**Министерство науки и высшего образования Российской Федерации**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»**

**(Университет ИТМО)**

Факультет **Инфокоммуникационных технологий**

Образовательная программа **Мобильные и облачные технологии**

Направление подготовки(специальность) **09.03.03 Прикладная информатика**

**ПРАКТИЧЕСКАЯ РАБОТА №0.3**

**По дисциплине «Программирование»**

**Тема: Разработка телеграм-бота**

**Выполнил** Шафиков М.А. K3139

**Проверил Терещенко В. В.**

**Дата** 12.10.2023

**Санкт-Петербург 2023**

**Цель работы**

Разработка чат-бота в telegram с помощью языка Python, который будет присылать пользователю ссылки по запросам и позволять проверить лабораторную работу №1

**Ход работы**

1. Создание телеграм-бота с помощью @FatherBot и получение API-токен для работы с ним
2. Изучение библиотеки telebot для работы с ботом
3. Создание и работа над проектом в Pycharm

Был создан проект и написана первая функция, которая позволит боту начать взаимодействие с пользователем (команада /start)

@bot.message\_handler(commands=['start'])  
def send\_welcome(message):  
 user = message.from\_user  
 fn = user.first\_name  
 ln = user.last\_name if user.last\_name else ""  
 bot.send\_message(message.from\_user.id, f"Приветствую вас, {fn} {ln}, в Институте Теплых Мужских Отношений."  
 f"\nМеня зовут It's Moshniy bot.\nЧем могу помочь?", reply\_markup=start\_keyboard())  
 user\_states[message.from\_user.id] = "menu"

Было решено добавить боту «кастомную клавиатуру», которая позволит пользователю быстро отправлять ключевые сообщения, на которые бот будет реагировать

def start\_keyboard():  
 keyboard = types.ReplyKeyboardMarkup(row\_width=2, resize\_keyboard=True)  
 link\_reg = types.KeyboardButton("Запись на сдачу")  
 link\_rep = types.KeyboardButton("Ссылки на учебные github")  
 test\_lab2 = types.KeyboardButton("Проверить Lab#1")  
 keyboard.add(link\_reg, link\_rep, test\_lab2)  
 return keyboard  
  
def end\_keyboard():  
 keyboard = types.ReplyKeyboardMarkup(row\_width=1, resize\_keyboard=True)  
 keyboard.add(types.KeyboardButton("Назад"))  
 return keyboard  
  
def reg\_keyboard():  
 keyboard = types.ReplyKeyboardMarkup(row\_width=2, resize\_keyboard=True)  
 inf = types.KeyboardButton("Информатика")  
 prog = types.KeyboardButton("Программирование")  
 alg = types.KeyboardButton("АиСД (Алгосы)")  
 back = types.KeyboardButton("Назад")  
 keyboard.add(inf, prog, alg, back)  
 return keyboard  
  
def rep\_keyboard():  
 keyboard = types.ReplyKeyboardMarkup(row\_width=3, resize\_keyboard=True)  
 inf = types.KeyboardButton("Информатика")  
 prog = types.KeyboardButton("Программирование")  
 alg = types.KeyboardButton("АиСД (Алгосы)")  
 infocom = types.KeyboardButton("ИнфоКом")  
 back = types.KeyboardButton("Назад")  
 keyboard.add(inf, prog, alg, infocom, back)  
 return keyboard

Так же был добавлен словарь состояния пользователей, который поможет боту определить в как сейчас отвечать пользователю. Были написаны другие функции общения

user\_states = {}

@bot.message\_handler(func=lambda message: user\_states.get(message.from\_user.id) == "menu")  
def menu(message):  
 if message.text == "Запись на сдачу":  
 bot.send\_message(message.from\_user.id, wait\_message(), reply\_markup=reg\_keyboard())  
 user\_states[message.from\_user.id] = "write"  
 elif message.text == "Ссылки на учебные github":  
 bot.send\_message(message.from\_user.id, wait\_message(), reply\_markup=rep\_keyboard())  
 user\_states[message.from\_user.id] = "github"  
 elif message.text == "Проверить Lab#1":  
 bot.send\_message(message.from\_user.id, wait\_message(), reply\_markup=end\_keyboard())  
 user\_states[message.from\_user.id] = "test"  
 else:  
 rand\_message(message.text)  
 bot.send\_message(message.from\_user.id, rand\_message(message.text))  
  
@bot.message\_handler(func=lambda message: user\_states.get(message.from\_user.id) == "test")  
def test(message):  
 if message.text == "Назад":  
 bot.send\_message(message.from\_user.id, wait\_message(), reply\_markup=start\_keyboard())  
 user\_states[message.from\_user.id] = "menu"  
 else:  
 if is\_correct\_string(message.text):  
 try:  
 result = eval(message.text)  
 bot.send\_message(message.from\_user.id, result)  
 except Exception as e:  
 bot.send\_message(message.from\_user.id, "Code Error!!!!")  
 else:  
 bot.send\_message(message.from\_user.id, "Неправильный формат строки")  
  
@bot.message\_handler(func=lambda message: user\_states.get(message.from\_user.id) == "write")  
def write(message):  
 if message.text == "Информатика":  
 bot.send\_message(message.from\_user.id, "https://docs.google.com/spreadsheets/d/1JYlDSwy1Vq476NOECvPuv2WRePUNuDbUh0foLmCudKk/edit#gid=0")  
 elif message.text == "Программирование":  
 bot.send\_message(message.from\_user.id, "https://docs.google.com/spreadsheets/d/15vU7odXOseENO622uKrtHb5fEtA1CyiPgBp7o71w6Jg/edit#gid=0")  
 elif message.text == "АиСД (Алгосы)":  
 bot.send\_message(message.from\_user.id, "https://docs.google.com/spreadsheets/d/1KqvCFxAKWfi1XBONL9vUcadn424o-0GKa3ziSlR94hU/edit#gid=1064470743")  
 elif message.text == "Назад":  
 bot.send\_message(message.from\_user.id, wait\_message(), reply\_markup=start\_keyboard())  
 user\_states[message.from\_user.id] = "menu"  
 else:  
 rand\_message(message.text)  
 bot.send\_message(message.from\_user.id, rand\_message(message.text))  
  
@bot.message\_handler(func=lambda message: user\_states.get(message.from\_user.id) == "github")  
def github(message):  
 canansw = ["Информатика", "Программирование", "АиСД (Алгосы)", "ИнфоКом"]  
 if message.text == "Информатика":  
 bot.send\_message(message.from\_user.id, "https://github.com/cs-itmo-2023")  
 elif message.text == "Программирование":  
 bot.send\_message(message.from\_user.id, "https://repeated-squid-629.notion.site/2023-2024-36763ed0edbf4cebbc722bdf21c94db6")  
 elif message.text == "АиСД (Алгосы)":  
 bot.send\_message(message.from\_user.id, "https://github.com/Algorithms-ICT-2023-2024")  
 elif message.text == "ИнфоКом":  
 bot.send\_message(message.from\_user.id, "https://drive.google.com/drive/folders/1TFb\_bPQIp8WMBziS-4qtCNUXyylWLdMS")  
 elif message.text == "Назад":  
 bot.send\_message(message.from\_user.id, wait\_message(), reply\_markup=start\_keyboard())  
 user\_states[message.from\_user.id] = "menu"  
 else:  
 rand\_message(message.text)  
 bot.send\_message(message.from\_user.id, rand\_message(message.text))

bot.polling(none\_stop=True, interval=0)

Так же была добавлены функции из лабораторной работы №1 и адаптирован код, чтобы пользователь мог сразу протестировать работу. (Код в отсчете по лабораторной работе №1)

**Вывод**

В результате проделанной работы была изучена библиотека telebot и основные способы работы с телеграм-ботом, а так же проверены сразу 2 лабораторные работы.