discord.Member = None, amount : int=None):  
 pass  
  
  
  
  
  
  
  
  
#only lab thing. delete after lab  
global state  
state = 0  
global joke\_line  
joke\_line=[]  
  
@client.event  
async def on\_message(message):  
 global state  
 if message.author != client.user:  
 if state !=0:  
 chan = client.get\_channel(920072025303830599)  
 if message.content == ' reset':  
 await reset()  
 await chan.send(embed=discord.Embed(  
 description = 'Module reseted'))  
 elif message.content == 'exit':  
 await exit()  
 await chan.send(embed=discord.Embed(  
 description = 'Module off'))  
 elif state == 1 :  
 if message.content == 'back':  
 await reset()  
 await chan.send(embed=discord.Embed(  
 description = 'We are back to previous stage'))  
 else:  
 joke\_line.append(message.content)  
 state+=1  
 await chan.send(embed=discord.Embed(  
 description = 'Enter second word to joke'))  
 elif state ==2:  
 if message.content == 'back':  
 state-=1  
 joke\_line.pop()  
 await chan.send(embed=discord.Embed(  
 description = 'We are back to previous stage'))  
 else:  
 joke\_line.append(message.content)  
 state+=1  
 await chan.send(embed=discord.Embed(  
 description = 'Enter third word to joke'))  
 elif state ==3:  
 if message.content == 'back':  
 state-=1  
 joke\_line.pop()  
 await chan.send(embed=discord.Embed(  
 description = 'We are back to previous stage'))  
 else:  
 joke\_line.append(message.content)  
 state+=1  
 await chan.send(embed=discord.Embed(  
 description = 'Do you want to see result of your work?(Yes/No)'))  
 elif state ==4:  
 if message.content.lower()=='yes':  
 await chan.send(embed=discord.Embed(  
 description = f'Идет {joke\_line[0]} по {joke\_line[1]}, видит, {joke\_line[2]} горит. Сел в нее и сгорел.'))  
 await reset()  
 elif message.content.lower()=='no':  
 await chan.send(embed=discord.Embed(  
 description = 'Im very sad to her that'))  
 await reset()  
 else:  
 await chan.send(embed=discord.Embed(  
 description = 'I SAID YES OR NO!! moron'))  
 await client.process\_commands(message)  
  
  
@client.command(aliases = ['start6'])  
async def \_\_start6(ctx):  
 global state  
 state = 1  
 await ctx.send(embed=discord.Embed(  
 description = 'Stipid joke module activated. Enter first word, whcih we will add to our joke'  
 ))  
  
async def reset():  
 global state  
 state = 1  
 global joke\_line  
 joke\_line = []  
  
async def exit():  
 global state  
 state = 0  
 global joke\_line  
 joke\_line = []  
  
  
  
client.run(TOKEN)

**Экранные формы с примерами выполнения программы**



