

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

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

Курс «Парадигмы и конструкции языков программирования »

Отчет по лабораторной работе №6
«Разработка бота на основе конечного автомата для Telegram с
использованием языка Python.»

Выполнил:

студент
группы
РТ5-31Б

Ермаков И.А.

Описание задания:

Разработайте бота для Telegram. Бот должен реализовывать конечный автомат из трех состояний.

Текст программы:

Telegram_bot.py

```
from aiogram import Bot, Dispatcher, F, Router
from aiogram.fsm.context import FSMContext
from aiogram.fsm.state import State, StatesGroup
from aiogram.fsm.storage.memory import MemoryStorage
from aiogram.types import Message
from aiogram.types import FSInputFile
import random
import asyncio

# Инициализация бота и диспетчера
API_TOKEN = '7146934587:AAEET9gXIfB3AXmozVuyeOV-Zp0DTzv_UXw'

bot = Bot(token=API_TOKEN)
storage = MemoryStorage()
router = Router()
dp = Dispatcher(storage=storage)
dp.include_router(router)

# Определение состояний
class Form(StatesGroup):
    state1 = State() # Первое состояние
    state2 = State() # Второе состояние
    state3 = State() # Третье состояние

# Список случайных ответов для оживления взаимодействий
responses = {
    "state1": [
        "Добро пожаловать в состояние 1! Что скажете?",
        "Ого, вы в состоянии 1. Это только начало!",
        "Состояние 1 приветствует вас! Ждем ваших действий. (Напиши дальше)"
    ],
    "state2": [
        "Теперь вы в состоянии 2. Продолжайте!",
```

```

        "Состояние 2! Как ваши успехи?",
        "Великолепно, вы достигли состояния 2! Что дальше?"
    ],
    "state3": [
        "Это состояние 3! Вот вы где.",
        "Финальный шаг в состоянии 3. Готовы завершить?",
        "Отлично, вы дошли до состояния 3. Пора заканчивать!"
    ],
    "finish": [
        "Вы прошли весь путь! Хотите начать сначала? Напишите /start.",
        "Поздравляем, вы завершили все состояния! Снова /start?",
        "Конец пути! Если хотите повторить, отправьте /start."
    ]
}

```

```

# Хэндлер для команды /start
@router.message(F.text == "/start")
async def start_handler(message: Message, state: FSMContext):
    await state.set_state(Form.state1)
    await message.answer(random.choice(responses["state1"]))

# Хэндлер для состояния 1
@router.message(Form.state1)
async def state1_handler(message: Message, state: FSMContext):
    await state.set_state(Form.state2)
    await message.answer(random.choice(responses["state2"]))

# Хэндлер для состояния 2
@router.message(Form.state2)
async def state2_handler(message: Message, state: FSMContext):
    await state.set_state(Form.state3)
    await message.answer(random.choice(responses["state3"]))

# Хэндлер для состояния 3
@router.message(Form.state3)
async def state3_handler(message: Message, state: FSMContext):
    await state.clear()
    await message.answer(random.choice(responses["finish"]))

async def main():
    await dp.start_polling(bot)

if __name__ == "__main__":
    asyncio.run(main())

```

Пример выполнения программы

