



# Контрольная работа на тему: «Разработка бота для мессенджера Telegram»

Выполнил: студент группы 15ИС16п Максимов Я.А.

Руководитель: ст. преподаватель кафедры ИТС Володин К.И.

# План

1. Цели и задачи
2. Техническое задание
3. UML диаграммы
4. Код
5. Содержание проекта
6. Скриншоты
7. Демонстрация работы
8. Инструментарий
9. Выводы по проделанной работе

# Цели

Цель контрольной работы – разработать приложение (бот) для мессенджера **Telegram** которое будет обладать следующими функциями:

- Вывод информации о погоде по запросу в любой точке мира
- Отображение глобальных и региональных новостей

# Задачи

Перед разработкой бота были поставлены следующие задачи:

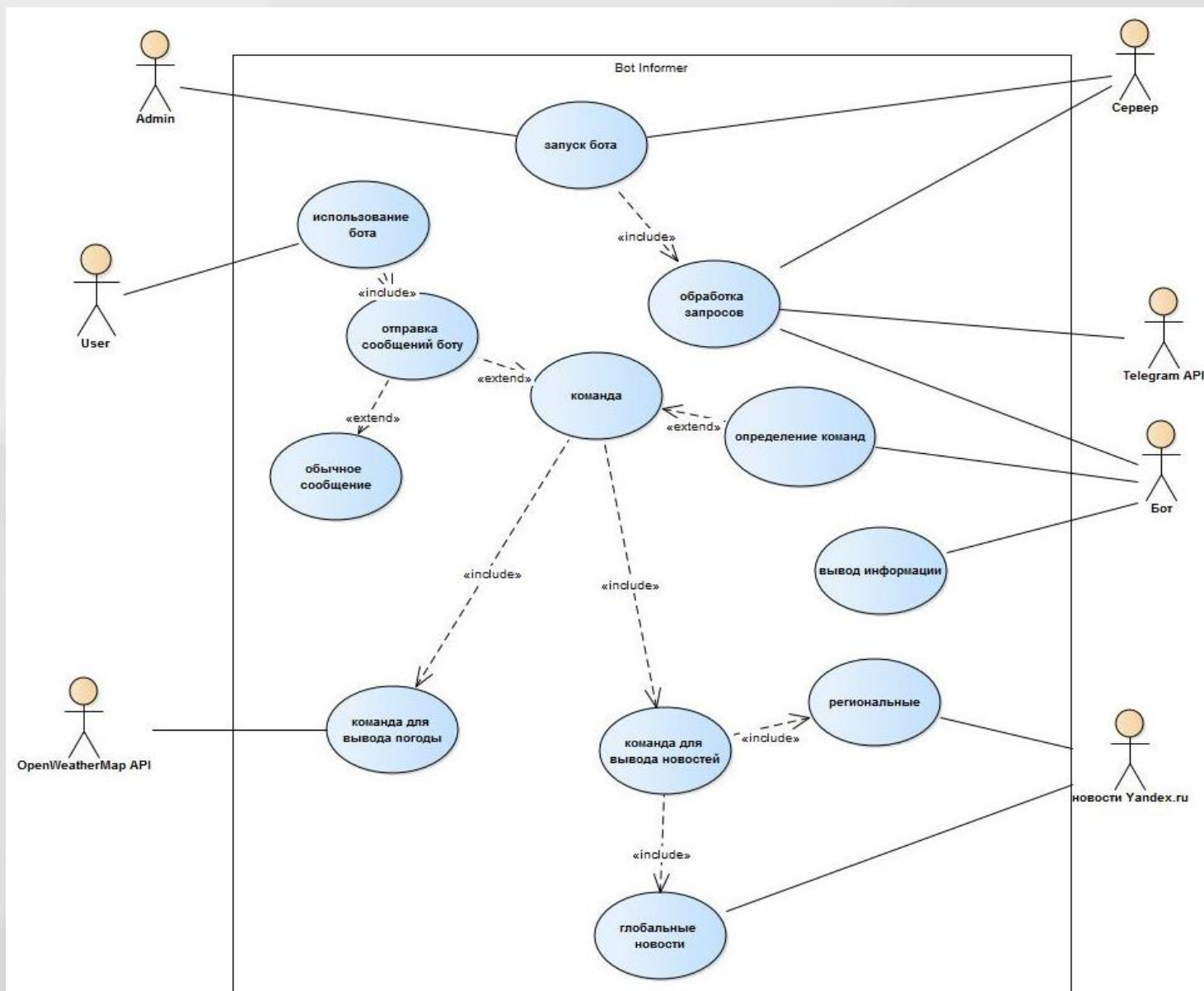
- создание бота в Telegram и получение токена
  - разбор Telegram API
  - разработка структуры бота
- продумать реализацию алгоритмов
  - написание кода

# Функционал

Приложение должно обладать следующим функционалом:

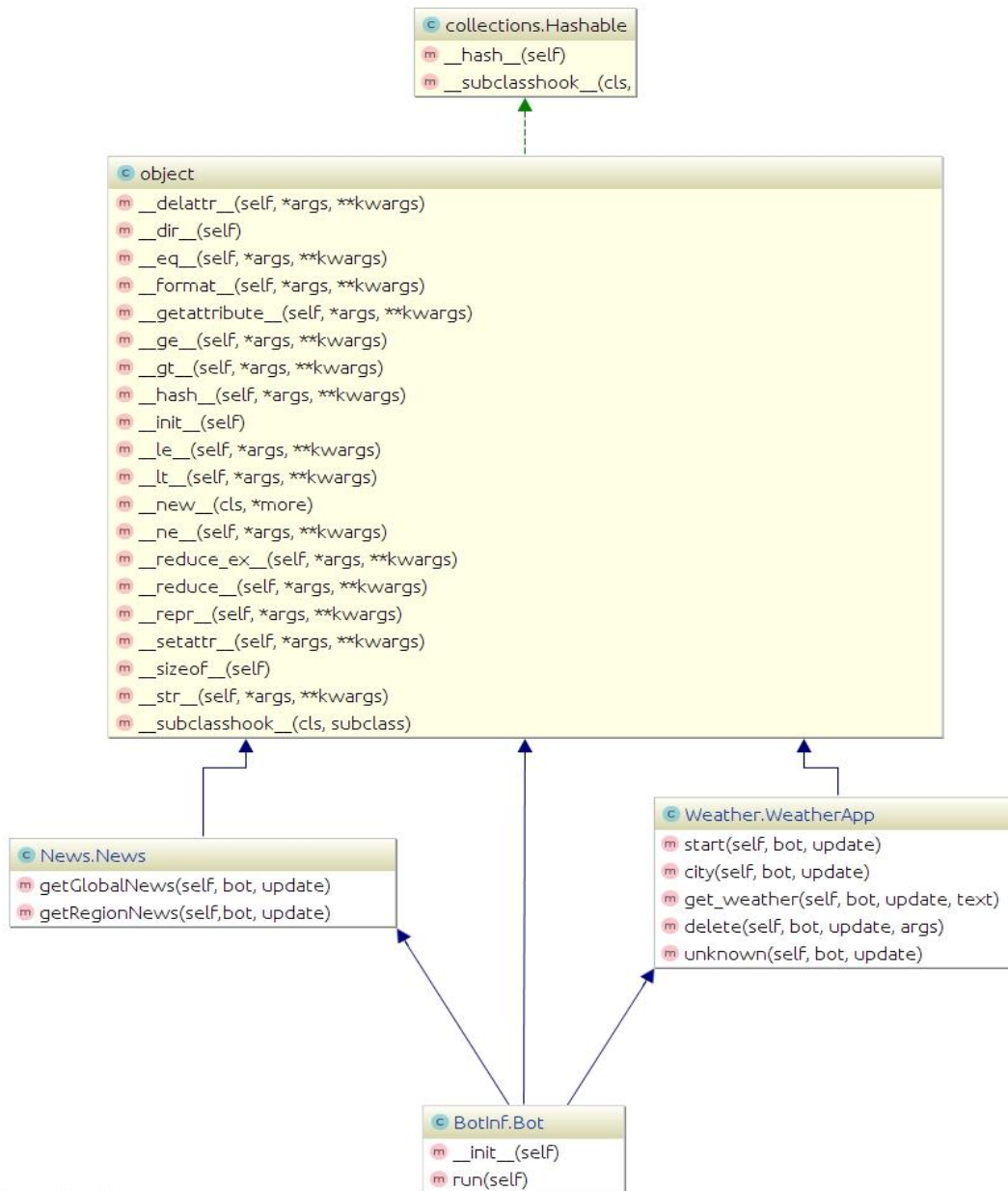
- отображение региональных новостей
- отображение глобальных новостей
- отображение погоды

# Диаграмма вариантов использования

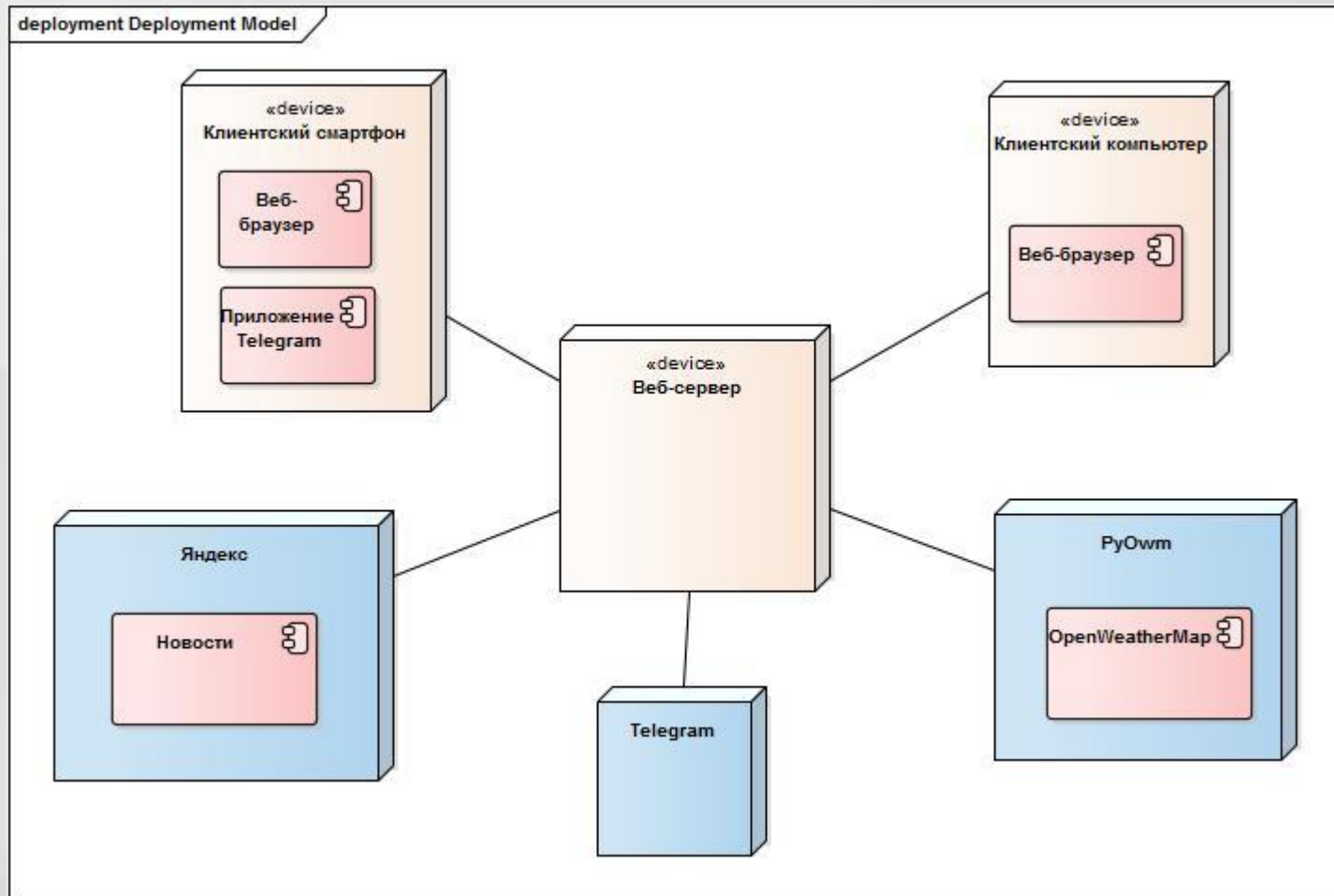




# Диаграмма классов



# Диаграмма развертывания





# Исходный код (App)

```
import Weather, News, Config, Info
import telegram, logging, pyowm
from telegram import Update
from telegram.ext import Updater, Filters
from mhandler import MessageHandler
from chandler import CommandHandler

class Bot:
    def __init__(self):
        self.updater = Updater(token = Config.BOT_TOKEN)
        # root = logging.getLogger()
        # root.setLevel(logging.INFO)
        # logging.basicConfig(format = '%(asctime)s - %(name)s - %(levelname)s - %(message)s')
        # logger = logging.getLogger(__name__)

    @bot.message_handler(content_types=["text"])
    # def feedback(message):
    #     bot.send_message(message.chat.id, message.text)

    def run(self): # данный метод обеспечивает запуск и работу бота
        dp = self.updater.dispatcher
        dp.add_handler(CommandHandler("start", Info.Information.getAbout))
        dp.add_handler(CommandHandler("rnews", News.News.getRegionNews))
        dp.add_handler(CommandHandler("gnews", News.News.getGlobalNews))
        dp.add_handler(CommandHandler("weather", Weather.WeatherApp.start))
        dp.add_handler(MessageHandler([Filters.text], Weather.WeatherApp.city))
        dp.add_handler(CommandHandler('delete', Weather.WeatherApp.delete, pass_args=True))
        dp.add_handler(MessageHandler([Filters.command], Weather.WeatherApp.unknown))
        self.updater.start_polling()
        self.updater.idle()
```

# Исходный код (News)

```
import requests
from bs4 import BeautifulSoup

class News: #класс новостей

    def getGlobalNews(self, bot, update): # получить новости из мира
        r = requests.get("http://yandex.ru")
        soup = BeautifulSoup(r.text, "html.parser")
        news = soup.find_all(class_="link list_item-content link_black_yes")
        for new_cur in news[0:5]:
            news_list = (new_cur.get("aria-label"))
            bot.sendMessage(chat_id=update.message.chat_id, text="• " + news_list)

    def getRegionNews(self, bot, update): # получить региональные новости
        r = requests.get("http://yandex.ru")
        soup = BeautifulSoup(r.text, "html.parser")
        news = soup.find_all(class_="link list_item-content link_black_yes")
        for new_cur in news[5:10]:
            region_list = new_cur.get("aria-label")
            bot.sendMessage(chat_id=update.message.chat_id, text="• " + region_list)
```

# Исходный код (News)

```
import requests
from bs4 import BeautifulSoup

class News: #класс новостей

    def getGlobalNews(self, bot, update): # получить новости из мира
        r = requests.get("http://yandex.ru")
        soup = BeautifulSoup(r.text, "html.parser")
        news = soup.find_all(class_="link list_item-content link_black_yes")
        for new_cur in news[0:5]:
            news_list = (new_cur.get("aria-label"))
            bot.sendMessage(chat_id=update.message.chat_id, text="• " + news_list)

    def getRegionNews(self, bot, update): # получить региональные новости
        r = requests.get("http://yandex.ru")
        soup = BeautifulSoup(r.text, "html.parser")
        news = soup.find_all(class_="link list_item-content link_black_yes")
        for new_cur in news[5:10]:
            region_list = new_cur.get("aria-label")
            bot.sendMessage(chat_id=update.message.chat_id, text="• " + region_list)
```



# Исходный код (Weather)

```
import BotInf, Config, Info
import telegram, pyowm
from pyowm import OWM

owm = OWM(Config.WEATHER_TOKEN, language='ru')
# Скрытие клавиатуры по ненадобности
hide_markup = telegram.ReplyKeyboardHide()

# Список городов в списке(основная клавиатура)
towns = []
town_keyboard = [['Помощь'], towns]
town_markup = telegram.ReplyKeyboardMarkup(town_keyboard, resize_keyboard = True)

# Список городов для их удаления из основного списка
remove_towns = []
remove_keyboard = [remove_towns]
remove_markup = telegram.ReplyKeyboardMarkup(remove_keyboard, resize_keyboard = True)

class WeatherApp: # погода
    # запуск работы с погодой
    def start(self, bot, update):
        bot.sendMessage(chat_id = update.message.chat_id, text = "Введите город для сохранения", reply_markup = town_markup)

    # Функция отвечающая заопределение команд которые вводит пользователь боту
    def city(self, bot, update):
        message = update.message
        chat_id = message.chat_id
        text = message.text

        if text == "Remove city":
            bot.sendMessage(chat_id = chat_id, text = "Выберите город который хотите удалить", reply_markup = remove_markup)
        elif text == "Помощь":
            bot.sendMessage(chat_id = chat_id, text = Info.Information.getAbout(self, bot, update), reply_markup = town_markup)
        else:
            if len(towns) == 0:
                towns.append(text)
                remove_towns.append("/delete " + text)
                town_keyboard.append(["Remove city"])
                town_keyboard.remove(["Помощь"])
                bot.sendMessage(chat_id = chat_id, text = "Сохранили", reply_markup = town_markup)
            elif len(towns) == 1:
```

# Скриншоты работы



Демо



Telegram

Welcome to the era of fast and secure  
messaging

Start Messaging



# Инструментарий

- Язык программирования: Python
  - IDE: [Pycharm](#)
- Библиотека для работы с TelegramAPI: [TeleBot](#)
- Библиотека для работы с информацией о погоде: [PyOwm](#)
- Предоставление новостей: [Yandex.ru](#)
  - BeautifulSoup 4

# Выводы:

В результате проделанной контрольной работы был разработан бот для мессенджера Telegram, который обладает следующим функционалом:

- отображение региональных новостей
- отображение глобальных новостей
  - отображение погоды



# Спасибо за внимание

Контакты:

E-mail : sHarveyWilliams@gmail.ru

Телефон : +79631041100

<https://github.com/sHarveyWilliams/BotInf>