

Защищено:
Гапанюк Ю.Е.

Демонстрация:
Гапанюк Ю.Е.

"__"_____2025 г.

"__"_____2025 г.

**Отчет по лабораторной работе № 5 по курсу
Парадигмы и конструкции языков программирования
ГУИМЦ**

**Тема работы: "Разработка простого бота для Telegram с
использованием языка Python."**

4

(количество листов)

ИСПОЛНИТЕЛЬ:

студент группы ИУ5Ц-54Б

Щетинин Д.С.

(подпись)

"__"_____2025 г.

1. Тема и задание для выполнения лабораторной работы.

Тема работы: "Изучение разработки простого бота для Telegram с использованием языка Python."

Задание:

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

2. Листинг программы

```
from telegram import Update, ReplyKeyboardMarkup, KeyboardButton
from telegram.ext import Application, CommandHandler, MessageHandler, filters, CallbackContext
import random

# Токен вашего бота
TOKEN = "8230982855:AAEi8j5V5-Lw8ADfoPKOrDEgHBZnOG09UEE"

# База данных тренировок
WORKOUTS = {
    "🏠 ♂ Тренировка дома": [
        "Комплекс на сегодня:\n1. Отжимания 4x15\n2. Приседания 4x20\n3. Планка 3 подхода по 1 минуте",
        "Домашний воркаут:\n1. Берпи 3x10\n2. Воздушные приседания 4x15\n3. Скручивания 3x20"
    ],
    "🏃 Кардио": [
        "Интервалы:\n5 мин разминка\n8 циклов: 30 сек спринт + 1 мин ходьба\n5 мин заминка",
        "Беговая программа:\n5 км в комфортном темпе\nПоследние 500 м - ускорение"
    ],
    "💪 Силовая": [
        "Базовая тренировка:\nЖим лежа 4x8\nСтановая тяга 3x6\nПодтягивания 4x макс",
        "Сплит на верх:\nЖим гантелей 4x10\nТяга штанги 4x8\nБицепс + трицепс 3x12"
    ]
}

NUTRITION_TIPS = [
    "Важно! Пейте 2-3 литра воды в день",
    "После тренировки: белок + быстрые углеводы",
    "1.5-2 г белка на кг веса для роста мышц",
    "Здоровые перекусы: орехи, творог, фрукты"
]

async def start(update: Update, context: CallbackContext) -> None:
    keyboard = [
        [KeyboardButton("🏠 ♂ Тренировка дома"), KeyboardButton("🏃 Кардио")],
        [KeyboardButton("💪 Силовая"), KeyboardButton("🍷 Питание")],
        [KeyboardButton("📊 Прогресс"), KeyboardButton("🧠 Мотивация")]
    ]

    reply_markup = ReplyKeyboardMarkup(keyboard, resize_keyboard=True)

    await update.message.reply_text(
        "Привет! Я твой тренер — Спорт!\n\n"
        "Выбери что тебя интересует:",
        reply_markup=reply_markup
    )
```

)

```
async def handle_message(update: Update, context: CallbackContext) -> None:  
    text = update.message.text
```

```
    if text in WORKOUTS:  
        response = random.choice(WORKOUTS[text])  
        await update.message.reply_text(response)  
    elif text == "🍷 Питание":  
        tip = random.choice(NUTRITION_TIPS)  
        await update.message.reply_text(f'Совет по питанию:\n{tip}')  
    elif text == "📊 Прогресс":  
        await update.message.reply_text(  
            "Записывай свои результаты каждый день!\n"  
            "Лучший прогресс — систематичность"  
        )  
    elif text == "🏆 Мотивация":  
        await update.message.reply_text(  
            random.choice([  
                "Сегодняшняя тренировка — завтрашний результат!",  
                "Не пропускай! Будущий ты скажет спасибо",  
                "Один шаг сегодня — победа завтра"  
            ]) )  
    else:  
        await update.message.reply_text(  
            "Используй кнопки для навигации!\n"  
            "Или напиши /start для перезапуска"  
        )
```

```
def main() -> None:  
    application = Application.builder().token(TOKEN).build()  
    application.add_handler(CommandHandler("start", start))  
    application.add_handler(MessageHandler(filters.TEXT & ~filters.COMMAND, handle_message))  
    application.run_polling()
```

```
if __name__ == "__main__":  
    main()
```

3. Результаты работы программы

