

Министерство образования и науки РФ  
Санкт-Петербургский политехнический университет Петра Великого  
Институт компьютерных наук и технологий  
Высшая школа программной инженерии

**Курсовая работа**  
по дисциплине “Конструирование программного обеспечения”

Выполнили:  
студенты гр. 3530904/80105

Малинин И. И.  
Пинаев Н.Д.

Преподаватель:

Иванов А.С.

Санкт-Петербург  
2020

## Проблема

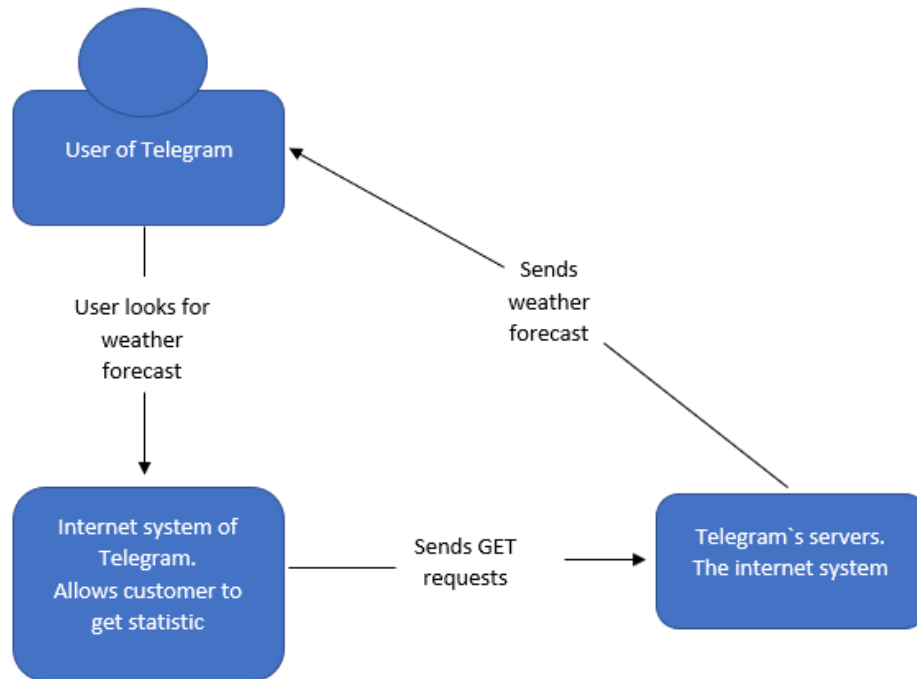
Приложение решает проблему доступа пользователя к актуальному прогнозу погоды, который необходим для эффективного планирования своей жизнедеятельности.

## Требования проекта

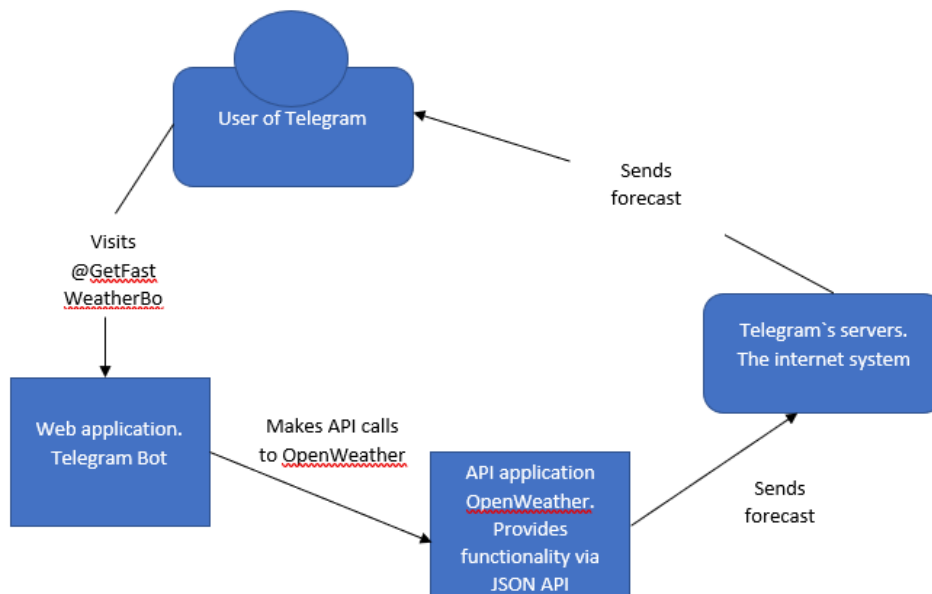
№	Формулировка	Тип
1	Используемый мессенджер — Telegram	Нефункциональное
2	Используемый язык — английский	Функциональное
3	Предоставление прогноза погоды в выбранном городе на текущий момент	Функциональное
4	Предоставление прогноза погоды в выбранном городе на следующий день	Функциональное
5	Предоставление прогноза погоды в выбранном городе на неделю	Функциональное

## Диаграммы

1. Level 1: A System Context diagram provides a starting point, showing how the software system in scope fits into the world around it.



2. Level 2: A Container diagram zooms into the software system in scope, showing the high-level technical building blocks.



## Тестирование

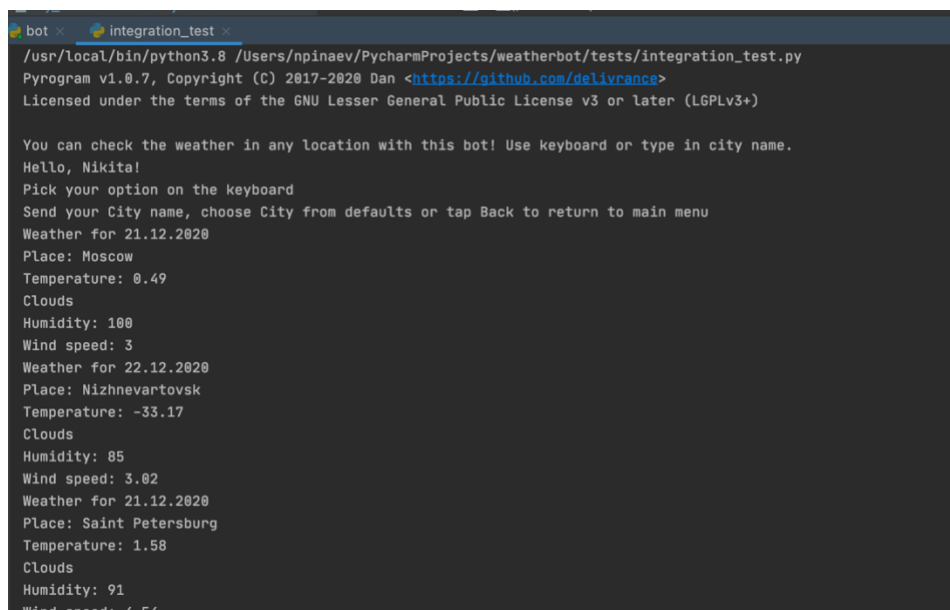
Папка tests содержит интеграционный тест, написанный с использованием библиотеки для тестирования tgintegration и pyrogram — фреймворка, который действует как полноценный клиент Telegram на основе MTProto.

Тестируются все доступные команды. Проверяется, что в ответ от бота пользователь получает нужное количество сообщений (одно либо два в случае команды /start), а также контекст сообщения, например наличие температуры в ответе. Мы произвели полный проход сценария пользователя.

Тестовые сообщения отправляются с аккаунта, api\_id и api\_hash которого указаны в конфигурационном файле (из API Development tools на <https://my.telegram.org/>).

Юнит тесты проверяют корректность работы функций обработки данных и базового класса. Мы использовали фреймворк pytest для запуска тестов.

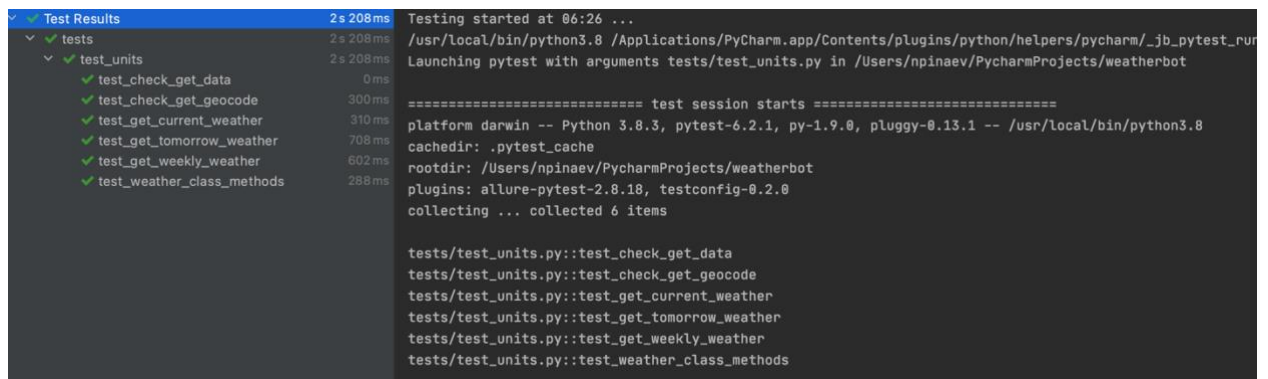
Работа интеграционного теста:



```
bot x integration_test x
/usr/local/bin/python3.8 /Users/npinaev/PycharmProjects/weatherbot/tests/integration_test.py
Pyrogram v1.0.7, Copyright (C) 2017-2020 Dan <https://github.com/delivrance>
Licensed under the terms of the GNU Lesser General Public License v3 or later (LGPLv3+)

You can check the weather in any location with this bot! Use keyboard or type in city name.
Hello, Nikita!
Pick your option on the keyboard
Send your City name, choose City from defaults or tap Back to return to main menu
Weather for 21.12.2020
Place: Moscow
Temperature: 0.49
Clouds
Humidity: 100
Wind speed: 3
Weather for 22.12.2020
Place: Nizhnevartovsk
Temperature: -33.17
Clouds
Humidity: 85
Wind speed: 3.02
Weather for 21.12.2020
Place: Saint Petersburg
Temperature: 1.58
Clouds
Humidity: 91
Wind speed: 4.56
```

## Работа юнит тестов:



The screenshot shows the PyCharm interface with the Test Results window on the left and the Terminal on the right. The Test Results window displays a tree view of test results for 'test\_units', showing sub-tests like 'test\_check\_get\_data', 'test\_check\_get\_geocode', 'test\_get\_current\_weather', 'test\_get\_tomorrow\_weather', 'test\_get\_weekly\_weather', and 'test\_weather\_class\_methods' with their respective execution times. The Terminal window shows the output of the pytest command, including the test session start, platform information, and the list of collected items.

```
Testing started at 06:26 ...
/usr/local/bin/python3.8 /Applications/PyCharm.app/Contents/plugins/python/helpers/pycharm/_jb_pytest_runner.py
Launching pytest with arguments tests/test_units.py in /Users/npinaev/PycharmProjects/weatherbot

===== test session starts =====
platform darwin -- Python 3.8.3, pytest-6.2.1, py-1.9.0, pluggy-0.13.1 -- /usr/local/bin/python3.8
cachedir: .pytest_cache
rootdir: /Users/npinaev/PycharmProjects/weatherbot
plugins: allure-pytest-2.8.18, testconfig-0.2.0
collecting ... collected 6 items

tests/test_units.py::test_check_get_data
tests/test_units.py::test_check_get_geocode
tests/test_units.py::test_get_current_weather
tests/test_units.py::test_get_tomorrow_weather
tests/test_units.py::test_get_weekly_weather
tests/test_units.py::test_weather_class_methods
```

## Сборка

1. Устанавливаем зависимости `pip install -r requirements.txt`. Нужен python 3.7+ и pip
2. Запускаем бота `python3 main.py`
3. Запускаем интеграционный тест

```
cd tests
```

```
python3 integration_test.py
```

Он работает только если запущен бот. Мы использовали Фреймворк pyrogram, он работает как полноценный клиент Telegram, если заменить в конфиге `api_id` и `api_hash` на свои можно проследить прохождение теста. В консоль выводятся полученные данные.

4. Запускаем интеграционный тест

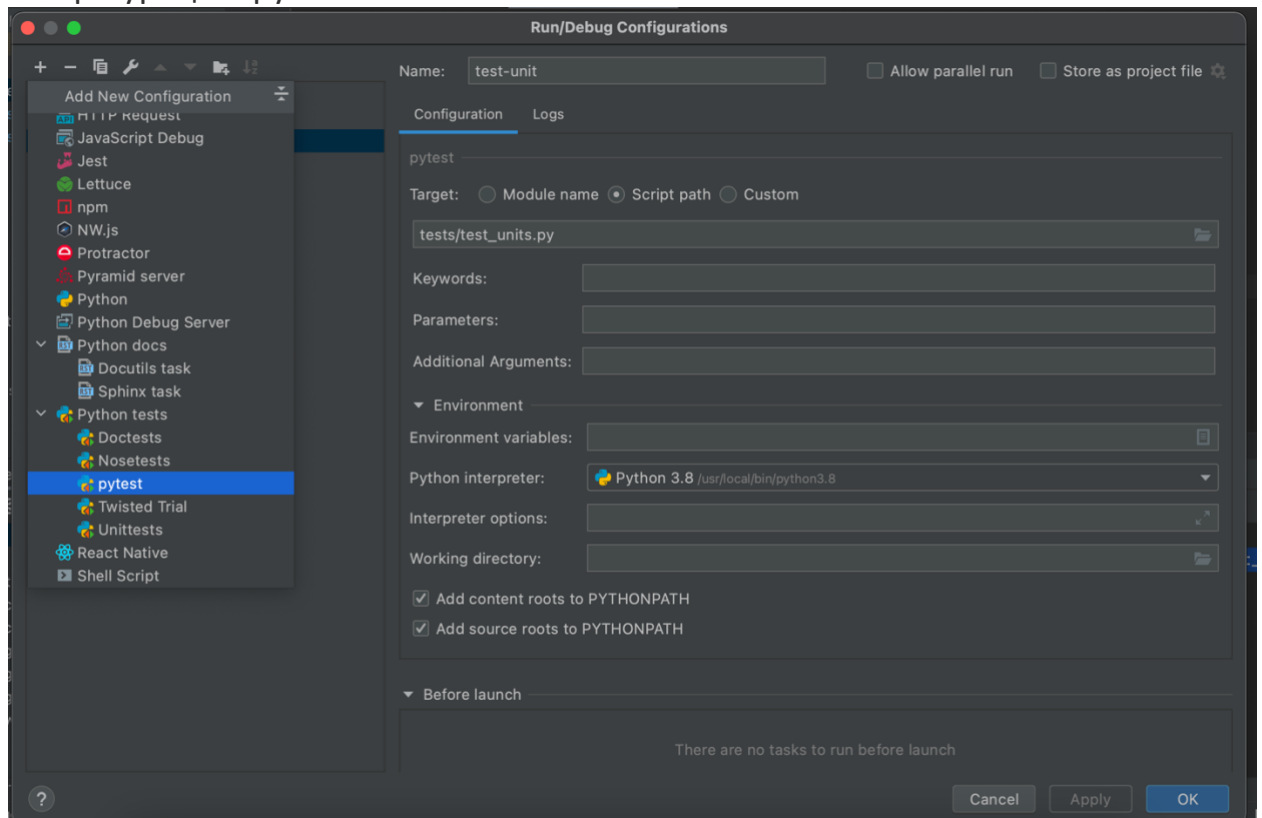
```
cd tests
```

```
python3 /"your_path_to_pytest_runner"/_jb_pytest_runner.py --path
test_units.py
```

У меня это выглядит так:

```
/usr/local/bin/python3.8
/Applications/PyCharm.app/Contents/plugins/python/helpers/pycharm/_jb_pytest_runner.py --
path tests/test_units.py
```

Но есть способ запустить pytest гораздо легче, для этого надо добавить конфигурацию pytest

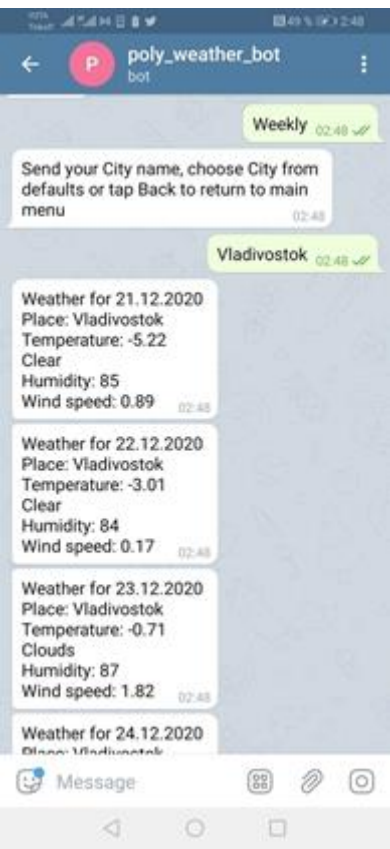
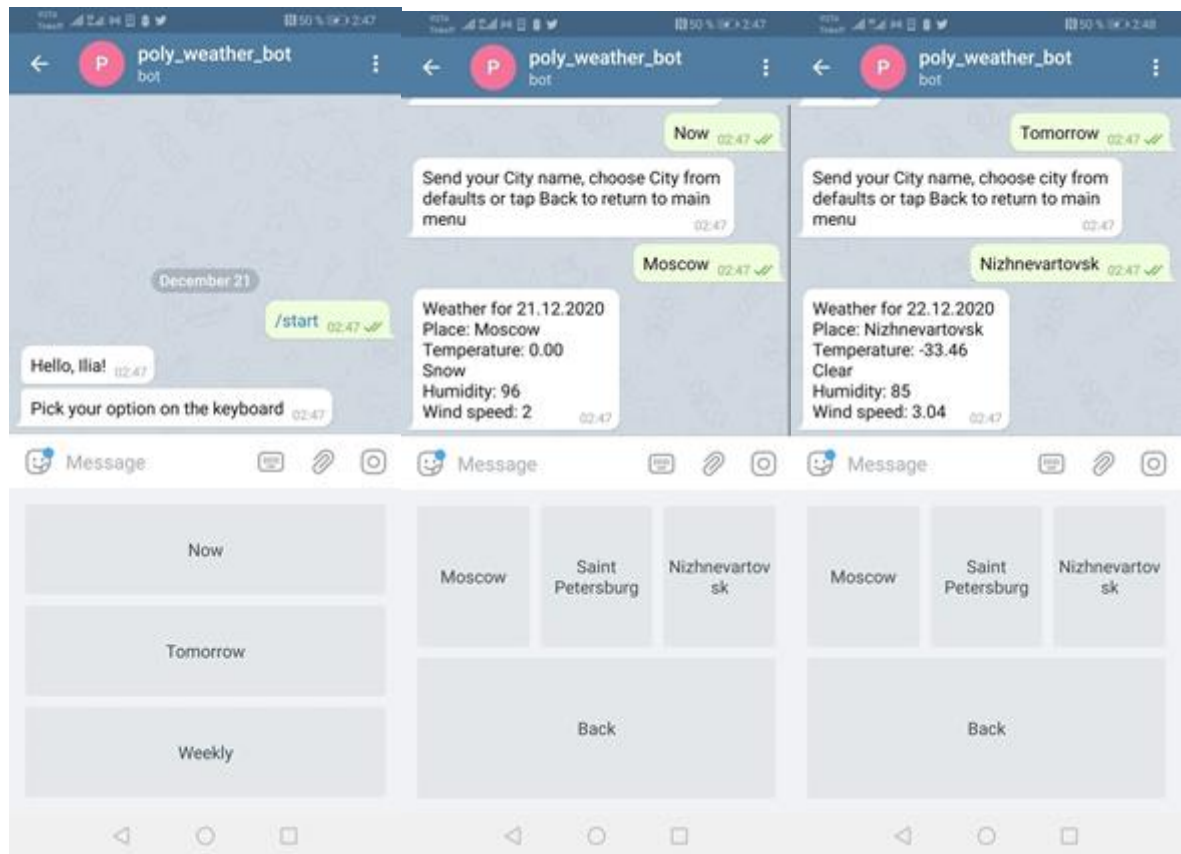


Add new configuration -> pytest -> add target tests/test\_units.py

## Github

<https://github.com/imlInn/weatherbot>

Скриншоты



## **Вывод**

В ходе курсовой работы были пройдены все этапы разработки качественного программного обеспечения. В итоге был разработан бот, позволяющий пользователю получить данные о прогнозе погоды в выбранном им городе и выбранный пользователем временной период.