

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования **«МИРЭА – Российский технологический университет»**

**РТУ МИРЭА**

**Институт кибербезопасности и цифровых технологий**

*(наименование института, филиала)*

**Кафедра КБ-8 «Информационное противоборство»**

*(наименование кафедры)*

**ОТЧЁТ ПО УЧЕБНОЙ** **ПРАКТИКЕ**

*(указать вид практики: учебная / производственная)*

**ОЗНАКОМИТЕЛЬНАЯ ПРАКТИКА**

(указать тип практики в соответствии с учебным планом)

Тема практики:   «Free Proxy Server»

Приказ Университета о направлении на практику от  
«26» января 2022 г. № 155-У

Отчёт предоставлен к рассмотрению:

|  |  |  |
| --- | --- | --- |
| Студент группы БББО-01-21 | «31»   мая   2022 г. | *(подпись и расшифровка)* |

Отчёт утверждён.

Допущен к защите:

|  |  |  |
| --- | --- | --- |
| Руководитель практики от кафедры | «31»   мая   2022 г. | *(подпись и расшифровка)* |



МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования **«МИРЭА – Российский технологический университет»  
РТУ МИРЭА**

**Институт кибербезопасности и цифровых технологий**

*(наименование института, филиала)*

**Кафедра КБ-8 «Информационное противоборство»**

*(наименование кафедры)*

**ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ НА УЧЕБНУЮ ПРАКТИКУ**

*(указать вид практики: учебная / производственная)*

**ОЗНАКОМИТЕЛЬНАЯ ПРАКТИКА**

*(указать тип практики в соответствии с учебным планом)*

**Студенту** 1 **курса учебной группы** БББО-01-21

Мысливцу Леониду Владимировичу

*(фамилия, имя и отчество)*

**Место и время практики:** кафедра КБ-8, с 09.02 по 31.05

**Должность на практике (при наличии):**

**1. СОДЕРЖАНИЕ ПРАКТИКИ:**

1.1. Изучить: технологии проектирования и автоматизации систем безопасности с использованием ЯП Python; действующие правовые нормы в области ИБ; научно-техническую информацию в области технологий обеспечения ИБ.

1.2. Практически выполнить: провести исследование в рамках проблематики ИБ в соответствии с индивидуальным вариантом задания; решить прикладные типовые задачи на ЯП Python; освоить ООП в Python; создать концепт продукта ИБ; обосновать полученные результаты и изложить их с соблюдением основных требований к оформлению научной продукции.

1.3. Ознакомиться: с методами работы с источниками информации, методами проектирования и создания концептов продуктов ИБ с использованием ЯП Python; современными методами научных исследований и экспериментов.

**2.** **ДОПОЛНИТЕЛЬНОЕ ЗАДАНИЕ:**

осуществить анализ продуктов и решений в области кибербезопасности представленных на тематических выставках.

**3. ОРГАНИЗАЦИОННО-МЕТОДИЧЕСКИЕ УКАЗАНИЯ:**

Определить цели практической работы. Оценить поставленные цели с позиций роли информации, информационных технологий и информационной безопасности в современном обществе; сформулировать соответствующий целям круг задач; проанализировать задачи, выделить базовые составляющие задач; осуществить поиск достоверной информации для решения задач; привести необходимые определения, правильным образом интерпретировать информацию, выделить главное; выбрать оптимальный способ решения задачи с учётом имеющихся ресурсов, ограничений и действующих правовых норм; при командной работе определить и реализовать свою роль в команде для достижения поставленных целей, активно взаимодействовать с членами команды, обмениваясь с ними информацией, знаниями и опытом. При работе с источниками обратить внимание на авторитетные издания (приоритет – входящие в перечень РИНЦ или ВАК). При разработке концепта продукта использовать открытые библиотеки.

Руководитель практики от кафедры:

|  |  |  |
| --- | --- | --- |
| «09» февраля 2022 г. |  | ( Отраднов К.К. ) |
|  | *Подпись* | *ФИО* |

Задание получил:

|  |  |  |
| --- | --- | --- |
| «09» февраля 2022 г. |  | ( Мысливец Л.В. ) |
|  | *Подпись* | *ФИО* |

**СОГЛАСОВАНО:**

Заведующий кафедрой:

|  |  |  |
| --- | --- | --- |
| «09» февраля 2022 г. |  | ( Григорьев В. Р. ) |
|  | *Подпись* | *ФИО* |

**Проведенные инструктажи:**

Охрана труда: « 09 » февраля 2022 г.

Инструктирующий \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ( Отраднов К.К., ст. препод. )   
 *Подпись Расшифровка, должность*

Инструктируемый \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ( Мысливец Л.В. )   
 *Подпись Расшифровка*

Техника безопасности « 09 » февраля 2022 г.

Инструктирующий \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ( Отраднов К.К., ст. препод. )   
 *Подпись Расшифровка, должность*

Инструктируемый \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ( Мысливец Л.В. )   
 *Подпись Расшифровка*

Пожарная безопасность « 09 » февраля 2022 г.

Инструктирующий \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ( Отраднов К.К., ст. препод. )   
 *Подпись Расшифровка, должность*

Инструктируемый \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ( Мысливец Л.В. )   
 *Подпись Расшифровка*

С правилами внутреннего распорядка ознакомлен: « 09 » февраля 2022 г.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ( Мысливец Л.В. )  *Подпись Расшифровка*



МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования  
**«МИРЭА – Российский технологический университет»  
РТУ МИРЭА**

**РАБОЧИЙ ГРАФИК ПРОВЕДЕНИЯ**

**УЧЕБНОЙ** **ПРАКТИКИ**

студента Мысливец Л.В. 1 курса группы БББО-01-21   очной

формы обучения, обучающегося по направлению подготовки 10.03.01, Информационная безопасность ,

профиль Организация и технологии защиты информации.

|  |  |  |  |
| --- | --- | --- | --- |
| **Неделя** | **Сроки выполнения** | **Этап** | **Отметка о выполнении** |
| 1-3 | 09.02.2022 | Решение прикладных типовых задач на ЯП Python |  |
| 4-5 | 02.03.2022 | Разработка концепта системы автономных взаимодействующих агентов на ЯП Python |  |
| 6-7 | 16.03.2022 | Анализ продуктов и решений в области кибербезопасности, представленных на выставке Securika |  |
| 8-9 | 30.03.2022 | Обзор существующих проектов и концепт идея разработки |  |
| 10-16 | 31.05.2022 | Подготовка руководства и оформление отчета |  |

|  |  |
| --- | --- |
| Руководитель практики от кафедры | / Отраднов К.К.  /ФИО, учёная степень, учёное звание/ |
| Обучающийся | / Мысливец Л.В.  /ФИО/ |

**Согласовано:**

|  |  |
| --- | --- |
| Заведующий кафедрой | / Григорьев В. Р., к.т.н.  /ФИО, учёная степень, учёное звание/ |

**ОТЧЕТ**

**по учебной практике**

студента 1 курса учебной группы БББО-01-21   
Института кибербезопасности и цифровых технологий

  Мысливца Леонида Владимировича

1. Практику проходил с 09 февраля 2022 г. по 31 мая 2022 г. на кафедре КБ-8 «Информационное противоборство».

2. Задание на практику:  выполнил в полном объеме

(указать: в полном объеме или частично)

2.1 Не выполнены следующие задания:  **–**

(указать также причины невыполнения)

3. Подробное содержание выполненной на практике работы и достигнутые результаты: Решение прикладных задач на ЯП Python, разработка концепта системы автономных взаимодействующих агентов, анализ продуктов и решений в области кибербезопасности и реализация собственного продукта ИБ на ЯП Python (включая обзор существующих проектов и описание концепт идеи разработки).

4. Предложения по совершенствованию организации и прохождения практики: **–**

|  |  |  |
| --- | --- | --- |
| Студент группы БББО-01-21 | «31»   мая   2022 г. | *(подпись и расшифровка)* |

5. Заключение руководителя практики:

Приобрел следующие профессиональные навыки: умение оценивать роль информации, ИТ и ИБ в современном обществе; умение осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач; умение осуществлять подбор, изучение и обобщение научно-технической литературы, нормативных и методических документов в целях решения задач профессиональной деятельности; умение определять круг задач в рамках поставленной цели и выбирать оптимальные способы их решения, исходя из действующих правовых норм, имеющихся ресурсов и ограничений.

Проявил себя как: ответственный, целеустремленный, дисциплинированный и инициативный студент.

**Отчет проверил:**

|  |  |  |
| --- | --- | --- |
| Руководитель практики от кафедры | «31»   мая   2022 г. | *(подпись и расшифровка)* |

**CОДЕРЖАНИЕ**

[**CОДЕРЖАНИЕ** 6](#_Toc105473533)

[**ВВЕДЕНИЕ** 7](#_Toc105473534)

[**1.ОСНОВНАЯ ЧАСТЬ** 8](#_Toc105473535)

[**1.1 Решение прикладных задач на языке программирования Python** 8](#_Toc105473536)

[**1.2 Разработка концепта системы автономных взаимодействующих агентов на языке программирования Python** 14](#_Toc105473537)

[**1.3 Анализ продуктов и решений в области кибербезопасности, представленных на выставке Securika 2022** 17](#_Toc105473538)

[**1.4 Обзор существующих проектов и концепт идея разработки** 21](#_Toc105473539)

[**1.5 Руководство** 24](#_Toc105473540)

[**1.5.1 Руководство пользователя** 24](#_Toc105473541)

[**1.5.2 Руководство программиста** 26](#_Toc105473542)

[**1.5.3 Руководство системного администратора** 27](#_Toc105473543)

[**ЗАКЛЮЧЕНИЕ** 32](#_Toc105473544)

[**СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ** 33](#_Toc105473545)

[**ПРИЛОЖЕНИЕ А** 34](#_Toc105473546)

[**ПРИЛОЖЕНИЕ Б** 42](#_Toc105473547)

**ВВЕДЕНИЕ**

В современном мире, мире технологического прогресса и информатизации разработка ПО и программирование в целом являются неотъемлемой частью этого мира. На курсе учебной практики студенты изучают язык разработки Python. Эта среда зарекомендовала себя, как достаточно простую в понимании и легкую в изучении. Сам по себе Python является скриптовым языком программирования, что так же отражается на логике и принципах самих функций языка. На протяжении всего семестра Python используется в разных областях, начиная с математики, заканчивая шифрованием и созданием игры. В рамках учебной практики студентам необходимо продемонстрировать уровень владения этим языком и знания в области ООП. Начинающим специалистам в области Информационной безопасности необходимо знать, уметь, владеть базовыми принципами построения алгоритмов, а также их продемонстрировать.

**1.ОСНОВНАЯ ЧАСТЬ**

**1.1 Решение прикладных задач на языке программирования Python**

Задание 1. Дан одномерный массив числовых значений, насчитывающий N элементов. Поменять местами элементы, стоящие на чётных и нечётных местах: A[1] ↔ A[2]; A[3] ↔ A[4]: чётным, кратным 4.

Листинг 1.1

n,m=int(input()),[]

for i in range (n):

m.append(int(input()))

for j in range (n-1):

if (j%2 == 0):

m[j],m[j+1]=m[j+1],m[j]

print(m)

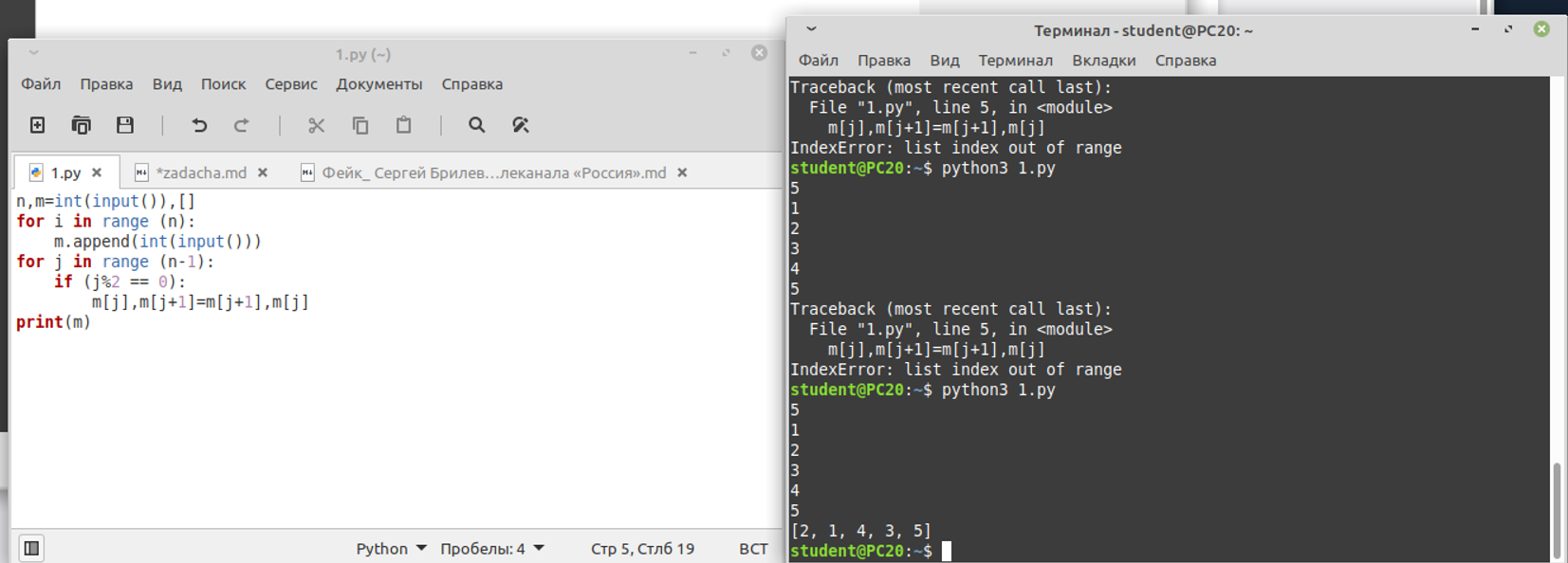


Рисунок 1.1 – Результат выполнения программы

Задание 2. Дан одномерный массив числовых значений, насчитывающий N элементов. Выполнить перемещение элементов массива по кругу вправо, т.е. A[1] → A[2]; A[2] → A[3]; ... A[n] → A[1].

Листинг 1.2

n,m=int(input()),[]

for i in range (n):

m.append(int(input()))

for i in range(n):

m[-i] , m[-i-1] = m[-i-1] , m[-i]

m[0] , m[1] = m[1] , m[0]

print(m)

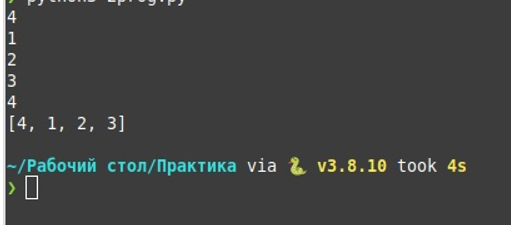


Рисунок 1.2 – Результат выполнения программы

Задание 3. Дан одномерный массив числовых значений, насчитывающий N элементов. Поменять местами первую и вторую половины массива.

Листинг 1.3

n,m=int(input()),[]

for i in range (n):

m.append(int(input()))

for i in range(len(m)//2):

t=m[i]

m[i]=m[len(m)//2+i]

m[len(m)//2+i]=t

print(m)

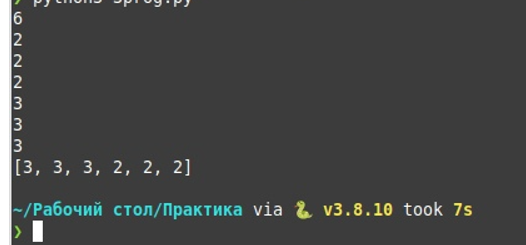


Рисунок 1.3 – Результат выполнения программы

Задание 4. Дан одномерный массив числовых значений, насчитывающий N элементов. Поменять местами группу из M элементов, начинающихся с позиции K с группой из M элементов, начинающихся с позиции P

Листинг 1.4

M,K,P,N,m = int(input()),int(input()),int(input()),int(input()),[]

for i in range (N):

m.append(int(input()))

print("N= " + str(N))

print("K= " + str(K))

print("P= " + str(P))

print("M= " + str(M))

print(m)

m[K : M + K + 1] , m[P : M+P+1] = m[P : M+P+1] , m[K : M+K+1]

print(m)

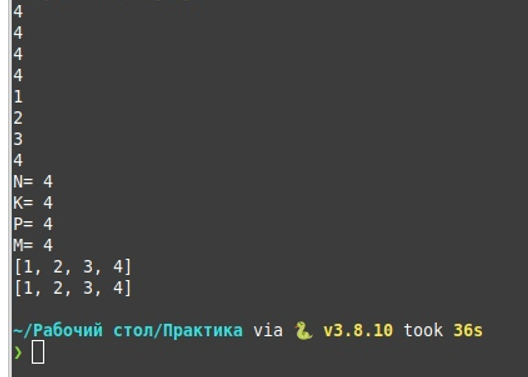


Рисунок 1.4 – Результат выполнения программ

Задание 5. Дан одномерный массив числовых значений, насчитывающий N элементов. Вставить группу из M новых элементов, начиная с позиции K.

Листинг 1.5

M,K,N=int(input()), int(input()),int(input())

m = []

for i in range (N):

m.append(int(input()))

print("N= " + str(N))

print("K= " + str(K))

print("M= " + str(M))

print(m)

m2 = []

for i in range (M):

m2.append(int(input()))

print(m2)

m.insert(K, m2)

print(m)

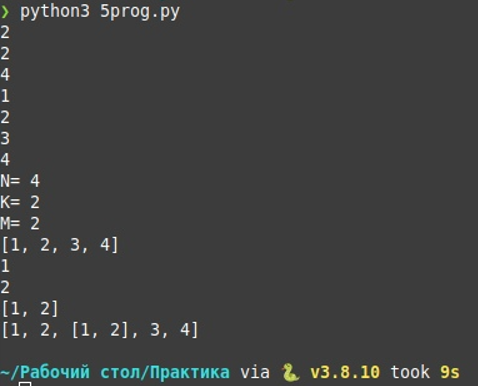


Рисунок 1.5 – Результат выполнения программы

Задание 6. Дан одномерный массив числовых значений, насчитывающий N элементов. Сумму элементов массива и количество положительных элементов поставить на первое и второе место

Листинг 1.6

n,m=int(input()),[]

for i in range (n):

m.append(int(input()))

S = sum(m)

print("Sum m: " + str(S))

K=0

for i in range(n):

if m[i] >= 0:

K+=1

print("Positive numbers in array: " + str(K))

m.insert(0, S)

m.insert(1, K)

print(m)

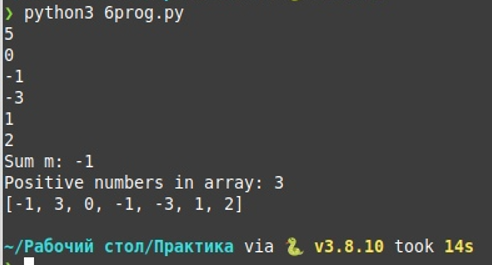


Рисунок 1.6 – Результат выполнения программы

Задание 7. Дан одномерный массив числовых значений, насчитывающий N элементов. Исключить из него M элементов, начиная с позиции K.

Листинг 1.7

n,m=int(input()),[]

for i in range (n):

m.append(int(input()))

K = int(input())

print("начиная с К=" + str(K))

M = int(input())

print("исключить M эл-ов=" + str(M))

m[K:K+M]=[]

print(m)

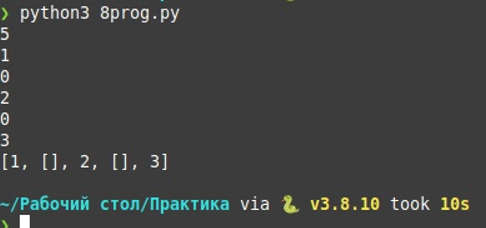


Рисунок 1.7 – Результат выполнения программы

Задание 8. Дан одномерный массив числовых значений, насчитывающий N элементов. Исключить все нулевые элементы.

Листинг 1.8

n,m=int(input()),[]

for i in range (n):

    m.append(int(input()))

for i in range(n):

    if m[i] == 0:m[i] = []

print(m)

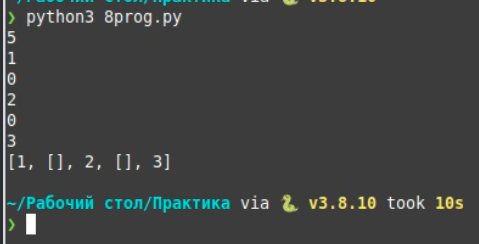


Рисунок 1.8 – Результат выполнения программы

Задание 9. Дан одномерный массив числовых значений, насчитывающий N элементов. После каждого отрицательного элемента вставить новый элемент, равный квадрату этого отрицательного элемента.

Листинг 1.9

n,m=int(input()),[]

for i in range (n):

m.append(int(input()))

for i in range(n):

if m[i] < 0:

m.insert( i+1 , m[i] 2)

if m[-1] < 0:m.append(m[-1] 2)

print(m)

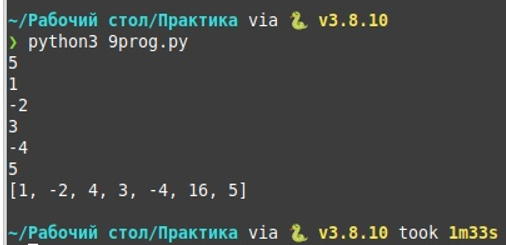


Рисунок 1.9 – Результат выполнения программы

Задание10. Дан одномерный массив числовых значений, насчитывающий N элементов. Определить, образуют ли элементы массива, расположенные перед первым отрицательным элементом, возрастающую последовательность.

Листинг 1.10

n,m,K=int(input()),[],1

for i in range (n):

m.append(int(input()))

for i in range(0, n):

if m[i] < 0:

break

if m[i] >= 0:

if m[i] > m[i + 1] and m[i + 1] >= 0:

K = 1

break

if m[i] < m[i + 1]:

K = 2

if K == 1:print("не возр")

elif K == 2:print("возр")

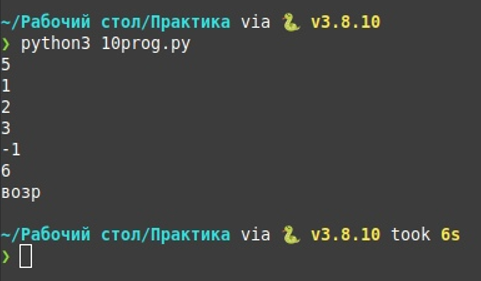


Рисунок 1.10 – Результат выполнения программы

**1.2 Разработка концепта системы автономных взаимодействующих агентов на языке программирования Python**

Мануал: У игрока есть некоторые статусы, а именно здоровье и выносливость (исчерпаем ресурс). Каждый ход затрачивает единицу выносливости, если она кончилась, то тратиться здоровье. Два игрока сражаются за первенство используя находящиеся на игровом поле зелья и атакуя друг друга. Атака отнимает единицу здоровья.

Листинг кода 1.2.1

import time

import os

import curses

import random

scr = curses.initscr()

curses.curs\_set(0)

curses.noecho()

objects = []

y\_coords = []

x\_coords = []

class Potion:

def \_\_init\_\_(self,hp\_restore,mana\_restore,x,y,char):

self.hp\_restore = hp\_restore

self.mana\_restore = mana\_restore

self.x = x

self.y = y

self.char = char

while self.x in x\_coords:

self.x = random.randint(1,19)

x\_coords.append(self.x)

while self.y in y\_coords:

self.y = random.randint(1,19)

y\_coords.append(self.y)

objects.append(self)

def action(self,target):

target.hp += self.hp\_restore

target.mana += self.mana\_restore

for i in range(random.randint(5,10)):

seed = random.randint(0,1)

if seed == 0:

Potion(150,0,random.randint(1,19),random.randint(0,20),'&')

else:

Potion(0,150,random.randint(1,19),random.randint(0,20),'O')

class Environment:

def \_\_init\_\_(self):

self.room = [["." for i in range(21)] for k in range(21)]

def move\_player(self, entity, direction):

if direction == (0,1):

entity.fov = 1

elif direction == (0,-1):

entity.fov = 3

elif direction == (1,0):

entity.fov = 4

elif direction == (-1,0):

entity.fov = 2

if entity.x + direction[1] >= 0 and entity.x + direction[1] <= 20 and entity.y + direction[0] >= 0 and entity.y + direction[0] <= 20:

for object in objects:

if entity.x + direction[1] == object.x and entity.y + direction[0] == object.y and type(object) == Player:

self.fight(entity,object)

return

entity.x += direction[1]

entity.y += direction[0]

if entity.mana > 0:

entity.mana -= 1

else:

entity.hp -= 1

for object in objects:

if entity.x == object.x and entity.y == object.y and type(object) == Potion:

objects.remove(object)

object.action(entity)

def fight(self,attacker,target):

if attacker.fov in (1,3) and target.fov in (2,4) or attacker.fov in (2,4) and target.fov in (3,1):

target.hp -= 2

elif attacker.fov == target.fov:

target.hp -= 3

else:

target.hp -= 1

def render\_all(self):

x = 0

y = 0

for i in range(len(self.room)):

x = 0

for k in range(len(self.room[0])):

scr.addstr(y,x,self.room[i][k])

x += 1

y += 1

for object in objects:

scr.addstr(object.y,object.x,object.char)

for i in range(3):

scr.addstr(i,22," ")

scr.addstr(0,22,"Player0 hp: " + str(Player0.hp))

scr.addstr(1,22,"Player0 mana: " + str(Player0.mana))

scr.addstr(2,22,"Player0 fov: " + str(Player0.fov))

scr.addstr(0,40,"Player1 hp: " + str(Player1.hp))

scr.addstr(1,40,"Player1 mana: " + str(Player1.mana))

scr.addstr(2,40,"Player1 fov: " + str(Player1.fov))

scr.refresh()

main\_environment = Environment()

def continuegame():

if Player0.hp > 0 and Player1.hp > 0:

return True

if Player0.hp < 0:

scr.addstr(25,0,"Player1 won!")

if Player1.hp < 0:

scr.addstr(25,0,"Player0 won!")

scr.refresh()

time.sleep(2)

return False

class Player:

def \_\_init\_\_(self,x,y,char,fov, hp, mana):

self.x = x

self.y = y

self.char = char

self.fov = fov

self.hp = hp

self.mana = mana

objects.append(self)

def pathfind(obj1,self):

dx = obj1.x - self.x

dy = obj1.y - self.y

rasstoyanie = (dx 2 + dy 2) 0.5

dx = round(dx/rasstoyanie)

dy = round(dy/rasstoyanie)

if abs(dx) == abs(dy) and dx != 0:

dx = 0

return (dy,dx)

def nearest(self):

distances = []

for object in objects:

if object != self:

distances.append( [ ((self.x-object.x) 2 + (self.y - object.y) 2) 0.5, object])

distances.sort(key = lambda x: x[0])

return distances[0][1]

def move\_or\_attack(self, environment):

seed = random.randint(0,10)

if seed in (9,10):

move\_seed = random.randint(1,4)

environment.move\_player(self, directions[str(move\_seed)])

elif seed in (7,8):

self.fov = random.randint(1,4)

elif seed in (1,2,3):

environment.move\_player(self, self.pathfind(self.nearest()))

else:

if self.char == '@':

environment.move\_player(self, Player1.pathfind(self))

elif self.char == '%':

environment.move\_player(self, Player0.pathfind(self))

Player0 = Player(0,0,'@',0,random.randint(50,100),100)

Player1 = Player(20,20,'%',0,random.randint(50,100),100)

main\_environment.render\_all()

directions = { '1' : (0,1),

'2' : (1,0),

'3' : (0,-1),

'4' : (-1,0) }

while continuegame():

time.sleep(1)

Player0.move\_or\_attack(main\_environment)

Player1.move\_or\_attack(main\_environment)

main\_environment.render\_all()

**1.3 Анализ продуктов и решений в области кибербезопасности, представленных на выставке Securika 2022**

По программе учебной практике студентам представилась уникальная возможность посетить выставку Securika 2022. В качестве экспонатов на выставке были представлены многочисленные варианты средств противопожарной безопасности, множество вариантов звукоизоляции, кабель-менеджмента, а также камер видеонаблюдения. Далее предоставлена их подробная характеристика в Таблице 1.3.1. и Таблице 1.3.2

Таблица 1.3.1



Таблица 3.2



На основании приведённых в Таблице 1.3.1. и Таблице 1.3.2 данных мы можем сделать вывод, что камеры аналогового типа на лидерство в рейтингах лучших камер видеонаблюдения уже давно не претендуют, хотя и востребованы в частных случаях. Если кратко, их достоинствами являются сравнительно меньшая стоимость, возможность использования более длинных линий передачи и отсутствие задержек. Покупка аналоговых устройств целесообразна еще и в том случае, когда целью ставится обновление существующей системы видеонаблюдения без замены кабельного хозяйства. Ориентироваться стоит на современные стандарты (семейство AHD, TVI и CVI). Кстати, их многочисленность и несовместимость нагляднее всего демонстрируют кризис аналоговых систем. Преимущества IP-камер заключаются в принципиальном отсутствии потерь при передаче, достаточной защищенности и полной совместимости со стандартными цифровыми каналами обмена информацией. Среди основных минусов: выше цена, в целом короче кабели подключения, сильнее задерживается сигнал. Самыми распространенными конструкциями уличных камер видеонаблюдения являются купольные (Dome) и цилиндрические (Bullet). Среди камер для помещений разнообразия значительно больше. Принятое деление по месту установки говорит только о защищенности камеры – во всех смыслах. Понятно, что уличную камеру внутри дома ставить никто не запрещает. Максимально «раздвинуть» границы контролируемой зоны позволяют устройства панорамного и поворотного типа, а также камеры с переменным фокусным расстоянием объектива. Выбор лучшей модели камеры видеонаблюдения для дома или улицы полностью определяется задачами, которые они должны решать. К примеру, для идентификации на дальних расстояниях потребуется камера с разрешением картинки повыше. Ее недостатки: архив «тяжелее», стоит дороже, в темное время суток снимает хуже (как правило). Второй вариант — ставить камеру с меньшей разрешающей способностью сенсора и большим фокусным расстоянием объектива. Самыми актуальными сейчас являются устройства на 2—5 МП.

В целом выставка вышла увлекательная и познавательная, с уверенностью можно сказать, что благодаря такого рода мероприятиям у студентов появляется большой стимул к образованию.

**1.4 Обзор существующих проектов и концепт идея разработки**

Использование одного из лучших прокси-сервисов на сегодняшний день обеспечивает простой вариант защиты вашей конфиденциальности в Интернете. По сути, лучший прокси-сервер служит шлюзом между вами и Интернетом, поэтому при использовании прокси-сервера сведения о посещаемых вами сайтах и другая онлайн-история сохраняются на этом сервере, а не на вашем компьютере. В рамках изучения среды разработки Python была поставлена задача реализовать продвинутый парсер для поиска наиболее лучших proxy-серверов. Далее представлена сравнительная характеристика существующих Proxy-сервисов.

1. AstroProxy

Тарифы: мобильные, резидентские и серверные по 15₽ за порт;

Доступные ГЕО: Россия, Украина, Казахстан, Беларусь, Германия, Франция, Чили, Европейский союз, Канада, Финляндия, Турция, Япония;

Бесплатный тест: есть;

Реферальная программа: от 10% до 15% всех платежей привлеченных рефералов;

1. ProxyLine

Тарифы: IPv4 Shared — 49,56₽ за IP, индивидуальные IPv4 — 72,68₽, IPv6/32 — 7,43₽ (цена указана за 5 дней использования);

Доступные ГЕО: для IPv4 — 46+, среди них Австралия, Бразилия, Великобритания, Ирландия, Казахстан, Россия и Польша, а для IPv6 только Россия и США;

Бесплатный тест: есть;

Реферальная программа: 10% от всех платежей реферала;

1. Proxys.io

Тарифы: Зарубежные IPv4 — $1,47, Индивидуальные IPv4 — $1.4 за IP (цена указана за 30 дней использования);

Доступные ГЕО: 26+, среди них Украина, Россия, Великобритания, США, Германия, Индия, Польша, Испания, Беларусь, Нидерланды, Казахстан, Франция;

Бесплатный тест: есть;

Реферальная программа: нет;

1. Soax

Тарифы: резидентские — $7/1ГБ, мобильные — $29/1ГБ;

Доступные ГЕО: 175+, среди них Англия, Франция, Канада, Польша, Россия, Австралия, США, Казахстан, Турция, Таиланд;

Бесплатный тест: нет;

Реферальная программа: нет;

Таблица 1.4.1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Название сервиса | Платная | Большое кол-во геолокаций | Надёжное соединения | Ежедненое обновление | Поддержка всех типов proxy |
| AstroProxy | + | + | + | + | + |
| ProxyLine | + | + | + | + | + |
| Proxys.io​ | + | + | + | - | + |
| Soax | + | + | + | + | + |
| FPS by Slasher | - | + | - | - | + |

На основание Таблицы 1.4.1 можно сделать вывод, что сервис FPS by Slasher не может гарантировать ни безопасность, ни ежедневное обновление, поскольку не имеет модераторов и службы поддержки, однако в отличие от своих конкурентов он не занимается поддержкой серверов, а забирает их с открытых ресурсов, которые предоставляют те же компании, а также находится в открытом доступе и совершенно бесплатный.

**1.5 Руководство**

**1.5.1 Руководство пользователя**

Для корректного функционирования программы на Linux скрипт необходимо открывать через терминал, как показано на рисунке 1.5.1.1

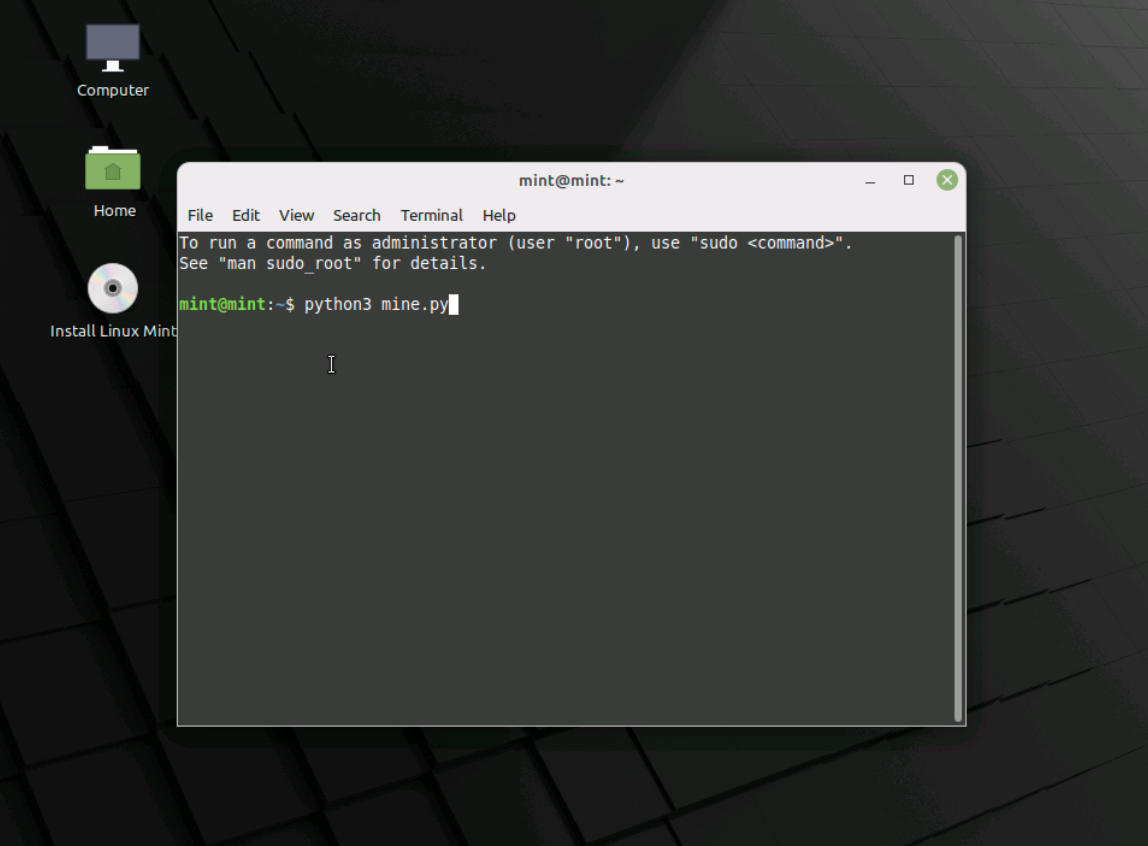


Рисунок 1.5.1.1 – Терминал с командой запуска

После запуска скрипта перед пользователем появляется окно с выбором функций реализуемых скриптом предоставленных на рисунке 1.5.1.2 [1]

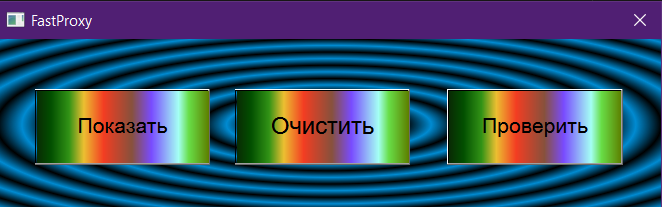


Рисунок 1.5.1.2 – Интерфейс программы

1. На кнопку “Показать” пользователю программа выводит пользователю базу данных с Proxy.[2],[4],[6]

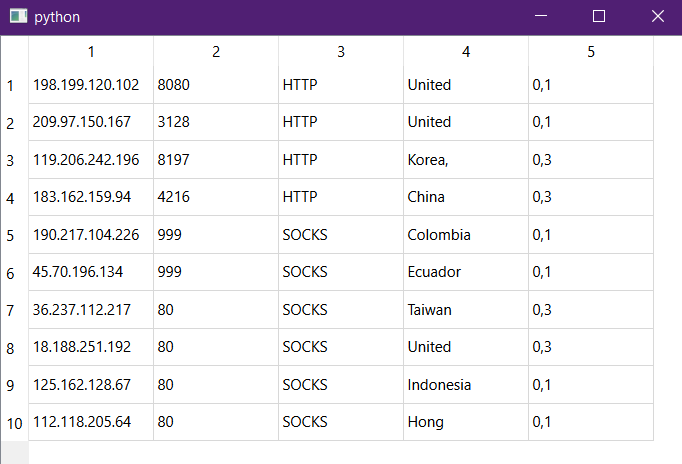


Рисунок 1.5.1.3 – Интерфейс программы

Как представлено на Рисунок 1.5.1.3 программа предоставляет полною информацию о прокси, а именно IP,порт, тип, страну, среднюю скорость.

1. На кнопку “Очистить” скрипт просто очищает базу данных.
2. Функционал кнопки “Проверить” на данный момент не реализован, для того, чтобы проверить скорость Proxy необходимо запустить отдельно реализованный скрипт как показано на рисунке 1.5.1.4

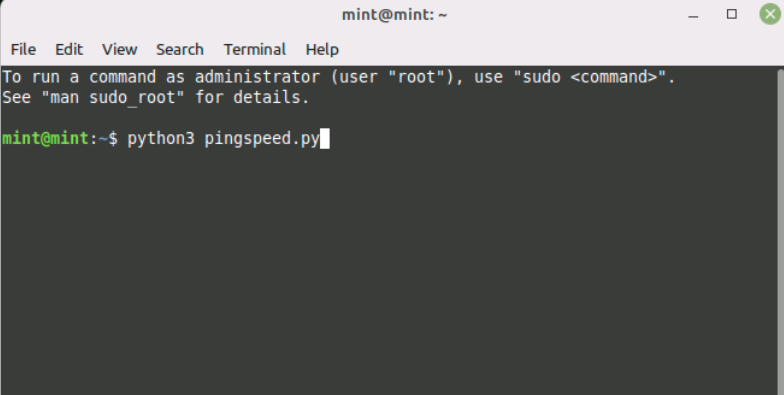


Рисунок 1.5.1.4 – Терминал с командой запуска

Скрипт откроет пользователю консольное приложение в котором необходимо ввести IP-адрес proxy вручную, как показано на рисунке 1.5.1.5 и в нижнем окне выведется скорость соединения пользователя с сервером.[10]

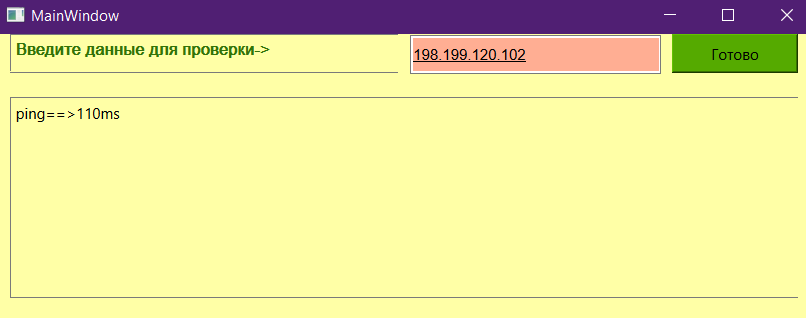


Рисунок 1.5.1.5 – Интерфейс программы

Для корректного функционирования пользователю необходимо установить среду разработки Python3 и ряд библиотек (в скобках команда для установки):

1. Pip-для корректной установки последующих библиотек (sudo apt -y install python3-pip)
2. Re – для использования регулярных выражений (pip install re)
3. sqlite3- для использования базы данных (pip install sqlite3)[4]
4. BeautifulSoup4-для использования (pip install beautifulsoup4)[3]
5. PyQt5, PyQt6, PyQt-tools- для использования GUI интерфейса [7],[8]
6. (pip install pyqt5; pip install pyqt6; pip install pyqt-tools) [7],[8]

**1.5.2 Руководство программиста**

**Описание логической структуры:**

Программа корректно функционирует на дистрибутиве Kali Linux

И поделена для простого восприятия на несколько файлов.[1],[10]

1) mine.py это главный файл программы в котором описано главное окно вывода, с помощью функционала PyQt в классе “Ui\_FastProxy” в методах “setupUi”,” retranslateUi”, “ add\_functions” , а также в “ write\_bd”, “ delete\_bd” , “ Test”, “ pingvich” вызываются отдельные функции для работы программы.

2)Parsik.py в этом файле реализована работа парсера в функциях в наименованиях которых “screb”, а заполнение в базу данных.[1],[2]

3)database.py в классе “Database” этого файла реализовано корректное обновление таблицы и инициализированы важные переменные для корректной работы с базой данных в других файлах.[4]

4)tablebd.py - файл программы в котором описано окно вывода таблицы Proxy, с помощью функционала PyQt.[7]

5)pingspeed.py- отдельный скрипт для проверки скорости Proxy по IP, реализованный с помощью функционала PyQt.[2]

Минимальные требования:

1. 64-битная версия Microsoft Windows 10, 8, 7 (SP1)
2. RAM: не менее 4 ГБ, рекомендуется 8 ГБ
3. 1,5 ГБ сводного места на жестком диске + не менее 1 ГБ для кэша
4. разрешение экрана — не менее 1024×768 пикселей
5. Python 2.7, Python 3.5 или более поздняя версия

Также пользователю необходимо установить ряд зависимостей:

Pip-для корректной установки последующих библиотек (sudo apt -y install python3-pip)

1. Re – для использования регулярных выражений (pip install re) [1]
2. sqlite3- для использования базы данных (pip install sqlite3) [4],[6]
3. BeautifulSoup4-для использования (pip install beautifulsoup4) [3]
4. PyQt5, PyQt6, PyQt-tools- для использования GUI интерфейса [7],[8]
5. (pip install pyqt5; pip install pyqt6; pip install pyqt-tools)

**Характеристики программы:**

На данный момент в программе не удалось реализовать одновременный запуск проверки скорости Proxy и парсинг Proxy с сайтов. Реализован приятный и простой интерфейс.

**1.5.3 Руководство системного администратора**

Условия выполнения программы:

Для корректного функционирования программы на Linux скрипт необходимо открывать через терминал, как показано на рисунке 1.5.1.1

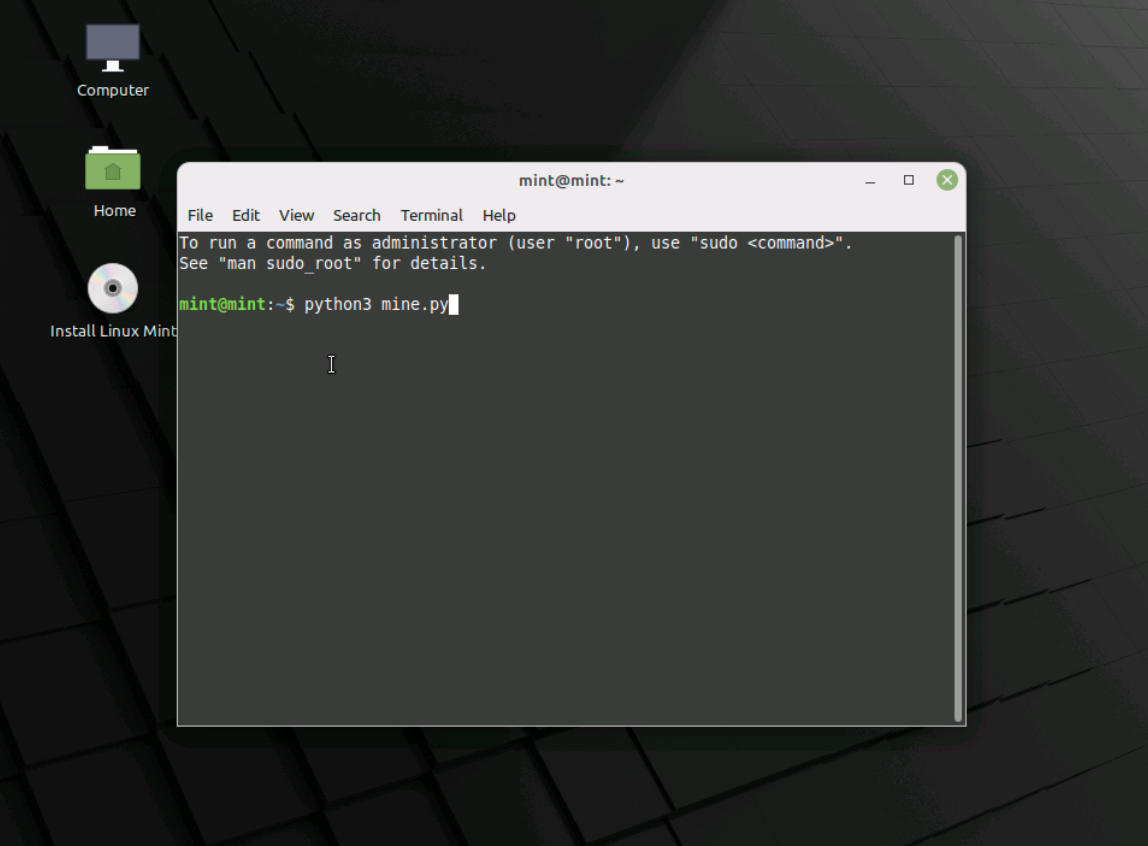


Рисунок 1.5.3.1 – Терминал с командой запуска

Настройка программы:

Программа работает корректно и не требует отдельных настроек помимо зависимостей среды разработки Python3,минимальных требований и набора библиотек для запуска скрипта.

Минимальные требования:

1. 64-битная версия Microsoft Windows 10, 8, 7 (SP1)
2. RAM: не менее 4 ГБ, рекомендуется 8 ГБ
3. 1,5 ГБ сводного места на жестком диске + не менее 1 ГБ для кэша
4. разрешение экрана — не менее 1024×768 пикселей
5. Python 2.7, Python 3.5 или более поздняя версия

Набор библиотек для запуска скрипта:

1. Pip-для корректной установки последующих библиотек (sudo apt -y install python3-pip)
2. Re – для использования регулярных выражений (pip install re)
3. sqlite3- для использования базы данных (pip install sqlite3)
4. BeautifulSoup4-для использования (pip install beautifulsoup4)
5. PyQt5, PyQt6, PyQt-tools- для использования GUI интерфейса
6. (pip install pyqt5; pip install pyqt6; pip install pyqt-tools)

Проверка программы:

Проверка программы осуществляется посредством запуска скрипта, в случае если программа запускается без ошибок, особых мер не предусматривается, однако необходимо постоянное обновление библиотеки PyQt поскольку алгоритмы библиотеки постоянно обновляются и это может повлиять на работоспособность.

Дополнительные возможности и функционал программы: После запуска скрипта перед пользователем появляется окно с выбором функций реализуемых скриптом предоставленных на рисунке 1.5.1.2

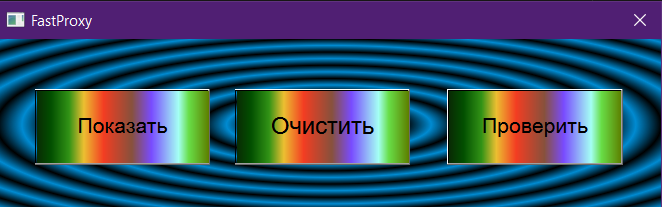


Рисунок 1.5.3.2 – Интерфейс программы

На кнопку “Показать” пользователю программа выводит пользователю базу данных с Proxy.

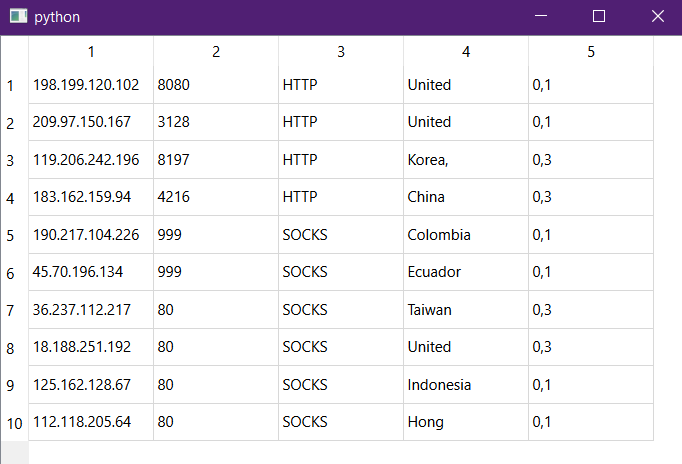


Рисунок 1.5.3.3 – Интерфейс программы

Как представлено на Рисунок 1.5.3.3 программа предоставляет полною информацию о прокси, а именно IP,порт, тип, страну, среднюю скорость.

На кнопку “Очистить” скрипт просто очищает базу данных.

Функционал кнопки “Проверить” на данный момент не реализован, для того, чтобы проверить скорость Proxy необходимо запустить отдельно реализованный скрипт как показано на рисунке 1.5.3.4

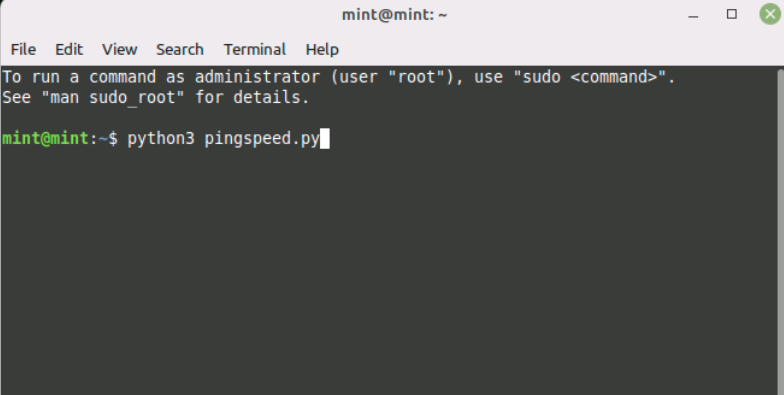


Рисунок 1.5.3.4 – Терминал с командой запуска

Скрипт откроет пользователю консольное приложение в котором необходимо ввести IP-адрес proxy вручную, как показано на рисунке 1.5.3.5 и в нижнем окне выведется скорость соединения пользователя с сервером.

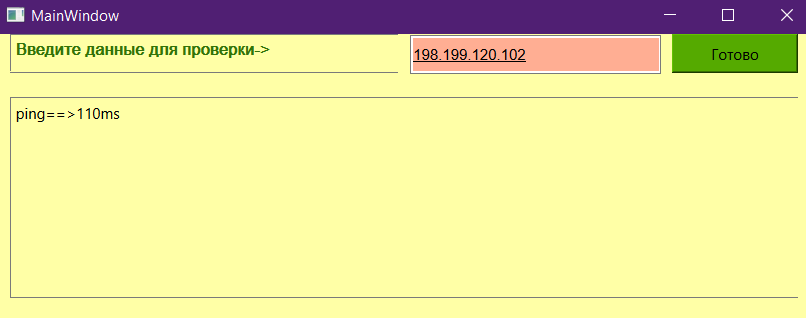


Рисунок 1.5.3.5 – Интерфейс программы

В данной программе реализован GUI интерфейс, что и является дополнительным функционалом программы.

# **ЗАКЛЮЧЕНИЕ**

В рамках учебной практики были предоставлены, в качестве подтверждения знаний, различные задания и различные данные, полученные за время обучения. В ходе проектной работы были подняты важные темы, которые необходимо было изучить будущим специалистам в сере Информационной безопасности, а также были изучены необходимые навыки владения языком программирования. Были подняты важные аспекты ООП, такие как актуальность изучения этой концепции, которая заключается в том, что она является востребованной парадигмой программирования при разработке программного обеспечения, что обусловливает программисту необходимость понимания и применения ООП. А также отдельное время было уделено посещению выставке Securika 2022, что особенно прибавило желания двигаться дальше в направлении информационной безопасности.

# **СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ**

1. Доступ к деревьям синтаксического анализа Python [Электронный ресурс].-URL: <https://docs.python.org/3.9/library/parser.html> (дата обращения 08.05.2022).

2. Руководство как написать свой первый парсер на Python [Электронный ресурс].-URL:<https://habr.com/ru/post/568334/> (дата обращения 08.05.2022).

3. Руководство Парсинг на Python с Beautiful Soup [Электронный ресурс].-URL: <https://pythonru.com/biblioteki/parsing-na-python-s-beautiful-soup> (дата обращения 08.05.2022).

4. Руководство по SQLite в Python [Электронный ресурс].-URL:<https://pythonru.com/osnovy/sqlite-v-python> (дата обращения 08.05.2022).

5. Руководство встроенные СУБД в Python Электронный ресурс].-URL: <https://habr.com/ru/company/ruvds/blog/514538/> (дата обращения 08.05.2022).

6. Руководство “Как подружить Python и базы данных SQL.” [Электронный ресурс].-URL: <https://proglib.io/p/kak-podruzhit-python-i-bazy-dannyh-sql-podrobnoe-rukovodstvo-2020-02-27> (дата обращения 08.05.2022).

7. Руководство по PyQt5 [Электронный ресурс].-URL: <https://python-scripts.com/pyqt5> (дата обращения 08.05.2022).

8. Руководство по PyQt6 [Электронный ресурс].-URL: <https://habr.com/ru/company/skillfactory/blog/599599/> (дата обращения 08.05.2022).

9. Установка PyQt6 [Электронный ресурс].-URL: <https://pypi.org/project/PyQt6/> (дата обращения 08.05.2022).

10. Работа с Python PyQT Framework [Электронный ресурс].-URL: <https://rukovodstvo.net/posts/id_514/> (дата обращения 08.05.2022).

# **ПРИЛОЖЕНИЕ А**

Листинг mine.py

**import** re

**import** requests

**import** sqlite3

**from** bs4 **import** BeautifulSoup **as** bsoup

**from** PyQt6 **import** QtCore**,** QtGui**,** QtWidgets

**import** pingspeed

**import** tablebd

**from** database **import** Database

**from** Parsik **import** Paarser

**class** **Ui\_FastProxy(object):**

**def** \_\_init\_\_**(**self**):**

**pass**

**def** setupUi**(**self**,** FastProxy**):**

FastProxy**.**setObjectName**(**"FastProxy"**)**

FastProxy**.**resize**(**531**,** 137**)**

font **=** QtGui**.**QFont**()**

font**.**setUnderline**(False)**

font**.**setStrikeOut**(False)**

FastProxy**.**setFont**(**font**)**

FastProxy**.**setAcceptDrops**(False)**

FastProxy**.**setStyleSheet**(**

"background-color: qradialgradient(spread:repeat, cx:0.5, cy:0.5, radius:0.077, fx:0.5, fy:0.5, stop:0 rgba(0, 169, 255, 147), stop:0.497326 rgba(0, 0, 0, 147), stop:1 rgba(0, 169, 255, 147));"**)**

FastProxy**.**setModal**(False)**

self**.**toolButton **=** QtWidgets**.**QToolButton**(**FastProxy**)**

self**.**toolButton**.**setGeometry**(**QtCore**.**QRect**(**30**,** 40**,** 141**,** 61**))**

font **=** QtGui**.**QFont**()**

font**.**setFamily**(**"Adobe Arabic"**)**

font**.**setPointSize**(**13**)**

font**.**setBold**(True)**

font**.**setItalic**(False)**

font**.**setWeight**(**75**)**

self**.**toolButton**.**setFont**(**font**)**

self**.**toolButton**.**setStyleSheet**(**

"background-color: qlineargradient(spread:pad, x1:0, y1:0, x2:1, y2:0, stop:0 rgba(9, 41, 4, 255), stop:0.085 rgba(2, 79, 0, 255), stop:0.19 rgba(50, 147, 22, 255), stop:0.275 rgba(236, 191, 49, 255), stop:0.39 rgba(243, 61, 34, 255), stop:0.555 rgba(135, 81, 60, 255), stop:0.667 rgba(121, 75, 255, 255), stop:0.825 rgba(164, 255, 244, 255), stop:0.885 rgba(104, 222, 71, 255), stop:1 rgba(93, 128, 0, 255));"**)**

self**.**toolButton**.**setObjectName**(**"toolButton"**)**

self**.**toolButton\_2 **=** QtWidgets**.**QToolButton**(**FastProxy**)**

self**.**toolButton\_2**.**setGeometry**(**QtCore**.**QRect**(**190**,** 40**,** 141**,** 61**))**

font **=** QtGui**.**QFont**()**

font**.**setFamily**(**"Adobe Arabic"**)**

font**.**setPointSize**(**14**)**

font**.**setBold**(True)**

font**.**setWeight**(**75**)**

self**.**toolButton\_2**.**setFont**(**font**)**

self**.**toolButton\_2**.**setStyleSheet**(**

"background-color: qlineargradient(spread:pad, x1:0, y1:0, x2:1, y2:0, stop:0 rgba(9, 41, 4, 255), stop:0.085 rgba(2, 79, 0, 255), stop:0.19 rgba(50, 147, 22, 255), stop:0.275 rgba(236, 191, 49, 255), stop:0.39 rgba(243, 61, 34, 255), stop:0.555 rgba(135, 81, 60, 255), stop:0.667 rgba(121, 75, 255, 255), stop:0.825 rgba(164, 255, 244, 255), stop:0.885 rgba(104, 222, 71, 255), stop:1 rgba(93, 128, 0, 255));"**)**

self**.**toolButton\_2**.**setObjectName**(**"toolButton\_2"**)**

self**.**toolButton\_3 **=** QtWidgets**.**QToolButton**(**FastProxy**)**

self**.**toolButton\_3**.**setGeometry**(**QtCore**.**QRect**(**360**,** 40**,** 141**,** 61**))**

font **=** QtGui**.**QFont**()**

font**.**setFamily**(**"Adobe Arabic"**)**

font**.**setPointSize**(**13**)**

font**.**setBold**(True)**

font**.**setItalic**(False)**

font**.**setUnderline**(False)**

font**.**setWeight**(**75**)**

self**.**toolButton\_3**.**setFont**(**font**)**

self**.**toolButton\_3**.**setStyleSheet**(**

"background-color: qlineargradient(spread:pad, x1:0, y1:0, x2:1, y2:0, stop:0 rgba(9, 41, 4, 255), stop:0.085 rgba(2, 79, 0, 255), stop:0.19 rgba(50, 147, 22, 255), stop:0.275 rgba(236, 191, 49, 255), stop:0.39 rgba(243, 61, 34, 255), stop:0.555 rgba(135, 81, 60, 255), stop:0.667 rgba(121, 75, 255, 255), stop:0.825 rgba(164, 255, 244, 255), stop:0.885 rgba(104, 222, 71, 255), stop:1 rgba(93, 128, 0, 255));"**)**

self**.**toolButton\_3**.**setObjectName**(**"toolButton\_3"**)**

self**.**retranslateUi**(**FastProxy**)**

QtCore**.**QMetaObject**.**connectSlotsByName**(**FastProxy**)**

self**.**add\_functions**()**

**def** retranslateUi**(**self**,** FastProxy**):**

\_translate **=** QtCore**.**QCoreApplication**.**translate

FastProxy**.**setWindowTitle**(**\_translate**(**"FastProxy"**,** "FastProxy"**))**

self**.**toolButton**.**setText**(**\_translate**(**"FastProxy"**,** "Показать"**))**

self**.**toolButton\_2**.**setText**(**\_translate**(**"FastProxy"**,** "Очистить"**))**

self**.**toolButton\_3**.**setText**(**\_translate**(**"FastProxy"**,** "Проверить"**))**

**def** add\_functions**(**self**):**

self**.**toolButton**.**clicked**.**connect**(**self**.**write\_bd**)**

self**.**toolButton**.**clicked**.**connect**(**self**.**Test**)**

self**.**toolButton\_2**.**clicked**.**connect**(**self**.**delete\_bd**)**

self**.**toolButton\_3**.**clicked**.**connect**(**self**.**pingvich**)**

**def** write\_bd**(**self**):**

urls **=** **{**'https://free.proxy-sale.com/'**,**

'http://foxtools.ru/Proxy'**,**

'https://best-proxies.ru/proxylist/free/'**,**

'https://www.freeproxy.world/'**,**

'https://www.proxy-list.download/HTTP'**}**

**for** url **in** urls**:**

Paarser**(**url**)**

db **=** Database**()**

rows **=** db**.**cursorObj**.**execute**(**'''SELECT

FROM users'''**).**fetchall**()**

i **=** 0

**for** row **in** rows**:**

**if** i **==** 10**:**

**break**

ip **=** row**[**0**]**

port **=** row**[**1**]**

**type** **=** row**[**2**]**

country **=** row**[**3**]**

speed **=** row**[**4**]**

**print(**ip**,**port**,type,**country**,**speed**)**

i**+=**1

db**.**close**()**

**def** delete\_bd**(**self**):**

d **=** Database**()**

d**.**delete**()**

**def** Test**(**self**):**

tablebd**.**window**.**show**()**

tablebd**.**window**.**setGeometry**(**950**,** 650**,** 550**,** 350**)**

**def** pingvich**(**self**):**

pingspeed**.**MainWindow**.**show**()**

FastProxy**.**hide**()**

**def** main**():**

**pass**

**if** \_\_name\_\_ **==** "\_\_main\_\_"**:**

main**()**

**import** sys

app **=** QtWidgets**.**QApplication**(**sys**.**argv**)**

FastProxy **=** QtWidgets**.**QDialog**()**

ui **=** Ui\_FastProxy**()**

ui**.**setupUi**(**FastProxy**)**

FastProxy**.**show**()**

sys**.exit(**app**.exec())**

Листинг Parsik.py

**import** re

**import** requests

**import** sqlite3

**from** bs4 **import** BeautifulSoup **as** bsoup

**from** PyQt6 **import** QtCore**,** QtGui**,** QtWidgets

**import** pingspeed

# import tablebd

**from** database **import** Database

**class** **Paarser:**

**def** \_\_init\_\_**(**self**,** url**):**

self**.**user\_ip\_list **=** **[]**

self**.**user\_country\_list **=** **[]**

self**.**user\_speed\_list **=** **[]**

self**.**user\_port\_list **=** **[]**

self**.**user\_type\_list **=** **[]**

**if** url **==** 'https://free.proxy-sale.com/'**:**

self**.**screb1**(**url**)**

**if** url **==** 'http://foxtools.ru/Proxy'**:**

self**.**screb2**(**url**)**

**if** url **==** 'https://best-proxies.ru/proxylist/free/'**:**

self**.**screb3**(**url**)**

**if** url **==** 'https://www.freeproxy.world/'**:**

self**.**screb4**(**url**)**

**if** url **==** 'https://www.proxy-list.download/HTTP'**:**

self**.**screb5**(**url**)**

**pass**

**def** clear**(**self**):**

self**.**user\_ip\_list **=** **[]**

self**.**user\_country\_list **=** **[]**

self**.**user\_speed\_list **=** **[]**

self**.**user\_port\_list **=** **[]**

self**.**user\_type\_list **=** **[]**

**def** screb1**(**self**,** url1**):**

#self.db.bd(user\_ip\_list, user\_port\_list, user\_type\_list, user\_country\_list, user\_speed\_list)

page **=** requests**.**get**(**url1**)**

result **=** bsoup**(**page**.**text**,** 'lxml'**)**

**for** el0 **in** result**.**find\_all**(**'td'**,** class\_**=**'ip'**):**

**for** el1 **in** el0**.**find\_all**(**'a'**):**

user\_IP **=** **((**el1**.**text**).**split**(**"\t"**)[**0**])**

self**.**user\_ip\_list**.**append**(**user\_IP**)**

**for** el00 **in** result**.**find\_all**(**'td'**,** class\_**=**'country'**):**

user\_country **=** **(**el00**.**text**.**split**()[**0**])**

self**.**user\_country\_list**.**append**(**user\_country**)**

**for** el00 **in** result**.**find\_all**(**'td'**):**

**for** el0 **in** result**.**find\_all**(**'div'**,** class\_**=**'progress-bar blue stripes'**):**

**for** el1 **in** el0**.**find\_all**(**'i'**):**

user\_speed **=** **(**el1**.**text**).**split**(**" мс"**)[**0**]**

**if** user\_speed **!=** ""**:**

self**.**user\_speed\_list**.**append**(**user\_speed**)**

**for** el0 **in** result**.**find\_all**(**'div'**,** class\_**=**'kc-elm kc-css-517022 kc-raw-code'**):**

el0 **=** re**.**sub**(**r'\s+'**,** ' '**,** **str(**el0**))**

\_start **=** **[**i **for** i **in** **range(len(**el0**))** **if** el0**.**startswith**(**'attrPorts = attrPorts + "," + '**,** i**)]**

\_end **=** **[**i **for** i **in** **range(len(**el0**))** **if** el0**.**startswith**(**'</script>'**,** i**)]**

**for** i **in** **range(len(**\_start**)):**

self**.**user\_port\_list**.**append**(int(**el0**[**\_start**[**i**]** **+** 30**:**\_end**[**i**]** **-** 2**]))**

**for** el0 **in** result**.**find\_all**(**'td'**):**

**for** el1 **in** el0**.**find\_all**(**'a'**):**

temp **=** **((**el1**.**text**).**split**(**'\t'**)[**0**])**

temp1 **=** **(**temp**.**split**(**'\n'**)[**0**])**

user\_type **=** **((**re**.**sub**(**'[^A-Z]'**,** ''**,** temp1**)))**

self**.**user\_type\_list**.**append**(**user\_type**)**

user\_type\_list **=** **[**i **for** i **in** self**.**user\_type\_list **if** i**]**

#db = Database().bd()

Database**().**bd**(**self**.**user\_ip\_list**,** self**.**user\_port\_list**,** user\_type\_list**,** self**.**user\_country\_list**,** self**.**user\_speed\_list**)**

self**.**clear**()**

**def** screb2**(**self**,**url2**):**

**iter** **=** **[]**

page **=** requests**.**get**(**url2**)**

result **=** bsoup**(**page**.**text**,** 'lxml'**)**

**for** ell1 **in** result**.**find\_all**(**'td'**):**

temp **=** **((**ell1**.**text**))**

**iter.**append**(**temp**)**

**for** i **in** **range(**1**,** **len(iter),** 8**):**

self**.**user\_ip\_list**.**append**(iter[**i**])**

**for** i **in** **range(**2**,** **len(iter),** 8**):**

self**.**user\_port\_list**.**append**(iter[**i**])**

**for** i **in** **range(**3**,** **len(iter),** 8**):**

self**.**user\_country\_list**.**append**(iter[**i**])**

**for** i **in** **range(**5**,** **len(iter),** 8**):**

tempp**=((**re**.**sub**(**'[^A-Z]'**,** ''**,iter[**i**])))**

self**.**user\_type\_list**.**append**(str(**tempp**))**

**for** i **in** **range(**6**,** **len(iter),** 8**):**

self**.**user\_speed\_list**.**append**(iter[**i**])**

Database**().**bd**(**self**.**user\_ip\_list**,** self**.**user\_port\_list**,** self**.**user\_type\_list**,** self**.**user\_country\_list**,**

self**.**user\_speed\_list**)**

self**.**clear**()**

**def** screb3**(**self**,** url3**):**

**iter** **=** **[]**

port **=** **[]**

ip **=** **[]**

country **=** **[]**

speed **=** **[]**

**type** **=** **[]**

page **=** requests**.**get**(**url3**)**

result **=** bsoup**(**page**.**text**,** 'lxml'**)**

**for** ell1 **in** result**.**find\_all**(**'td'**):**

temp **=** **((**ell1**.**text**))**

**iter.**append**(**temp**)**

# print(iter)

**for** i **in** **range(**1**,** **len(iter),** 8**):**

port**.**append**(iter[**i**])**

port **=** **[**line**.**strip**()** **for** line **in** port**]**

self**.**user\_port\_list**.**extend**(**port**[:**10**])**

**for** i **in** **range(**0**,** **len(iter),** 8**):**

ip**.**append**(iter[**i**])**

ip **=** **[**line**.**strip**()** **for** line **in** ip**]**

self**.**user\_ip\_list**.**extend**(**ip**[:**10**])**

**for** i **in** **range(**2**,** **len(iter),** 8**):**

country**.**append**(iter[**i**])**

country **=** **[**line**.**strip**()** **for** line **in** country**]**

self**.**user\_country\_list**.**extend**(**country**[:**10**])**

**for** i **in** **range(**4**,** **len(iter),** 8**):**

speed**.**append**(iter[**i**])**

speed **=** **[**line**.**strip**()** **for** line **in** speed**]**

self**.**user\_speed\_list**.**extend**(**speed**[:**10**])**

**for** i **in** **range(**5**,** **len(iter),** 8**):**

**type.**append**(iter[**i**])**

**type** **=** **[**line**.**strip**()** **for** line **in** **type]**

self**.**user\_type\_list**.**extend**(type[:**10**])**

Database**().**bd**(**self**.**user\_ip\_list**,** self**.**user\_port\_list**,** self**.**user\_type\_list**,** self**.**user\_country\_list**,**

self**.**user\_speed\_list**)**

self**.**clear**()**

**def** screb4**(**self**,** url4**):**

**iter** **=** **[]**

port **=** **[]**

ip **=** **[]**

country **=** **[]**

speed **=** **[]**

**type** **=** **[]**

page **=** requests**.**get**(**url4**)**

result **=** bsoup**(**page**.**text**,** 'lxml'**)**

**for** el2 **in** result**.**find\_all**(**'tr'**):**

**for** el3 **in** el2**.**find\_all**(**'td'**):**

tabl **=** **(**el3**.**text**.**strip**())**

**iter.**append**(**tabl**)**

**for** i **in** **range(**1**,** **len(iter),** 8**):**

port**.**append**(iter[**i**])**

port **=** **[**line**.**strip**()** **for** line **in** port**]**

self**.**user\_port\_list**.**extend**(**port**[:**10**])**

**for** i **in** **range(**0**,** **len(iter),** 8**):**

ip**.**append**(iter[**i**])**

ip **=** **[**line**.**strip**()** **for** line **in** ip**]**

self**.**user\_ip\_list**.**extend**(**ip**[:**10**])**

**for** i **in** **range(**2**,** **len(iter),** 8**):**

country**.**append**(iter[**i**])**

country **=** **[**line**.**strip**()** **for** line **in** country**]**

self**.**user\_country\_list**.**extend**(**country**[:**10**])**

**for** i **in** **range(**4**,** **len(iter),** 8**):**

speed**.**append**(iter[**i**])**

speed **=** **[**line**.**strip**()** **for** line **in** speed**]**

self**.**user\_speed\_list**.**extend**(**speed**[:**10**])**

**for** i **in** **range(**5**,** **len(iter),** 8**):**

**type.**append**(iter[**i**])**

**type** **=** **[**line**.**strip**()** **for** line **in** **type]**

self**.**user\_type\_list**.**extend**(type[:**10**])**

Database**().**bd**(**self**.**user\_ip\_list**,** self**.**user\_port\_list**,** self**.**user\_type\_list**,** self**.**user\_country\_list**,**

self**.**user\_speed\_list**)**

self**.**clear**()**

**def** screb5**(**self**,**url5**):**

**iter** **=** **[]**

port **=** **[]**

ip **=** **[]**

country **=** **[]**

speed **=** **[]**

**type** **=** **[]**

page **=** requests**.**get**(**url5**)**

result **=** bsoup**(**page**.**text**,** 'lxml'**)**

**for** el2 **in** result**.**find\_all**(**'td'**):**

tabl **=** **(**el2**.**text**.**strip**())**

**iter.**append**(**tabl**)**

**for** i **in** **range(**1**,** **len(iter),** 5**):**

port**.**append**(iter[**i**])**

port **=** **[**line**.**strip**()** **for** line **in** port**]**

self**.**user\_port\_list**.**extend**(**port**[:**10**])**

**for** i **in** **range(**0**,** **len(iter),** 5**):**

ip**.**append**(iter[**i**])**

ip **=** **[**line**.**strip**()** **for** line **in** ip**]**

self**.**user\_ip\_list**.**extend**(**ip**[:**10**])**

**for** i **in** **range(**0**,** 10**):**

**type.**append**(**'HTTP'**)**

self**.**user\_type\_list**.**extend**(type)**

**for** i **in** **range(**3**,** **len(iter),** 5**):**

country**.**append**(iter[**i**])**

country **=** **[**line**.**strip**()** **for** line **in** country**]**

self**.**user\_country\_list**.**extend**(**country**[:**10**])**

**for** i **in** **range(**4**,** **len(iter),** 5**):**

speed**.**append**(iter[**i**])**

speed **=** **[**line**.**strip**()** **for** line **in** speed**]**

self**.**user\_speed\_list**.**extend**(**speed**[:**10**])**

Database**().**bd**(**self**.**user\_ip\_list**,** self**.**user\_port\_list**,** self**.**user\_type\_list**,** self**.**user\_country\_list**,**

self**.**user\_speed\_list**)**

self**.**clear**()**

Листинг database.py

**import** sqlite3

**class** **Database:**

**def** \_\_init\_\_**(**self**):**

self**.**con **=** sqlite3**.**connect**(**'mydatabase.db'**)**

self**.**cursorObj **=** self**.**con**.**cursor**()**

self**.**cursorObj**.**execute**(**"""CREATE TABLE IF NOT EXISTS users (

IP TEXT,

port TEXT,

type TEXT,

country TEXT,

speed TEXT

)"""**)**

self**.**con**.**commit**()**

**def** bd**(**self**,** user\_ip\_list**,** user\_port\_list**,** user\_type\_list**,** user\_country\_list**,** user\_speed\_list**):**

**for** i **in** **range(**0**,** **len(**user\_ip\_list**),** 1**):**

self**.**con**.**execute**(**"INSERT INTO users VALUES (?, ?, ?, ?, ?)"**,**

**(**user\_ip\_list**[**i**],** user\_port\_list**[**i**],** user\_type\_list**[**i**],** user\_country\_list**[**i**],**

user\_speed\_list**[**i**]))**

self**.**con**.**commit**()**

self**.**con**.**close**()**

**def** delete**(**self**):**

rows **=** self**.**cursorObj**.**execute**(**'''SELECT FROM users'''**).**fetchall**()**

**if** rows **!=** **[]:**

self**.**cursorObj**.**execute**(**'DELETE FROM users'**)**

self**.**con**.**commit**()**

**print(**"Очищенно"**)**

**else:**

**print(**'Данных нет'**)**

**def** close**(**self**):**

self**.**con**.**close**()**

Листинг tablebd.py

**import** sys

**from** PyQt6 **import** QtCore**,** QtGui**,** QtWidgets

**from** PyQt6**.**QtCore **import** Qt

**from** database **import** Database

**import** Parsik

**class** **TableModel(**QtCore**.**QAbstractTableModel**):**

**def** \_\_init\_\_**(**self**,** data**):**

**super(**TableModel**,** self**).**\_\_init\_\_**()**

self**.**\_data **=** data

**def** data**(**self**,** index**,** role**):**

**if** role **==** Qt**.**ItemDataRole**.**DisplayRole**:**

# See below for the nested-list data structure.

# .row() indexes into the outer list,

# .column() indexes into the sub-list

**return** self**.**\_data**[**index**.**row**()][**index**.**column**()]**

**def** rowCount**(**self**,** index**):**

# The length of the outer list.

**return** **len(**self**.**\_data**)**

**def** columnCount**(**self**,** index**):**

# The following takes the first sub-list, and returns

# the length (only works if all rows are an equal length)

**return** **len(**self**.**\_data**[**0**])**

**class** **MainWindow(**QtWidgets**.**QMainWindow**):**

**def** \_\_init\_\_**(**self**):**

**super().**\_\_init\_\_**()**

db **=** Database**()**

row **=** db**.**cursorObj**.**execute**(**'''SELECT

FROM users'''**).**fetchone**()**

**if** row **is** **not** **None:**

self**.**other\_func**(**db**)**

**else:**

Parsik**.**Paarser

**def** other\_func**(**self**,**db**):**

self**.**table **=** QtWidgets**.**QTableView**()**

rows **=** db**.**cursorObj**.**execute**(**'''SELECT

FROM users'''**).**fetchall**()**

**if** rows **is** **not** **None:**

data **=** **[**rows**[**0**],**rows**[**1**],**rows**[**2**],**rows**[**3**],**rows**[**4**],**rows**[**5**],**rows**[**6**],**rows**[**7**],**rows**[**8**],**rows**[**9**]]**

self**.**model **=** TableModel**(**data**)**

self**.**table**.**setModel**(**self**.**model**)**

self**.**setCentralWidget**(**self**.**table**)**

app**=**QtWidgets**.**QApplication**(**sys**.**argv**)**

window**=**MainWindow**()**

window**.**show**()**

app**.exec()**

Листинг pingspeed.py

**from** PyQt6 **import** QtCore**,** QtWidgets

**import** subprocess

**class** **Ui\_MainWindow(object):**

**def** setupUi**(**self**,** MainWindow**):**

MainWindow**.**setObjectName**(**"MainWindow"**)**

MainWindow**.**resize**(**656**,** 274**)**

MainWindow**.**setStyleSheet**(**"\n"

"background-color: rgb(255, 255, 166);"**)**

self**.**centralwidget **=** QtWidgets**.**QWidget**(**MainWindow**)**

self**.**centralwidget**.**setObjectName**(**"centralwidget"**)**

self**.**textEdit **=** QtWidgets**.**QTextEdit**(**self**.**centralwidget**)**

self**.**textEdit**.**setGeometry**(**QtCore**.**QRect**(**10**,** 0**,** 311**,** 31**))**

self**.**textEdit**.**setObjectName**(**"textEdit"**)**

self**.**lineEdit **=** QtWidgets**.**QLineEdit**(**self**.**centralwidget**)**

self**.**lineEdit**.**setGeometry**(**QtCore**.**QRect**(**330**,** 1**,** 201**,** 31**))**

self**.**lineEdit**.**setStyleSheet**(**"background-color: rgb(255, 174, 147);\n"

"font: 75 9pt \"MS Shell Dlg 2\";\n"

"text-decoration: underline;"**)**

self**.**lineEdit**.**setObjectName**(**"lineEdit"**)**

self**.**pushButton **=** QtWidgets**.**QPushButton**(**self**.**centralwidget**)**

self**.**pushButton**.**setGeometry**(**QtCore**.**QRect**(**540**,** 0**,** 101**,** 31**))**

self**.**pushButton**.**setStyleSheet**(**"background-color: rgb(85, 170, 0);"**)**

self**.**pushButton**.**setObjectName**(**"pushButton"**)**

self**.**textEdit\_2 **=** QtWidgets**.**QTextEdit**(**self**.**centralwidget**)**

self**.**textEdit\_2**.**setGeometry**(**QtCore**.**QRect**(**10**,** 50**,** 631**,** 161**))**

self**.**textEdit\_2**.**setObjectName**(**"textEdit\_2"**)**

MainWindow**.**setCentralWidget**(**self**.**centralwidget**)**

self**.**menubar **=** QtWidgets**.**QMenuBar**(**MainWindow**)**

self**.**menubar**.**setGeometry**(**QtCore**.**QRect**(**0**,** 0**,** 656**,** 26**))**

self**.**menubar**.**setObjectName**(**"menubar"**)**

MainWindow**.**setMenuBar**(**self**.**menubar**)**

self**.**statusbar **=** QtWidgets**.**QStatusBar**(**MainWindow**)**

self**.**statusbar**.**setObjectName**(**"statusbar"**)**

MainWindow**.**setStatusBar**(**self**.**statusbar**)**

self**.**retranslateUi**(**MainWindow**)**

QtCore**.**QMetaObject**.**connectSlotsByName**(**MainWindow**)**

self**.**ad\_functions**()**

**def** retranslateUi**(**self**,** MainWindow**):**

\_translate **=** QtCore**.**QCoreApplication**.**translate

MainWindow**.**setWindowTitle**(**\_translate**(**"MainWindow"**,** "MainWindow"**))**

self**.**textEdit**.**setHtml**(**\_translate**(**"MainWindow"**,** "<!DOCTYPE HTML PUBLIC \"-//W3C//DTD HTML 4.0//EN\" \"http://www.w3.org/TR/REC-html40/strict.dtd\">\n"

"<html><head><meta name=\"qrichtext\" content=\"1\" /><style type=\"text/css\">\n"

"p, li { white-space: pre-wrap; }\n"

"</style></head><body style=\" font-family:\'MS Shell Dlg 2\'; font-size:7.8pt; font-weight:400; font-style:normal;\">\n"

"<p style=\" margin-top:0px; margin-bottom:0px; margin-left:0px; margin-right:0px; -qt-block-indent:0; text-indent:0px;\"><span style=\" font-size:10pt; font-weight:600; color:#337309;\">Введите данные для проверки-&gt;</span></p></body></html>"**))**

self**.**pushButton**.**setText**(**\_translate**(**"MainWindow"**,** "Готово"**))**

**def** ad\_functions**(**self**):**

self**.**pushButton**.**clicked**.**connect**(lambda:** self**.**textEdit\_2**.**setText**(**self**.**get\_ping**(**self**.**lineEdit**.**text**())))**

**def** get\_ping**(**self**,**ip**):**

result **=** subprocess**.**run**([**'ping'**,** f'{ip}'**],** stdout**=**subprocess**.**PIPE**)**

output **=** result**.**stdout**.**decode**(**'windows-1251'**,** errors**=**'ignore'**).**split**(**'\n'**)**

output **=** output**[len(**output**)** **-** 2**].**split**(**' '**)**

output **=** **int(**output**[len(**output**)** **-** 2**])**

pinga**=**'ping==>'**+str(**output**)+**'ms'

**return** pinga

**if** \_\_name\_\_ **==** "\_\_main\_\_"**:**

**import** sys

app **=** QtWidgets**.**QApplication**(**sys**.**argv**)**

MainWindow **=** QtWidgets**.**QMainWindow**()**

ui **=** Ui\_MainWindow**()**

ui**.**setupUi**(**MainWindow**)**

MainWindow**.**show**()**

sys**.exit(**app**.exec())**

**ПРИЛОЖЕНИЕ Б**

Фейк 1.

О чем фейк

В росте цен на бензин в США обвиняют лично Владимира Путина. Такие сообщения стали публиковать популярные TikTok блогеры.

Как на самом деле

Изданию Fox Business стало известноо том, что администрация Байдена на прошлой неделе подключила 30 популярных блогеров для внедрения тезисов о позиции США по ситуации на Украине и обвинения Владимира Путина в повышении цен на бензин.

В качестве примера приводится 18-летняя Элли Цайлер с 10 млн подписчиков. Блогер, которая раньше выкладывала танцы, рассказы о моде и косметике, вдруг поделилась с аудиториейсвоей версией того, почему в стране дорожает бензин.

Судя по всему, власти привлекли блогеров после того, как на автозаправках рядом с экранами на стоимость топлива стали появляться стикеры с изображениями Байдена и подписью «Я это сделал!» и вице-премьера Камилы Харрис со словами «А я помогала».

Энергетическая политика Байдена и запрет импорта нефти из России привел к топливному кризису внутри страны, считают недовольные американские водители.

Фейк 2.

О чем фейк

Россию ждет тотальный дефицит лекарств. Такие сообщения появились в ряде СМИ.

Как на самом деле

Не спешите бежать в аптеку. Во-первых, все крупные аптечные сети имеют крупные склады, которые могут обеспечивать розницу препаратами на месяцы вперед. Во-вторых, как отметил гендиректор Ассоциации Российских фармацевтических производителей Виктор Дмитриев, производственных площадок для изготовления лекарств в России даже боьше, чем того требуется для нормального функционирования рынка.

О нехватке импортных препаратов речи также не идет — ведущие зарубежные фармкомпании уже официально выразили готовность продолжить поставки лекарств в Россию.

Необходимости закупать лекарства в прок нет в принципе. Такое «вложение» [обернется](https://nsn.fm/society/srednik-chek-vyros-v-pyat-raz-rossiyanam-posovetovali-ne-zakupatsya-lekarstvami-vprok) пустой тратой денег, поскольку все лекарства имеют ограниченный срок годности. Напомним, что подобная ситуация наблюдалась в начале пандемии, но и тогда, даже в условиях ажиотажного спроса, препараты не исчезли с аптечных прилавков.

Предположительно, слухи о массовом дефиците лекарств появились после того, как в некоторых регионах страны стали возникать трудности с покупкой препаратов для лечения заболеваний щитовидной железы «Эутирокса» и «L-Тироксина». Однако трудности оказались временными — до конца марта эти лекарства вернутся в аптеки.

Фейк 3.

Фейк: Сергей Брилев уходит с телеканала «Россия»

О чем фейк

Заместитель генерального директора телеканала «Россия» по специальным информационным проектам Сергей Брилев уходит с канала и переезжает из страны. Об этом со ссылкой на журналиста Романа Супера сообщают украинские телеграм-каналы.

Как на самом деле:

Телеведущий Сергей Брилев продолжает работу на телеканале, сейчас он находится в командировке в Бразилии. Журналист лично сообщил об этом агентству ТАСС.

«Я в Бразилии, куда сегодня ночью перелетел из Эмиратов и Армении. Только что был в эфире «России-24», сейчас надо дальше готовить съемки», — сказал Брилев.

Сейчас вбрасывается немало фейков о том, что журналисты государственных каналов в России уезжают из страны. Рекомендуем доверять официальной информации самих медиа и словам самих работников СМИ.

![рилпруф](https://войнасфейками.рф/wp-content/uploads/2022/03/photo\_2022-03-15\_21-02-32-2-768x460.jpg)