 МІНІСТЕРСТВО ОСВІТИ I НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ   
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

ФАКУЛЬТЕТ БІОМЕДИЧНОЇ ІНЖЕНЕРІЇ

КАФЕДРА БІОМЕДИЧНОЇ КІБЕРНЕТИКИ

**Розрахунково-графічна робота**

з дисципліни «Веб-технології та веб-дизайн-2.»

Варіант №13

**Виконав:**

студент гр. БС-83

Розмариця О.А.

**Перевірив:**

ас. Давидько О. Б.,

ас. Матвійчук О. В.

Зараховано від \_\_\_.\_\_\_.\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис викладача)

Київ-2020

1. **Вибір та опис специфікації проекту.**

Додатком є онлайн бот для телеграму - онлайн калькулятор для геометричного круга (розрахунок площі, радіусу, діаметру і його довжини по заданими даними).

Користувач буде вибирати круг, вводити доступні йому параметри і вибирати, який параметр йому потрібно знайти (наприклад, площу круга по його радіусу). У відповідь бот має присилати рішення. Буде можливість написати адміністратору.

Кількість користувачів - може стати великою в робочий час, якщо такий бот буде популяризовуватись. Можливості downtime - так як це бот, він має працювати цілодобово, але вимикати його можна краще всього вночі через мінімальну кількість можливих користувачів. інтеграції з іншими сервісами - Це бот для додатку "Telegram" на комп'ютери та смартфони, тому інтеграція з додатком "Telegram". Безпека - боту не будуть надаватись будь-які персональні дані, тому безпека буде на "плечах" додатку "Telegram". Вимоги до тестування - тестування введених даних.

**2. Вибір та обґрунтування технологій для реалізації веб сервісу.**

Вибираю Python, бо це високорівнева мова програмування, яку можна застосовувати і для створення прикладних програм, і для WEB розробки. Продуктивність платформи вельми висока, код відрізняється простою і читабельністю.

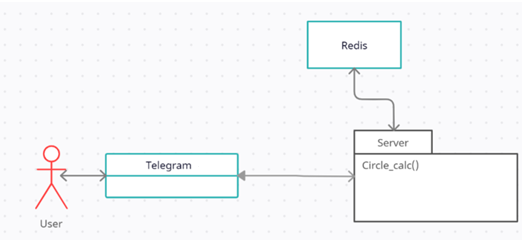
В якості фреймворка був використаний PyTelegramBotApi через його зручність в використанні і зв'язок з Telegram.

Global Interpreter Lock не дозволяє в одному інтерпретаторі Python ефективно використовувати більше одного потоку. Захисники GIL стверджують, що однопоточні програми при наявності GIL працюють набагатое фективніше. Але наявність GIL означає, що паралельні обчислення з використанням безлічі потоків і загальної пам'яті неможливі. А це досить сильне обмеження в сучасному багатоядерному світі. Щоб використати багатопоточність, необхідно використати модуль multiprocessing.

**3. Опис архітектурних рішень що використовувались в ході розробки проекту**

Додаток реалізований монолітом. Цей вибір обумовлений тим, що додаток не буде масштабним. Він не має великої кількості інтеграцій. І через реалізацією однією людиною та доступних їй ресурсів він не буде масштабуватись

UML - діаграма:



4. **Розгортання програмного продукту**

● Скрипт контейнеризації сервісу (dockerfile)

FROM python:3.9.0

ADD . /src

WORKDIR /src

RUN pip install -r requirements.txt

CMD python server\_part.py

● Скрипт оркестрації сервісів (docker-compose.yml)

version: '3.8'

services:

web:

build: .

volumes:

- .:/src

depends\_on:

- redis\_service

env\_file:

- redis.env

redis\_service:

image: redis

ports:

- "6379:6379"

**Висновки:**

Телеграм-бота простіше всього реалізувати на Python через PyTelegramBotApi, тому вибір фреймворка я вважаю, вірний. Основний сервер реалізований одним файлом через його невеликий розмір (в плані кількості коду). Додаток був розгорнутий через Docker та без нього на ОС Windows 7 та 10. Маштабованість не була протестована через відсутність користувачів. Додаток навряд можна назвати маштабованим через неоптимізований код.

**Додаток 1(лістинг програмного коду з найважливішим функціоналом)**

import telebot

from telebot import types

from requests import get

from PIL import Image

from math import sqrt

from redis import Redis

import redis

import time

import os

handle = open("token.txt", "r")

token = handle.readline()

bot = telebot.TeleBot(token)

r = Redis(host=os.environ['REDIS\_HOST'], port=os.environ['REDIS\_PORT'])

#os.environ['REDIS\_HOST'] = '127.0.0.1'

#os.environ['REDIS\_PORT'] = '6379'

redis.REDIS\_PORT=6379

figure\_chose\_markup = telebot.types.ReplyKeyboardMarkup(True, True)

figure\_chose\_markup.row('Треугольник')

figure\_chose\_markup.row('Круг')

figure\_chose\_markup.row('Квадрат')

@bot.message\_handler(commands=['start'])

def start\_message(message):

bot.send\_message(message.chat.id,

'Привет, это бот для вычисления параметров геометрических фигур. Чтобы начать, выберите фигуру или напишите нам сообщение, мы обязательно на него ответим:'

,reply\_markup = figure\_chose\_markup)

# Написать помощнику

#@bot.message\_handler(commands=['admin'])

#def default\_test(message):

#keyboard = types.InlineKeyboardMarkup()

#url\_button = types.InlineKeyboardButton(text="Написать помощнику", url="https://t.me/Vladimir\_dolar")

#keyboard.add(url\_button)

#bot.send\_message(message.chat.id, "Эта кнопка поможет Вам связатся с помощником в решении Ваших задач:", reply\_markup=keyboard)

#Rediska.asd(message.from\_user.id, message)

@bot.message\_handler(content\_types=["text"])

def send\_text(message):

if message.text == 'Треугольник':

bot.send\_message(message.chat.id, 'Выбери круг',reply\_markup = figure\_chose\_markup)

elif message.text == 'Круг':

circle\_unknown\_param(message)

elif message.text == 'Квадрат':

bot.send\_message(message.chat.id, 'Выбери круг ',reply\_markup = figure\_chose\_markup)

else:

print(time.ctime())

print(message.from\_user.id)

print(message.text)

r.append(str(message.from\_user.id), str(message.text + " | "))

def circle\_unknown\_param(message):

circle\_choose\_unknown\_param = telebot.types.ReplyKeyboardMarkup(True, True)

circle\_choose\_unknown\_param.row('r (найти радиус)', 'd (найти диаметр)' )

circle\_choose\_unknown\_param.row('l (найти длину круга)', 'S (найти площадь)')

photo = Image.open('circle.png')

bot.send\_photo(message.chat.id, photo)

#bot.send\_message(message.chat.id, 'photo')

bot.send\_message(message.chat.id, 'Выбери неизвестный параметр: ',reply\_markup = circle\_choose\_unknown\_param)

bot.register\_next\_step\_handler(message, circle\_choose\_known\_param)

def circle\_choose\_known\_param(message):

if message.text == 'r (найти радиус)':

circle\_choose\_known\_param = telebot.types.ReplyKeyboardMarkup(True, True)

circle\_choose\_known\_param.row('Длина, неизвестен радиус')

circle\_choose\_known\_param.row('Диаметр, неизвестен радиус')

circle\_choose\_known\_param.row('Площадь, неизвестен радиус')

bot.send\_message(message.chat.id, 'Выберите известный параметр:', reply\_markup=circle\_choose\_known\_param)

bot.register\_next\_step\_handler(message, circle\_radius\_by\_smth)

elif message.text == 'd (найти диаметр)':

circle\_choose\_known\_param = telebot.types.ReplyKeyboardMarkup(True, True)

circle\_choose\_known\_param.row('Радиус, неизвестен диаметр')

circle\_choose\_known\_param.row('Длина, неизвестен диаметр')

circle\_choose\_known\_param.row('Площадь, неизвестен диаметр')

bot.send\_message(message.chat.id, 'Выберите известный параметр:', reply\_markup=circle\_choose\_known\_param)

bot.register\_next\_step\_handler(message, circle\_diametr\_by\_smth)

elif message.text == 'l (найти длину круга)':

circle\_choose\_known\_param = telebot.types.ReplyKeyboardMarkup(True, True)

circle\_choose\_known\_param.row('Радиус, неизвестна длина')

circle\_choose\_known\_param.row('Диаметр, неизвестна длина')

circle\_choose\_known\_param.row('Площадь, неизвестна длина')

bot.send\_message(message.chat.id, 'Выберите известный параметр:', reply\_markup=circle\_choose\_known\_param)

bot.register\_next\_step\_handler(message, circle\_length\_by\_smth)

elif message.text == 'S (найти площадь)':

circle\_choose\_known\_param = telebot.types.ReplyKeyboardMarkup(True, True)

circle\_choose\_known\_param.row('Радиус, неизвестна площадь')

circle\_choose\_known\_param.row('Диаметр, неизвестна площадь')

circle\_choose\_known\_param.row('Длина, неизвестна площадь')

bot.send\_message(message.chat.id, 'Выберите известный параметр:', reply\_markup=circle\_choose\_known\_param)

bot.register\_next\_step\_handler(message, circle\_square\_by\_smth)

else:

bot.send\_message(message.chat.id, 'Что-то пошло не так, начни со /start')

def circle\_radius\_by\_smth(message):

if message.text == 'Длина, неизвестен радиус':

bot.send\_message(message.chat.id, 'Введите длину(см):')

bot.register\_next\_step\_handler(message, circle\_radius\_by\_length)

elif message.text == 'Диаметр, неизвестен радиус':

bot.send\_message(message.chat.id, 'Введите длину(см):')

bot.register\_next\_step\_handler(message, circle\_radius\_by\_diametr)

elif message.text == 'Площадь, неизвестен радиус':

bot.send\_message(message.chat.id, 'Введите площадь(см^2) \*π:')

bot.register\_next\_step\_handler(message, circle\_radius\_by\_square)

else:

bot.send\_message(message.chat.id, 'Что-то пошло не так, начни со /start')

def circle\_radius\_by\_length(message):

length = int(message.text)/2

length\_text = str(length)

answer = ['Ваш радиус: ', length\_text, '/π', ' (см)']

answer = ''.join(answer)

bot.send\_message(message.chat.id, answer)

def circle\_radius\_by\_diametr(message):

diam = int(message.text)/2

diam\_text = str(diam)

answer = ['Ваш радиус: ', diam\_text, ' (см)']

answer = ''.join(answer)

bot.send\_message(message.chat.id, answer)

def circle\_radius\_by\_square(message):

square = sqrt(int(message.text))

square\_text = str(square)

answer = ['Ваш радиус: ', square\_text, ' (см)']

answer = ''.join(answer)

bot.send\_message(message.chat.id, answer)

def circle\_diametr\_by\_smth(message):

if message.text == 'Радиус, неизвестен диаметр':

bot.send\_message(message.chat.id, 'Введите радиус(см):')

bot.register\_next\_step\_handler(message, circle\_diametr\_by\_radius)

elif message.text == 'Длина, неизвестен диаметр':

bot.send\_message(message.chat.id, 'Введите длину(см):')

bot.register\_next\_step\_handler(message, circle\_diametr\_by\_length)

elif message.text == 'Площадь, неизвестен диаметр':

bot.send\_message(message.chat.id, 'Введите площадь(см^2) \*π:')

bot.register\_next\_step\_handler(message, circle\_diametr\_by\_square)

else:

bot.send\_message(message.chat.id, 'Что-то пошло не так, начни со /start')

def circle\_diametr\_by\_radius(message):

radius = int(message.text)\*2

radius\_text = str(radius)

answer = ['Ваш диаметр: ', radius\_text, ' (см)']

answer = ''.join(answer)

bot.send\_message(message.chat.id, answer)

def circle\_diametr\_by\_length(message):

length = int(message.text)

length\_text = str(length)

answer = ['Ваш диаметр: ', length\_text, '/π', ' (см)']

answer = ''.join(answer)

bot.send\_message(message.chat.id, answer)

def circle\_diametr\_by\_square(message):

square = (sqrt(int(message.text)))\*2

square\_text = str(square)

answer = ['Ваш диаметр: ', square\_text, ' (см)']

answer = ''.join(answer)

bot.send\_message(message.chat.id, answer)

def circle\_length\_by\_smth(message):

if message.text == 'Радиус, неизвестна длина':

bot.send\_message(message.chat.id, 'Введите радиус(см):')

bot.register\_next\_step\_handler(message, circle\_length\_by\_radius)

elif message.text == 'Диаметр, неизвестна длина':

bot.send\_message(message.chat.id, 'Введите диаметр(см):')

bot.register\_next\_step\_handler(message, circle\_length\_by\_diametr)

elif message.text == 'Площадь, неизвестна длина':

bot.send\_message(message.chat.id, 'Введите площадь(см^2) \*π:')

bot.register\_next\_step\_handler(message, circle\_length\_by\_square)

else:

bot.send\_message(message.chat.id, 'Что-то пошло не так, начни со /start')

def circle\_length\_by\_radius(message):

radius = int(message.text)\*2

radius\_text = str(radius)

answer = ['Ваша длина круга: ', radius\_text, 'π', ' (см)']

answer = ''.join(answer)

bot.send\_message(message.chat.id, answer)

def circle\_length\_by\_diametr(message):

diametr = int(message.text)

diametr\_text = str(diametr)

answer = ['Ваша длина круга: ', diametr\_text, 'π', ' (см)']

answer = ''.join(answer)

bot.send\_message(message.chat.id, answer)

def circle\_length\_by\_square(message):

square = sqrt((int(message.text)\*4))

square\_text = str(square)

answer = ['Ваша длина круга: ', square\_text, ' (см)']

answer = ''.join(answer)

bot.send\_message(message.chat.id, answer)

def circle\_square\_by\_smth(message):

if message.text == 'Радиус, неизвестна площадь':

bot.send\_message(message.chat.id, 'Введите радиус(см):')

bot.register\_next\_step\_handler(message, circle\_square\_by\_radius)

elif message.text == 'Диаметр, неизвестна площадь':

bot.send\_message(message.chat.id, 'Введите диаметр(см):')

bot.register\_next\_step\_handler(message, circle\_square\_by\_diametr)

elif message.text == 'Длина, неизвестна площадь':

bot.send\_message(message.chat.id, 'Введите длину(см):')

bot.register\_next\_step\_handler(message, circle\_square\_by\_length)

else:

bot.send\_message(message.chat.id, 'Что-то пошло не так, начни со /start')

def circle\_square\_by\_radius(message):

radius = (int(message.text))\*\*2

radius\_text = str(radius)

answer = ['Ваша площадь: ', radius\_text, 'π', ' (см^2)']

answer = ''.join(answer)

bot.send\_message(message.chat.id, answer)

def circle\_square\_by\_diametr(message):

diametr = (int(message.text)/2)\*\*2

diametr\_text = str(diametr)

answer = ['Ваша площадь: ', diametr\_text, 'π', ' (см^2)']

answer = ''.join(answer)

bot.send\_message(message.chat.id, answer)

def circle\_square\_by\_length(message):

length = ((int(message.text))\*\*2)/4

length\_text = str(length)

answer = ['Ваша площадь: ', length\_text,'π', ' (см^2)']

answer = ''.join(answer)

bot.send\_message(message.chat.id, answer)

bot.polling()

**Додаток 2 (обробники веб запитів)**

import telebot

import redis

handle = open("token.txt", "r")

token = handle.readline()

r = redis.Redis(host='127.0.0.1', port=6379)

bot = telebot.TeleBot(token)

#for getting from redis

#r.get('IP').decode("utf-8")

#for sending messages

#bot.send\_message(IP, 'message')

**Додаток 3 (робота додатку)**

