

Московский государственный технический университет

имени Н. Э. Баумана

Факультет «Информатика и системы управления»

Кафедра ИУ5 «Системы обработки информации и управления»

**Отчёт по домашнему заданию**

Выполнила:

Рыжкова Юлия Николаевна

Группа ИУ5-31Б

Проверил:

Канев Антон Игоревич

Кафедра ИУ5

Москва 2021г.

## **Задание**

Модифицируйте код лабораторной работы №6 таким образом, чтобы он был пригоден для модульного тестирования.

Используя материалы лабораторной работы №4 создайте модульные тесты с применением TDD - фреймворка (2 теста) и BDD - фреймворка (2 теста).

## Текст программы

### Файл config.py

```
from enum import Enum
token = '504*****g'
weather_api = {'key': '2ba80993017dfb2cff22275885728205'}
# Файл базы данных Vedis
db_file = "db.vdb"
# Ключ записи в БД для текущего состояния
CURRENT_STATE = "CURRENT_STATE"
# Состояния автомата
class States(Enum):
    STATE_START = "STATE_START" # Начало нового диалога
    STATE_SITE = "STATE_SITE"
    STATE_TAG = "STATE_TAG"
    STATE_CITY = "STATE_CITY"
```

### Файл bot.py

```
import requests
import telebot
import config
import dbworker
import time
import functions
from functions import *

bot = telebot.TeleBot(config.token)

#Стартовое сообщение
@bot.message_handler(commands=['start'])
def start_message(message):
    bot.send_message(message.chat.id, "Привет!")
    time.sleep(1)
    bot.send_message(message.chat.id, "Меня зовут Микото Мисака, хотя ты наверное уже догадался :)")
    time.sleep(1)
    bot.send_message(message.chat.id, "Я умею создавать картинки с погодой")
    time.sleep(1)
    bot.send_message(message.chat.id, "Мне уже не терпится, давай скорее приступим к созданию!")
    time.sleep(1)
```

```

    bot.send_message(message.chat.id, "И помни, что в
любой момент можешь начать сначала, прописав
/reset")
    time.sleep(1)
    dbworker.set(dbworker.make_key(message.chat.id,
config.CURRENT_STATE),
config.States.STATE_SITE.value)
    markup =
telebot.types.ReplyKeyboardMarkup(resize_keyboard=Tr
ue)
    buttons = ["konachan.net", "deviantart.com"]
    markup.add(*buttons)
    bot.send_message(message.chat.id, "Выбери сайт, с
которого мы возьмём картинку:", reply_markup=markup)

```

```

# По команде /reset будем сбрасывать состояния,
возвращаясь к началу диалога
@bot.message_handler(commands=['reset'])
def cmd_reset(message):
    bot.send_message(message.chat.id, 'Сбрасываем
результаты предыдущего ввода.')
    dbworker.set(dbworker.make_key(message.chat.id,
config.CURRENT_STATE),
config.States.STATE_SITE.value)
    markup =
telebot.types.ReplyKeyboardMarkup(resize_keyboard=Tr
ue)
    buttons = ["konachan.net", "deviantart.com"]
    markup.add(*buttons)
    bot.send_message(message.chat.id, "Выбери сайт, с
которого мы возьмём картинку:", reply_markup=markup)

```

```

# Обработка введённого сайта
@bot.message_handler(func=lambda message:
dbworker.get(dbworker.make_key(message.chat.id,
config.CURRENT_STATE)) ==
config.States.STATE_SITE.value)
def site(message):
    text = message.text
    if text != "konachan.net" and text !=
"deviantart.com":
        # Состояние не изменяется, выводится
сообщение об ошибке

```

```
        bot.send_message(message.chat.id,
'Пожалуйста, выбери из предложенного списка')
        return
    else:
        # Меняем текущее состояние
```

```
dbworker.set(dbworker.make_key(message.chat.id,
config.CURRENT_STATE),
config.States.STATE_CITY.value)
        # Сохраняем сайт
```

```
dbworker.set(dbworker.make_key(message.chat.id,
config.States.STATE_SITE.value), text)
        bot.send_message(message.chat.id, 'Теперь
напиши мне название любого города:',
reply_markup=telebot.types.ReplyKeyboardRemove())
```

```
# Обработка введённых городов
@bot.message_handler(func=lambda message:
dbworker.get(dbworker.make_key(message.chat.id,
config.CURRENT_STATE)) ==
config.States.STATE_CITY.value)
def city(message):
    city = message.text
    if city_check(city):
        # Состояние не изменяется, выводится
сообщение об ошибке
        bot.send_message(message.chat.id, 'Кажется,
такого города не существует...')
        bot.send_message(message.chat.id, 'Попробуй
ввести название города ещё раз:')
        return
    else:
        # Меняем текущее состояние
```

```
dbworker.set(dbworker.make_key(message.chat.id,
config.CURRENT_STATE),
config.States.STATE_TAG.value)
        # Сохраняем город
```

```
dbworker.set(dbworker.make_key(message.chat.id,
config.States.STATE_CITY.value), city)
```

```
        bot.send_message(message.chat.id, 'Отлично!  
И, наконец, выскажи мне предпочтения по картинкам:',  
reply_markup=telebot.types.ReplyKeyboardRemove())
```

```
# Обработка введенных тегов и создание демонтажатора  
@bot.message_handler(func=lambda message:  
dbworker.get(dbworker.make_key(message.chat.id,  
config.CURRENT_STATE)) ==  
config.States.STATE_TAG.value)  
def tag(message):  
    tag = message.text  
    site =  
dbworker.get(dbworker.make_key(message.chat.id,  
config.States.STATE_SITE.value))  
    image_url = ""  
    if site == "konachan.net":  
        image_url = konachan(tag)  
    elif site == "deviantart.com":  
        image_url = deviantart(tag)  
    if image_url == "ERROR":  
        # Состояние не изменяется, выводится  
сообщение об ошибке  
        bot.send_message(message.chat.id, 'К  
сожалению, по такому запросу ничего не найдено...')  
        time.sleep(1)  
        bot.send_message(message.chat.id, 'Попробуй  
ввести другой:')  
        return  
    else:  
        city =  
dbworker.get(dbworker.make_key(message.chat.id,  
config.States.STATE_CITY.value))  
        weather = functions.weather(city)  
        try: #отправляем найденную картинку с  
подписью о погоде  
            bot.send_photo(message.chat.id,  
requests.get(image_url).content, caption=weather)  
        except:  
            bot.send_message(message.chat.id, "Что-то  
пошло не так...")  
        # Начинаем сначала
```

```
dbworker.set(dbworker.make_key(message.chat.id,
```

```

config.CURRENT_STATE),
config.States.STATE_SITE.value)
    time.sleep(1)
    bot.send_message(message.chat.id, "А теперь
всё по новой!")
    time.sleep(1)
    markup =
telebot.types.ReplyKeyboardMarkup(resize_keyboard=Tr
ue)
    buttons = ["konachan.net", "deviantart.com"]
    markup.add(*buttons)
    bot.send_message(message.chat.id, "Выбери
сайт, с которого мы возьмём картинку:",
reply_markup=markup)

bot.polling(none_stop=True, interval=0)

```

#### Файл dbworker.py

```

from vedis import Vedis
import config

# Чтение значения
def get(key):
    with Vedis(config.db_file) as db:
        try:
            return db[key].decode()
        except KeyError:
            # в случае ошибки значение по умолчанию -
            начало диалога
            return config.States.STATE_START.value

# Запись значения
def set(key, value):
    with Vedis(config.db_file) as db:
        try:
            db[key] = value
            return True
        except:
            # тут желательно как-то обработать
            ситуацию
            return False

# Создание ключа для записи и чтения

```

```
def make_key(chatid, keyid):
    res = str(chatid) + '_' + str(keyid)
    return res
```

### Файл functions.py

```
from requests import get
import random
import config
import pymorphy2
from rss_parser import Parser
```

```
def konachan(args):
    """поиск картинки по тэгам на сайте
    konachan.net"""
    arg = (str(args).replace(' ', '_')).lower()
    r = get(f'https://konachan.net/post.json?page=1&tags={arg}%20rating:safe&limit=1000')
    json_data = r.json()
    if json_data:
        posts_count = len(json_data)
        try:
            rand = random.randint(0, posts_count - 1)
        except ValueError:
            rand = 0
        try:
            json_data = json_data[rand]
        except TypeError:
            print("ERROR")
        return str(json_data['file_url'])
    else:
        return "ERROR"
```

```
def deviantart(args):
    """поиск картинки по тэгам на deviantart.com"""
    arg = (str(args).replace(' ', '-')).lower()
    try:
        #print("starting parsing")
        xml = get(f'https://backend.deviantart.com/rss.xml?q={arg}')
        parser = Parser(xml=xml.content)
```



```

        feed = parser.parse()
        url_list = []
        for item in feed.feed:
            url_list.append(item.link)
        #print("starting choosing")
        if len(url_list)==0:
            return "ERROR"
        else:
            if len(url_list) == 1:
                r =
get(f'https://backend.deviantart.com/oembed?url={url_list[0]}')
                json_data = r.json()
                return json_data['url']
            else:
                count = 0
                while(True):
                    count += 1
                    if count > 50:
                        break
                    url = url_list[random.randint(0,
len(url_list)-1)]
                    r =
get(f'https://backend.deviantart.com/oembed?url={url}')
                    json_data = r.json()
                    if json_data['safety'] ==
"nonadult":
                        return json_data['url']
                    return "ERROR"
        except:
            return "ERROR"

```

```

def city_check(city: str):
    '''Проверка на существование городов'''
    r =
get(f'https://api.openweathermap.org/data/2.5/weathe
r?q={city}&lang=ru&units=metric&appid={config.weathe
r_api["key"]}')
    json_weather = r.json()
    if json_weather['cod'] != 200:
        if json_weather['message'] == "city not
found":
            return True

```

```

        return False

def weather(city):
    '''Создание описания с погодой'''
    try:
        r =
get(f'https://api.openweathermap.org/data/2.5/weathe
r?q={city}&lang=ru&units=metric&appid={config.weathe
r_api["key"]}')
        json_weather = r.json()
    except:
        return "ERROR"
    '''Склонение названия города'''
    morph = pymorphy2.MorphAnalyzer()
    city_parse = morph.parse(json_weather['name'])[0]
    arr = [city.split]
    if len(arr) > 1:
        tmp_city = ""
        for wrd in arr:
            tmp_city = tmp_city +
morph.parse(wrd)[0].inflect({'loct'}).word.capitalize
e() + " "
        city_parse = tmp_city.rstrip()
    else:
        city_parse =
city_parse.inflect({'loct'}).word.capitalize()
        if '-' in city_parse:
            city_parse = "-".join(list(map(lambda x:
x[0].upper() + x[1:], city_parse.split('-'))))
        if json_weather['main']['temp'] > 0:
            plus = '+'
        else:
            plus = ''
        temperature = "В " + city_parse + " сейчас " +
plus + str(round(json_weather['main']['temp'])) +
"°C"
        weather_description =
json_weather['weather'][0]['description'].capitalize
()
        return str(temperature + "\n" +
weather_description)

```

Файл **tdd.py**

```

import unittest
from functions import *

class Tests(unittest.TestCase):
    def setUp(self):
        self.city_correct = "Moscow"
        self.city_uncorrect = "Мазква"
        self.normal_request = "cat"
        self.weird_request =
"fgycsynvuyqcu8947qwyyfHuank"
    def test_city_check(self):
        self.assertFalse(city_check(self.city_correct))
        self.assertTrue(city_check(self.city_uncorrect))
    def test_wrong_image_request(self):
        self.assertNotEqual(konachan(self.normal_request),
"ERROR")
        self.assertEqual(konachan(self.weird_request), "ERROR")
        self.assertEqual(deviantart(self.weird_request),
"ERROR")

if __name__ == "__main__":
    unittest.main()

```

### Файл bdd.py

```

from behave import given, when, then
from functions import *
@given(u'An existing city')
def step_impl(context):
    context.city = "Moscow"
@given(u'A non-existing city')
def step_impl(context):
    context.city = "Мазква"
@when(u'Trying to check it')
def step_impl(context):
    context.result = str(city_check(context.city))
@given(u'A normal image request')
def step_impl(context):
    context.image_request = "cat"
@given(u'A weird image request')
def step_impl(context):
    context.image_request =
"fgycsynvuyqcu8947qwyyfHuank"
@when(u'Finding image')
def step_impl(context):

```

```

    context.result = konachan(context.image_request)
@then(u'The next result is expected: "{result}"')
def step_impl(context, result: str):
    assert context.result == result
@then(u'We expect to receive an image')
def step_impl(context):
    assert context.result != "ERROR"

```

Файл bdd.feature

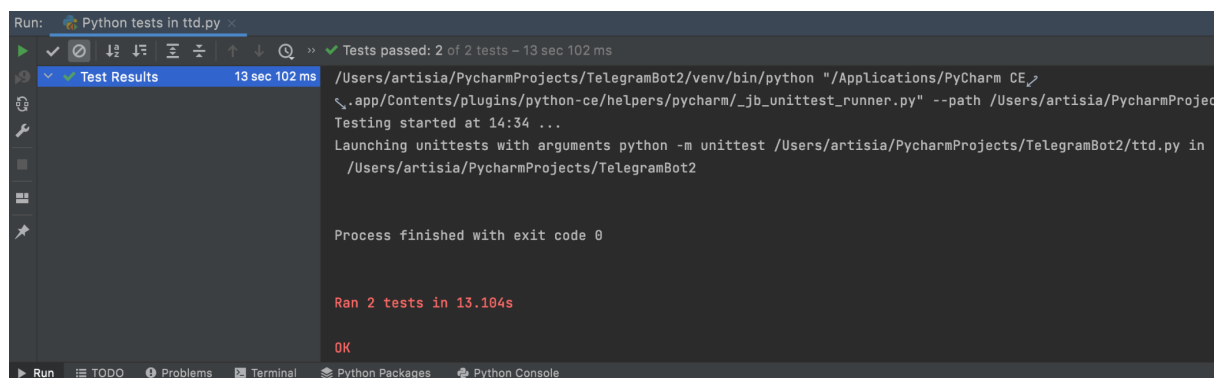
```

Feature: Functions checking
  Scenario: Checking an existing city
    Given An existing city
    When Trying to check it
    Then The next result is expected: "False"
  Scenario: Checking a non-existing city
    Given A non-existing city
    When Trying to check it
    Then The next result is expected: "True"
  Scenario: Sending a normal image request
    Given A normal image request
    When Finding image
    Then We expect to receive an image
  Scenario: Sending a weird image request
    Given A weird image request
    When Finding image
    Then The next result is expected: "ERROR"

```

**Экранные формы с модульным тестированием:**

**Результат tdd-тестирования**



**Результат bdd-тестирования**

```
Terminal: Local x + v
(venv) artisias@MacBook-Air-Ulia TelegramBot2 % behave features/bdd.feature
Feature: Functions checking # features/bdd.feature:1

Scenario: Checking an existing city # features/bdd.feature:2
  Given An existing city # features/steps/bdd.py:4 0.002s
  When Trying to check it # features/steps/bdd.py:10 0.273s
  Then The next result is expected: "False" # features/steps/bdd.py:22 0.000s

Scenario: Checking a non-existing city # features/bdd.feature:6
  Given A non-existing city # features/steps/bdd.py:7 0.000s
  When Trying to check it # features/steps/bdd.py:10 0.247s
  Then The next result is expected: "True" # features/steps/bdd.py:22 0.001s

Scenario: Sending a normal image request # features/bdd.feature:10
  Given A normal image request # features/steps/bdd.py:13 0.000s
  When Finding image # features/steps/bdd.py:19 2.437s
  Then We expect to receive an image # features/steps/bdd.py:25 0.000s

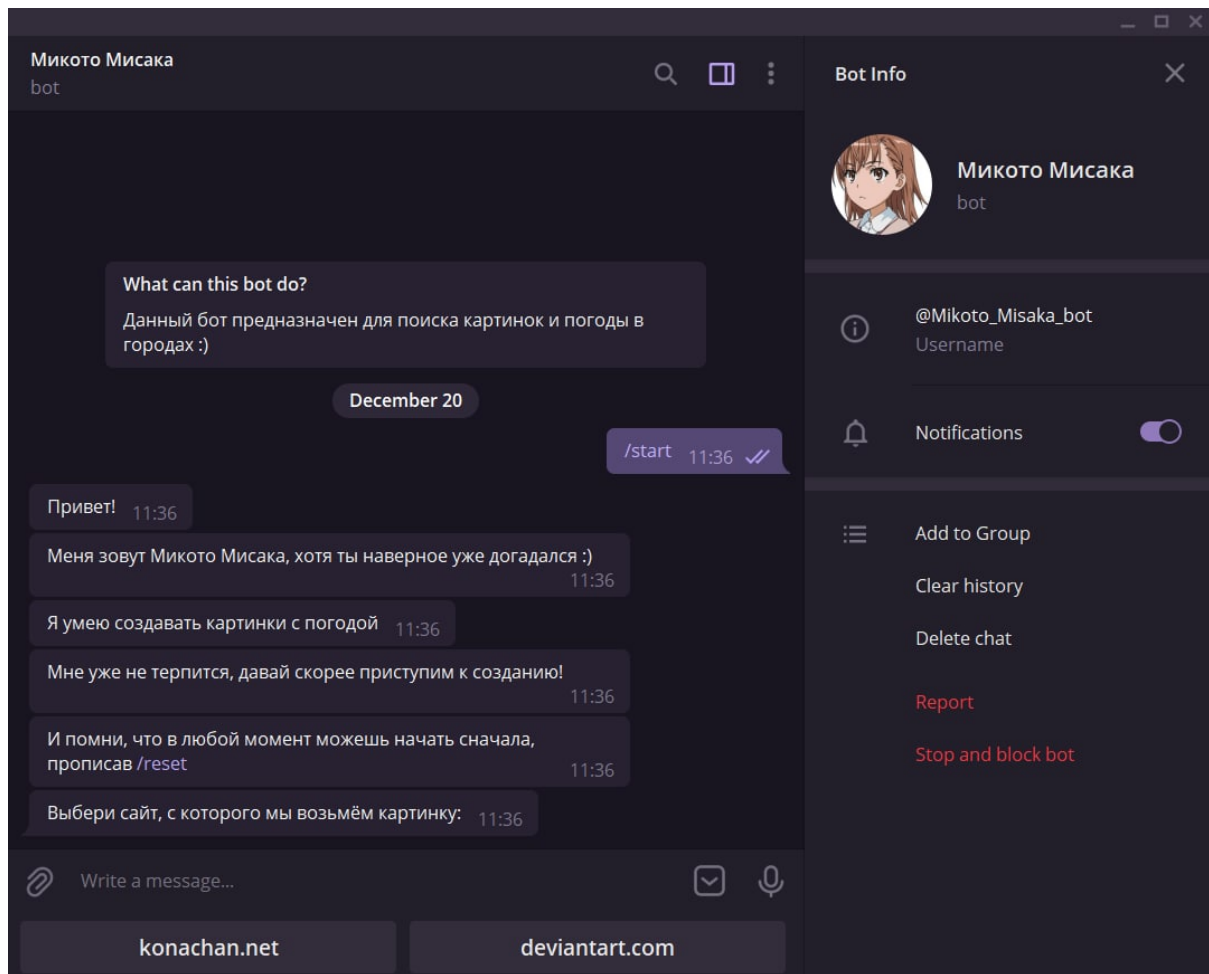
Scenario: Sending a weird image request # features/bdd.feature:14
  Given A weird image request # features/steps/bdd.py:16 0.000s
  When Finding image # features/steps/bdd.py:19 0.279s
  Then The next result is expected: "ERROR" # features/steps/bdd.py:22 0.000s

1 feature passed, 0 failed, 0 skipped
4 scenarios passed, 0 failed, 0 skipped
12 steps passed, 0 failed, 0 skipped, 0 undefined
Took 0m3.240s
(venv) artisias@MacBook-Air-Ulia TelegramBot2 %
```

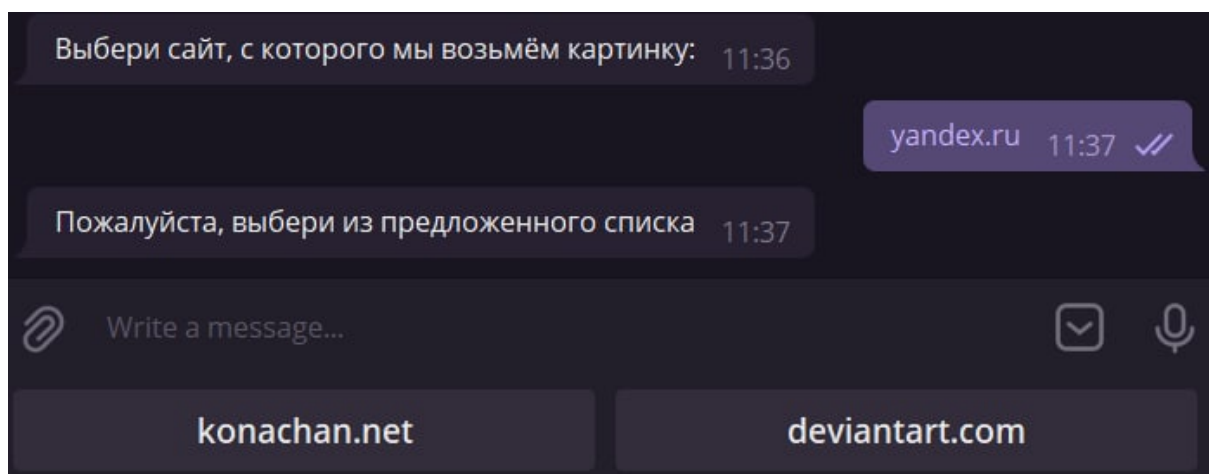
Run | TODO | Problems | Terminal | Python Packages | Python Console

## Экранные формы с примерами работы бота:

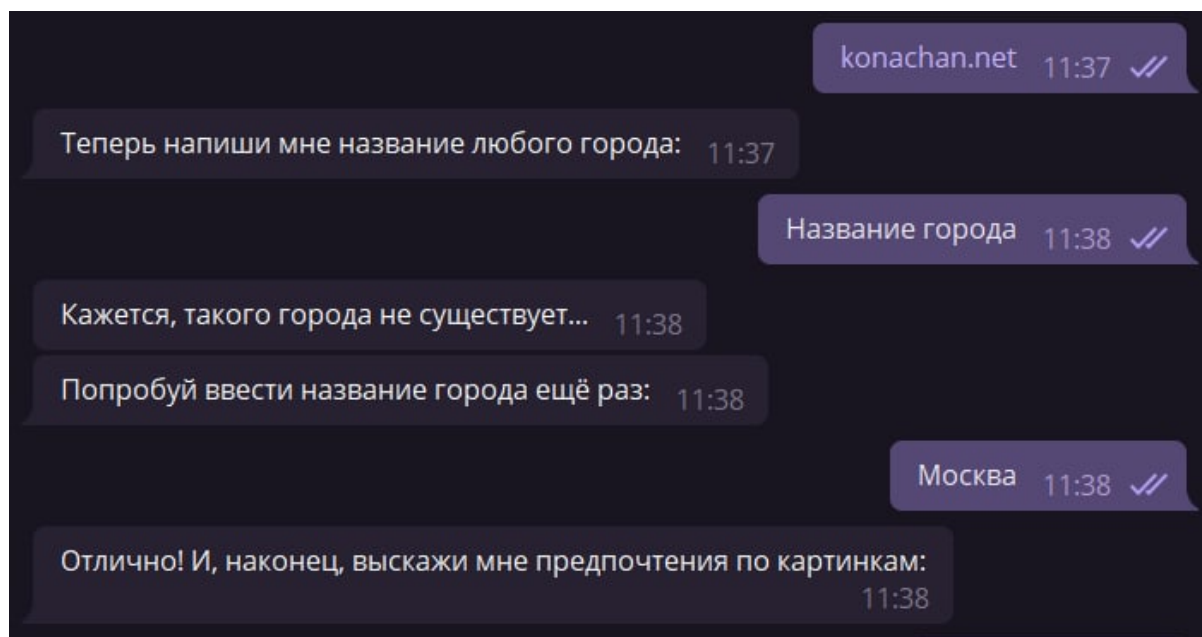
Начало работы:



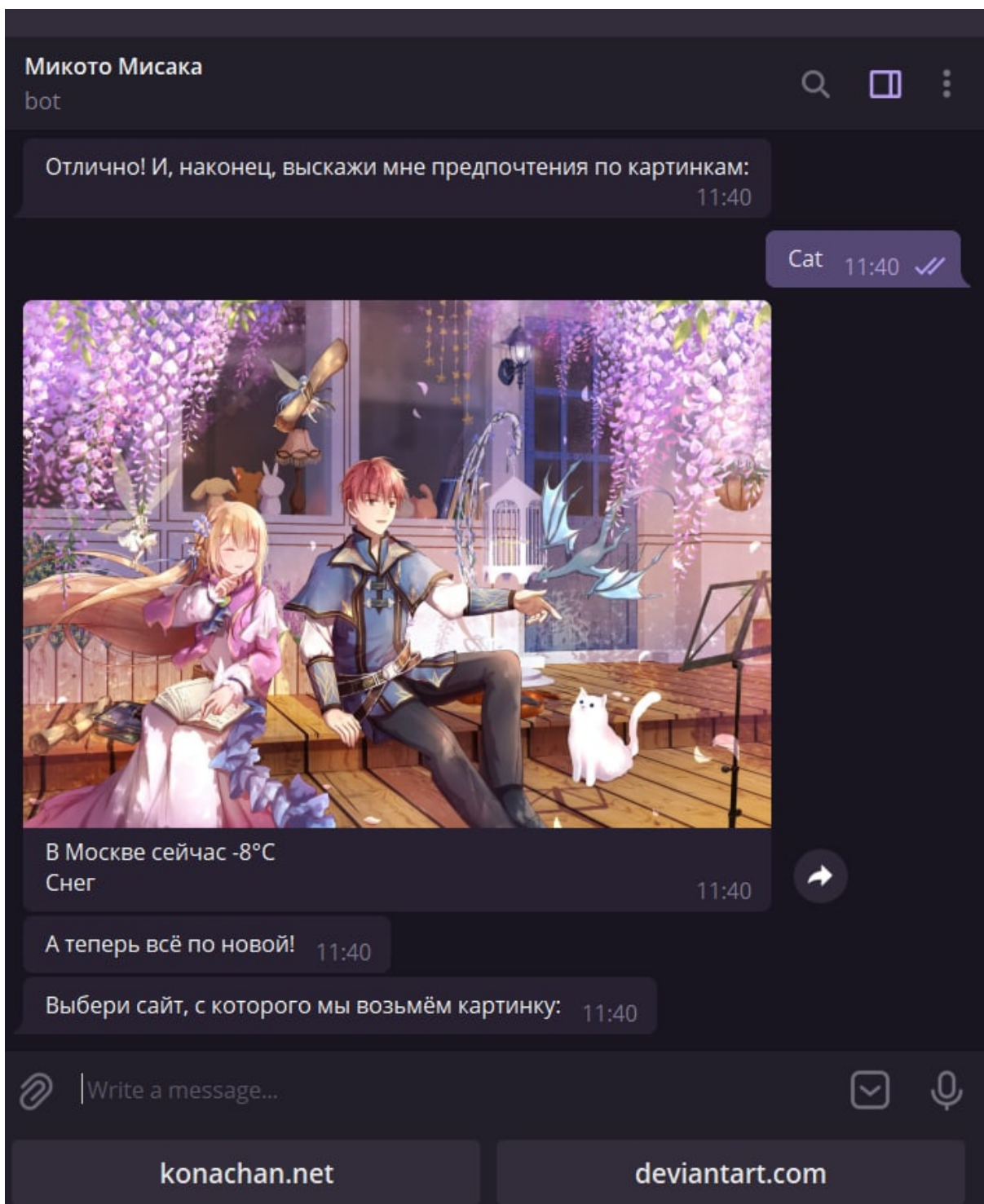
Ввод сайта не из списка:



Ввод неверного названия города:



Генерация картинки:



Примеры работы с другими параметрами:



А теперь всё по новой! 11:40

Выбери сайт, с которого мы возьмём картинку: 11:40

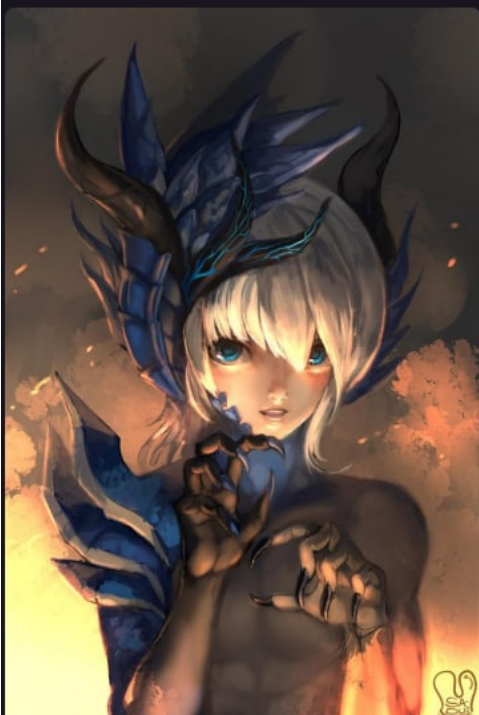
deviantart.com 11:41 ✓✓

Теперь напиши мне название любого города: 11:41

New York 11:41 ✓✓

Отлично! И, наконец, выскажи мне предпочтения по картинкам: 11:41

Dragon 11:41 ✓✓



В Нью-Йорке сейчас -3°C  
Ясно

11:44



Выбери сайт, с которого мы возьмём картинку: 11:44

konachan.net 11:45 ✓✓

Теперь напиши мне название любого города: 11:45

Ростов-на-Дону 11:45 ✓✓

Отлично! И, наконец, выскажи мне предпочтения по картинкам: 11:45

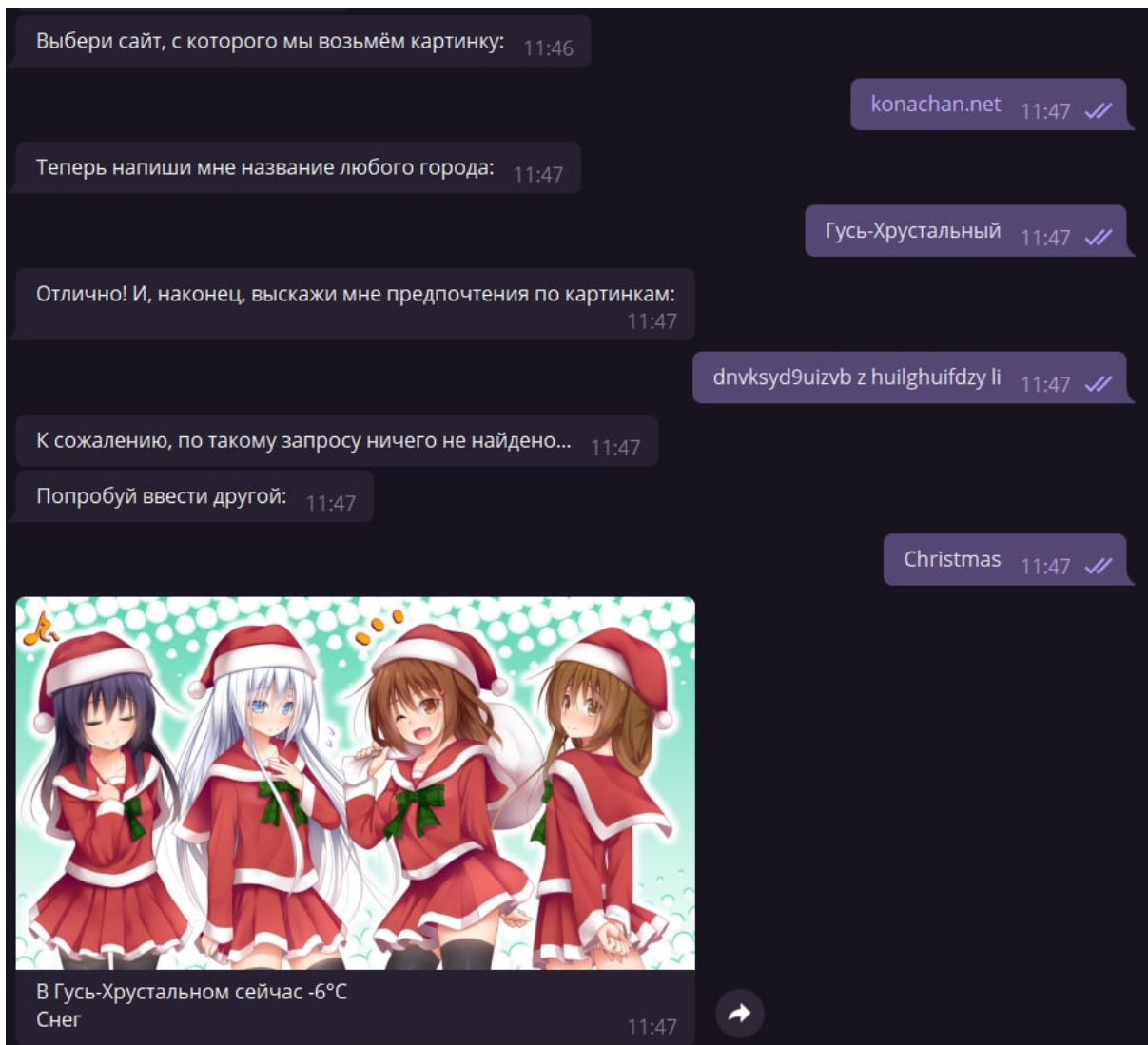
frog 11:45 ✓✓



В Ростове-На-Дону сейчас +4°C  
Пасмурно

11:45





Сброс функцией /reset:

