Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

Факультет «ГУИМЦ»

Кафедра ИУ5 «Системы обработки информации и управления»

Дисциплина «Базовые компоненты ИТ»

ОТЧЕТ

ПО ЛАБОРАТОРНОЙ РАБОТЕ №6, ДЗ

«Объектно-ориентированные возможности языка Python»

Студент: Амосов П.А., группа ИУ5Ц-52Б

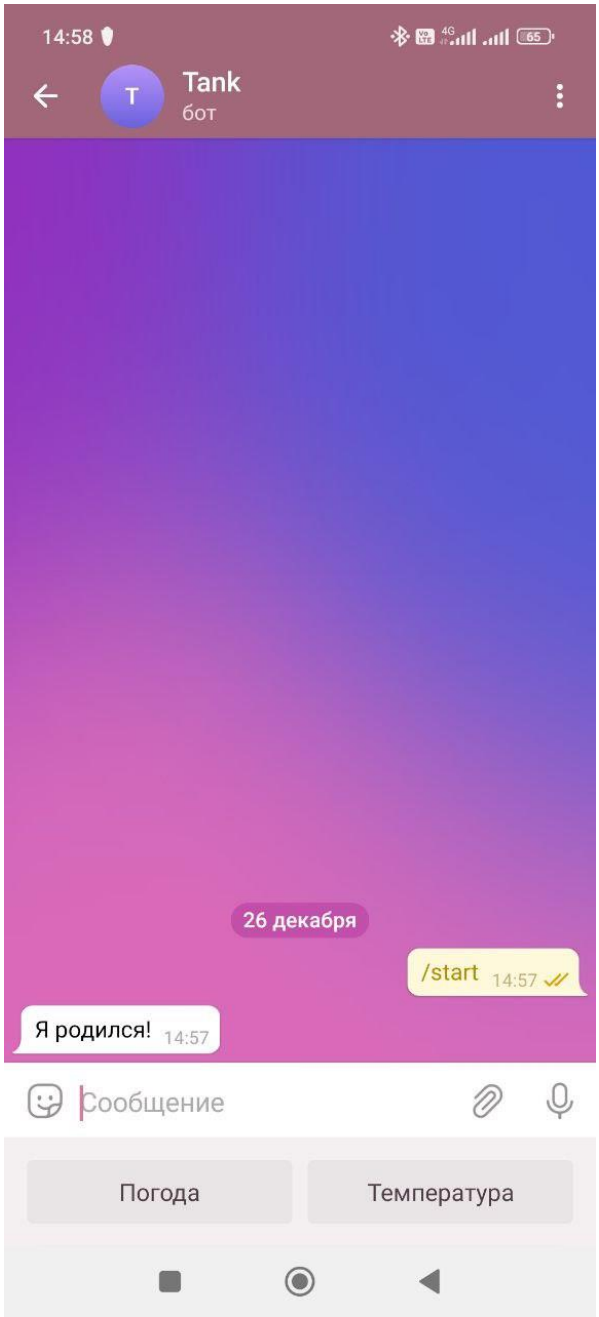
Преподаватель: Канев А.И.

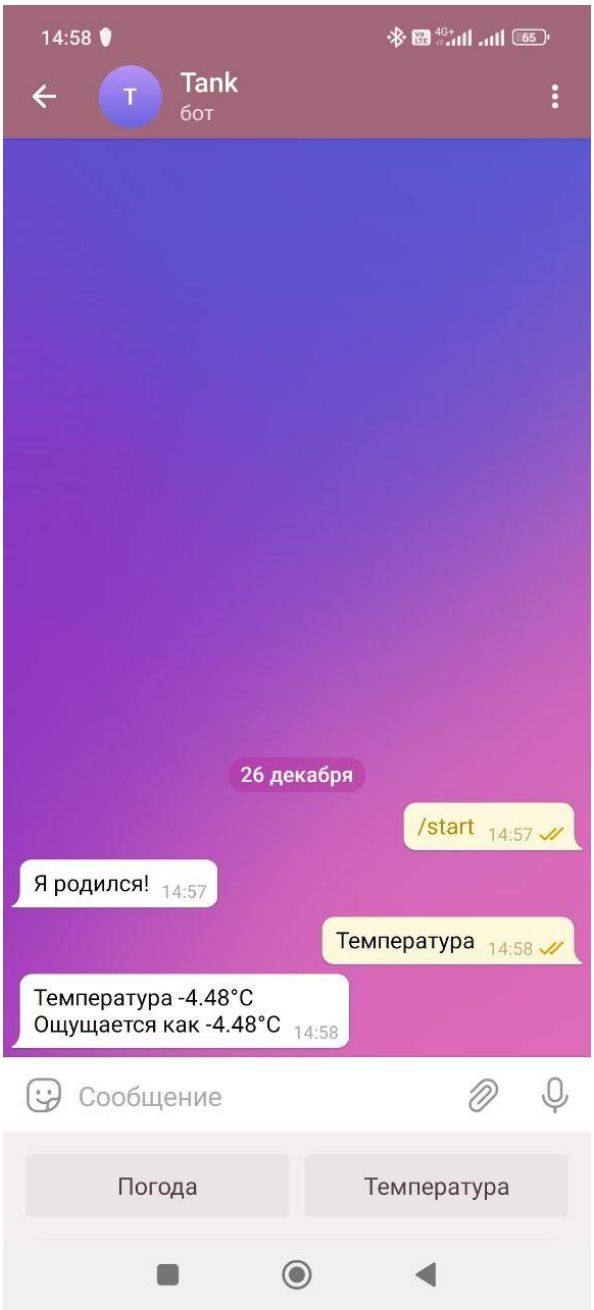
2022г.

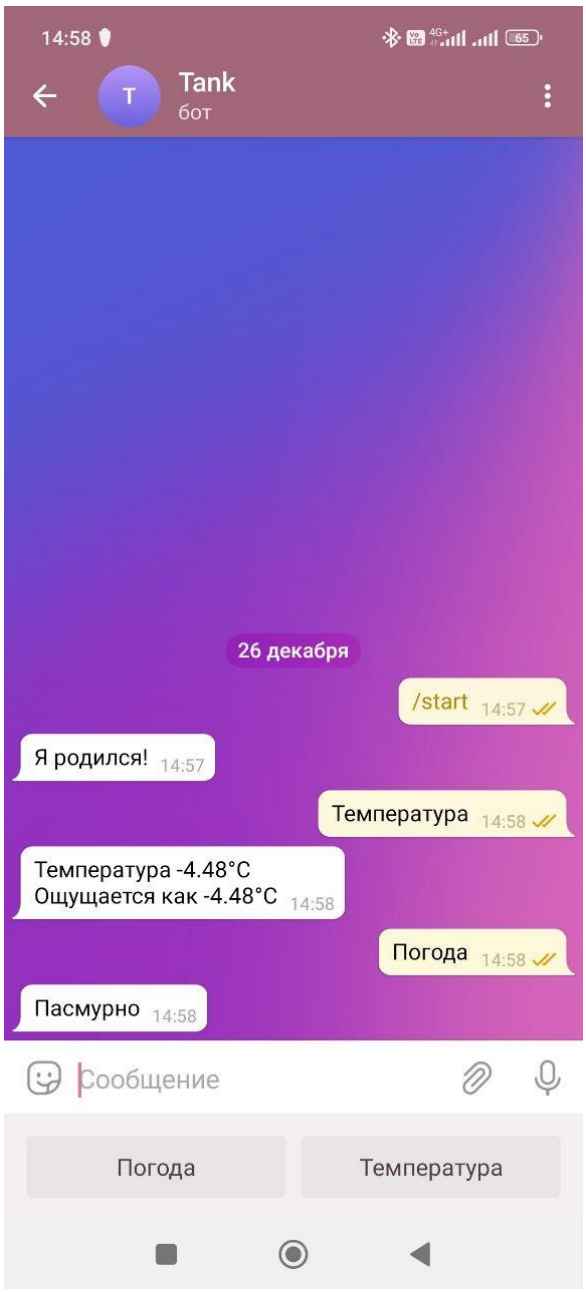
Текст программы

```
import unittest
import telebot
import requests
from telebot import types

bot = telebot.TeleBot('5980815969:AAEXNweXFTqXgayEtuvsbTJJvazVf7ck2II')
appid = '30c3e36da023870d9b246649e8d0b868'
s_city = "Moscow (RU)"
city_id = 0


@bot.message_handler(commands=['start'])
def start(message):
    markup = types.ReplyKeyboardMarkup(resize_keyboard=True)
    btn1 = types.KeyboardButton("Погода")
    btn2 = types.KeyboardButton("Температура")
    markup.add(btn1, btn2)
    if type(message) != str:
        bot.send_message(message.chat.id, text="Я родился!",
reply_markup=markup)
    else:
        raise Exception("Message is str, not telebot object")

    global message1

    message1 = message

    if message == None:
        return None

    return message1


@bot.message_handler(content_types=['text'])
def func(message):
    try:
        res = requests.get(

f"http://api.openweathermap.org/data/2.5/forecast?id=524901&appid={appid}&lan
g=ru&units=metric")
        data = res.json()
        if (not data) or (str(data['cod']) == '404'):
            raise Exception('Page not Found 404')
    except Exception as e:
        bot.send_message(message.chat.id, text=f"Сервер упал :(\nОшибка:
{e}")
        return
    if message == None:
        return
    if message.text == "Погода":
        weather = data['list'][0]['weather'][0]['description'].title()
        bot.send_message(message.chat.id, text=weather)
        markup = types.ReplyKeyboardMarkup(resize_keyboard=True)
        back = types.KeyboardButton("Назад")
        markup.add(back)

    elif message.text == "Температура":
        temp = 'Температура ' + str(data['list'][0]['main']['temp']) + '°C'
        feels_temp = 'Ощущается как ' +
str(data['list'][0]['main']['feels_like']) + '°C'
        markup = types.ReplyKeyboardMarkup(resize_keyboard=True)
        bot.send_message(message.chat.id, text=temp + "\n" + feels_temp,
```

```
reply_markup=markup)
        back = types.KeyboardButton("Назад")
        markup.add(back)

    elif message.text == "Назад":
        markup = types.ReplyKeyboardMarkup(resize_keyboard=True)
        button1 = types.KeyboardButton("Погода")
        button2 = types.KeyboardButton("Температура")
        markup.add(button1, button2)
        bot.send_message(message.chat.id, text="Вы вернулись в главное меню,
чем могу помочь?", reply_markup=markup)
    else:
        bot.send_message(message.chat.id, text="Не знаю такого..")


class test_class(unittest.TestCase):
    def test_false_parameters(self):
        with self.assertRaises(Exception) as context:
            start("text")
        self.assertEqual(
            "Message is str, not telebot object",
            str(context.exception)
        )

    def test_zero_parameters(self):
        with self.assertRaises(TypeError) as context:
            start()
        self.assertEqual(
            "start() missing 1 required positional argument: 'message'",
            str(context.exception)
        )

    def test_func(self):
        message1 = None
        self.assertEqual(func(message1), None)

    def test_func_zero_parameters(self):
        with self.assertRaises(TypeError) as context:
            func()
        self.assertEqual(
            "func() missing 1 required positional argument: 'message'",
            str(context.exception)
        )


bot.polling(none_stop=True)
unittest.main()
```


14:58
Tank
бот
26 декабря
/start 14:57
Я родился! 14:57
Сообщение
Погода
Температура


14:58
Tank
бот
26 декабря
/start 14:57
Я родился! 14:57
Температура 14:58
Температура -4.48°C
Ощущается как -4.48°C 14:58
Сообщение
Погода
Температура



....

Ran 4 tests in 0.170s

OK

Process finished with exit code 0