**Московский государственный технический**

**университет им. Н.Э. Баумана**

Факультет «Информатика и системы управления»

Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Парадигмы и конструкции языков программирования»

Отчет по лабораторной работе №6

# «Разработка простого бота для Telegram с использованием языка Python»

|  |  |  |
| --- | --- | --- |
| Выполнил: |  | Проверил: |
| студент группы ИУ5-32Б |  | преподаватель каф. ИУ5 |
| Борисова К.А. |  | Гапанюк Ю.Е. |
| Подпись и дата: |  | Подпись и дата: |

Москва, 2024 г.

**Постановка задачи:**

Разработайте простого бота для Telegram. Бот должен использовать функциональность создания кнопок.

**Текст программы:**

*import* logging

*from* aiogram *import* Bot, Dispatcher, types

*from* aiogram.dispatcher.filters *import* Command, Text

*from* aiogram.utils *import* executor

*from* aiogram.contrib.fsm\_storage.memory *import* MemoryStorage

*from* aiogram.types *import* InlineKeyboardButton, InlineKeyboardMarkup, CallbackQuery

*# Set up logging*

logging.basicConfig(*level*=logging.INFO)

*# Initialize bot and dispatcher*

bot = Bot(*token*='')

storage = MemoryStorage()

dp = Dispatcher(bot, storage)

masters={

    "Парикмахер": "Запись на услугу парикмахера",

    "Мастер маникюра": "Запись на услугу мастра маникюра",

    "Бровист": "Запись на услугу бровиста",

    "Косметолог": "Запись на услугу косметолога"

}

@dp.message\_handler(Command("start"))

async def start(*message*: types.Message):

*await* message.answer('Привет! Я бот салона красоты. Какие услуги вас интересуют?')

    keyboard = InlineKeyboardMarkup()

*for* master *in* masters.keys():

        keyboard.add(InlineKeyboardButton(master, *callback\_data*=master))

*await* message.answer("Выберите мастера:", *reply\_markup*=keyboard)

@dp.callback\_query\_handler(lambda *c*: c.data *in* masters.keys())

async def process\_callback(*callback\_query*: CallbackQuery):

    master\_name = callback\_query.data

*await* bot.send\_message(callback\_query.from\_user.id, masters[master\_name])

*await* callback\_query.answer()

*if* \_\_name\_\_ == '\_\_main\_\_':

    executor.start\_polling(dp, *skip\_updates*=True)

**Анализ результатов:**

