ГБОУ СОШ №139 с углубленным изучением математики
г. Санкт-Петербург

Проектно-исследовательская работа

по информатике
На тему:
“Антивирус на базе языка Python”

Работа ученика 10 А класса:
Пак Чер Хо
Руководитель проекта:
Штраус Никита Алексеевич

Подпись руководителя проекта
_____(_____)

г. Санкт-Петербург,
2021

Паспорт проекта

Название проекта: Антивирус на базе языка Python

Руководитель проекта: Штраус Никита Алексеевич

Автор проекта: Пак Чер Хо

Учебная дисциплина: Информатика

Тип проекта: Практико-ориентированный

Цель работы: Создать антивирус для защиты компьютера на базе языка Python

Результат проекта: Работающий антивирус, созданный с помощью языка Python, на операционной системе Windows

Содержание проекта

ВВЕДЕНИЕ

Актуальность проекта:

Многие из нас наверняка сталкивались с такой проблемой, как компьютерные вирусы. Они проникают в компьютер с целью получить личные данные пользователя такие как: личная информация, пароли, фото, скрытые файлы. Или ухудшают его работу, продвигая ненужную рекламу и устанавливая приложения.

Вирус внедряется в код других программ, системные области памяти, загрузочные сектора и распространяет свой код по разнообразным средам компьютера. Если вовремя его не обезвредить, то это может привести к полному заражению компьютера, что чревато последствиями.

Для борьбы с компьютерными вирусами был создан антивирус. Его главная цель обнаружить вирусы и вредоносные программы на компьютере и обезвредить их с дальнейшим предотвращением заражения файлов. С его помощью можно безопасно открывать файлы и сайты, не беспокоясь подцепить вирус.

На каждом компьютере должен быть антивирус, ведь он необходим для его безопасной работы. Именно поэтому автор решил сделать антивирус, на языке программирования Python, способный защитить его компьютер.

Задачи проекта:

1. Ознакомиться с программой ’Антивирус’
2. Ознакомиться с языком Python
3. Написать антивирус на языке программирования Python

4. Проверить работу программы и включить его в работу ПК.

ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

Глава 1. Что такое антивирус?

1.1. История появления вирусов и антивирусов

Антивирусные программы – специализированные программы, направленные на обнаружение и обезвреживание вирусов, для безопасной работы компьютера.

Их история начинается с причины их появления – компьютерных вирусов. Первый компьютерный вирус был создан в 1960 г., но в то время не было среды для передачи информации, поэтому компьютер, на котором он был создан, стал его первой и последней жертвой. Всё это осталось бы интересным экспериментом, не появись интернет на свет.

Уже в 1975 г. в сети Telenet, был разослан первый сетевой вирус под названием “The Creeper” и в противовес ему так же была создана первая антивирусная программа “The Reeper”. Тогда компьютерные вирусы не представляли большой угрозы, так как их приток был невероятно низким, и люди даже не задумывались осваивать новую область программирования. Но уже тогда американский информатик Ф. Коэн начал проводить опыты с ПО, которое может размножаться и делать себе подобных, и в 1984 г. он выступил на конференции безопасности в Соединенных Штатах, высказывая своё мнение по поводу новой угрозы в сфере программирования.

В то же время в Пакистане браться Амнджад создали, сами того не понимая, компьютерный вирус. Всё началось с того, что они продавали ПО и заметили, что кто-то незаконно распространяет их продукт бесплатно. Они могли потерять деньги, заработанные тяжелым трудом, и поэтому написали программу ’The Brain’ и внедрили её в свою работу. Данная программа

начинала работать копировании файлов. Это стало началом рождения будущих вирусов. Новоиспечённый вирус резко перешёл границы страны и поверг весь мир в шок.

Так, в 1987 г. вышла книга ’ **Computer Viruses: A High Tech Disease** ’ о вирусах и способах борьбы с ними. Её написал Ральф Бюргер, к которому попала копия, известного в то время вируса – Vienna. Данная литература стала толчком к созданию сотен и тысяч вирусов, так как объясняла принцип работы вирусов. С этого момента люди поняли, что необходимо создавать программы, способные бороться с ними, тем самым леча устройство.

Первые антивирусы сильно отличаются от нынешних. Фактически, они были одноразовыми программами, направленными на какой-то определённый вирус, да и передача данных программ была достаточно дорогой и долгой, так как антивирусы записывались на дискеты и рассылались покупателям. Но в 1989 г. на свет выходят множество антивирусов таких как: McAfee VIRUSCAN, F-Prot, ThunderBYTE, Norman Virus Control. Проблема компьютерных вирусов становилось всё более глобальной и привлекала внимание мировых гигантов. Так, в октябре того же года на свет вышел антивирус от IBM, получивший название Virscan.

Вирусы продолжали развиваться, дошло до того, что в 1990 г. появились так называемые полиморфные вирусы, в которых отсутствовал постоянный код. Но у них был недостаток, при расшифровке кода они всегда имели одинаковую структуру, за счёт этого появился первый антивирус с эмулятором, позволяющим расшифровать и излечить полиморфный вирус. Он получил название AVP.

В конце 1992 г. кол-во вирусов насчитывало тысячи. И разработчики антивирусов начали задумываться, как сделать так, чтобы антивирус мог сам обнаруживать неизвестные вирусы и запоминать их. Первым кто смог этого

достичь был наш соотечественник Евгений Сусликов и его Lie Detector. Такие антивирусы получили название эвристические анализаторы.

Вторым этапом было появление скрипт-вирусов. Прославились они за счёт того, что у них был открытый исходный код, что предопределило появление на свет новых модификаций вирусов. Одним из таких был скрипт-вирус LoveLetter. 5 мая он попал в книгу рекордов Гиннеса, как самая большая компьютерная эпидемия.

Но уже в 21 веке появились антивирусы, способные защищать компьютер в режиме реального времени и термин ’компьютерный вирус’ как таковой исчез, но появились вредоносные программы, которые охотятся за личными данные на вашем устройстве.

Таким образом, антивирусы, под действием всё новых и новых вирусов, формировались в течение долгого времени. Так мы и получили известные нам сейчас антивирусные программы, которые могут защитить нас от любого вируса.

1.2. Классификация антивирусов

Антивирусы делятся по трём категориям:

По способу блокирования:

1. ***Программные*** - защита осуществляется на операционной системе
2. ***Аппаратные*** – защита осуществляется на веб-шлюзе

По признаку размещения в оперативной памяти:

1. ***Резидентные*** – запускаются вместе с операционной системой и наблюдают за системой в реальном времени
2. ***Нерезидентные*** – запускаются по запросу пользователя, либо по расписанию

По способу защиты от вирусов:

1. ***Детекторы*** – обнаруживают вирусы в ОЗУ и на внешних носителях, сообщая о них пользователю при обнаружении.
2. ***Доктора*** – обнаруживают заражённые файлы и сразу же лечат их, удаляя тело вируса и возвращая файлы в исходное состояние. В первую очередь, они уничтожают вирус в оперативной памяти, а уже потом лечат файлы. Также выделяют полифаги, целью которых является удаление больших групп вирусов.
3. ***Иммунизаторы*** – ’Вакцинируют’ систему и файлы, что защищает их от вирусов.
4. ***Ревизоры*** – Запоминают исходное состояние файлов, системы, папок, а затем сравнивает их на постоянной основе, тем самым выявляя изменения при появлении вирусов. Они являются самыми надёжными среди способов защиты.

5. ***Фильтры*** – резидентные программы, обнаруживающие подозрительный действия во время работы компьютера.

Также существует 4 метода защиты от вирусов:

1. **Метод сигнатур –** основан на поиске паттерна (сигнатуры) кода, характерного для определённого вируса. При обнаружении вируса, загружает его код в специальную базу, в которой вирус обезвреживается. Плюсом данного метода является низкий процент ложны срабатываний, а главным минусом – невозможность находить новые вирусы из-за ограниченной базы.
2. **Метод контроля целостности –** основан на выслеживании неожиданных изменений данных на диске. Вирус всегда оставляет след своего пребывания тем самым давая возможность антивирусу отследить его и обезвредить. Факт изменения данных легко находится путём сравнения контрольной суммы, заранее подсчитанной для исходного кода. Данный метод работает быстрее метода сканирования сигнатур, поскольку побайтового сравнения кодовых фрагментов медленнее, кроме того он позволяет обнаруживать следы деятельности любых, в том числе неизвестных, вирусов, для которых в базе данных еще нет сигнатур.
3. **Метод сканирования подозрительных команд –** основан на выявлении подозрительных команд или признаков подозрительных кодовых действий в сканируемом файле. При обнаружении которых делается более тщательная проверка, после которой вирус, при его наличии, удаляется. Данный метод быстрый и эффективный, но не способен находить новые вирусы
4. **Метод отслеживания поведения программ –** основан на отслеживании действий запущенных программ. Данный метод

зачастую требует активного вмешательства пользователя в работу антивируса, так как необходимо принимать решения по поводу предупреждений от антивируса. Частота ложных срабатываний при превышении определённого порога делает этот метод неэффективным.

Глава 2. Язык программирования Python

2.1 История создания

В 1982 г. Гвидо ван Россум окончил университет и попал в команду разработчиков CWI (Амстердамского университета информатики и математики). До 1986 г. Он занимался проектированием языка ABC, который в будущем стал основой для Python. ABC задумывался как инструмент для пользователей, которые до этого не программировали и не разбирались в этой сфере деятельности. Он должен был быть простым и удобным, языком, на котором легко учиться.

Но в 1987 г. проект ABC закрылся. По мнению Гвидо, причиной стало отсутствие доступного интернета, так как язык медленно распространялся и не получал обратной связи от пользователей.

После неудачного опыта, Гвидо оценил силу общества. Он понял, что язык программирования не может существовать без активных пользователей, дающих обратную связь. — нужна система, в которой каждый разработчик будет предлагать улучшения или голосовать за предложения других участников сообщества. Так в будущем, в Python появился PEP – индекс – регламент по внесению изменений в структуру и синтаксис языка. Любой разработчик может предложить улучшения для Python сообщества.

1989 г. Гвидо начал мини проект со своими коллегами из CWI и уже 20 февраля 1991 г. был создан дистрибутив и опубликован код языка Python через сеть Usenet. Первый прототип состоял из простой виртуальной машины, парсера и среды выполнения. Главное преимущество заключалась в том, что язык предлагал гибкую модель расширяемости — то есть, помимо

стандартных возможностей, каждый программист мог самостоятельно добавить в систему нужные типы объектов. Так появилась версия 0.9.0

29 июня 1994 г. На форуме вышла статья, затронувшая проблему зависимости Python сообщества от решений Гвидо – автор высказался, что крупные компании опасаются использовать технологии, привязанные к одному человеку. Так, совместно с автором статьи, Гвидо создал Python Software Foundation – некоммерческую организацию, которая отвечает за развитие и защиту Python.

Так язык продолжил развиваться и уже в 2008 г. появилась 3 версия Python, используемая разработчиками по сей день.

2.2 Характеристики языка Python

Интерпретируемый. Это значит, что код в Python не переводится в машинный код, а выполняется программой-интерпретатором. Благодаря этому код в Python запустится на любой платформе, на которой установлен интерпретатор. Из минусов — код обрабатывается не заранее, а в процессе, а это снижает скорость;

Объектно-ориентированный. Объектно-ориентированный язык — это язык, созданный по модели объектно-ориентированного программирования *(подробнее* здесь*)*. В ней существуют классы и объекты. **Классы —** это типы данных, а **объекты —** представители классов. Вы создаёте их сами, даёте названия и присваиваете им свойства и операции, которые с ними можно выполнять. Это не значит, что на Python вы можете программировать только по этой модели: **он поддерживает и функциональное, и императивное программирование,** и другие.

Со строгой динамической типизацией. Это значит, что типы переменных определяются после запуска программы, а не при компиляции, а сочетать в выражениях разные типы нельзя. Несмотря на строгую типизацию, Python позволяет больше, чем некоторые другие языки, — например, вы не можете сложить строку (*string*) с целым числом (*int*), зато десятичное число (*float*) с целым — можете.

Преимущества:

1. Хорошо подходит для новичков.
2. Простой минималистичный синтаксис: код легко писать, читать и поддерживать.
3. Большая стандартная библиотека и много дополнительных библиотек.
4. Большой выбор фреймворков.
5. Поддерживает объектно-ориентированное программирование и другие парадигмы.
6. Кроссплатформенность и поддержка почти всех современных систем.

Недостатки:

1. Низкая скорость.
2. Плохо подходит для разработки мобильных приложений.

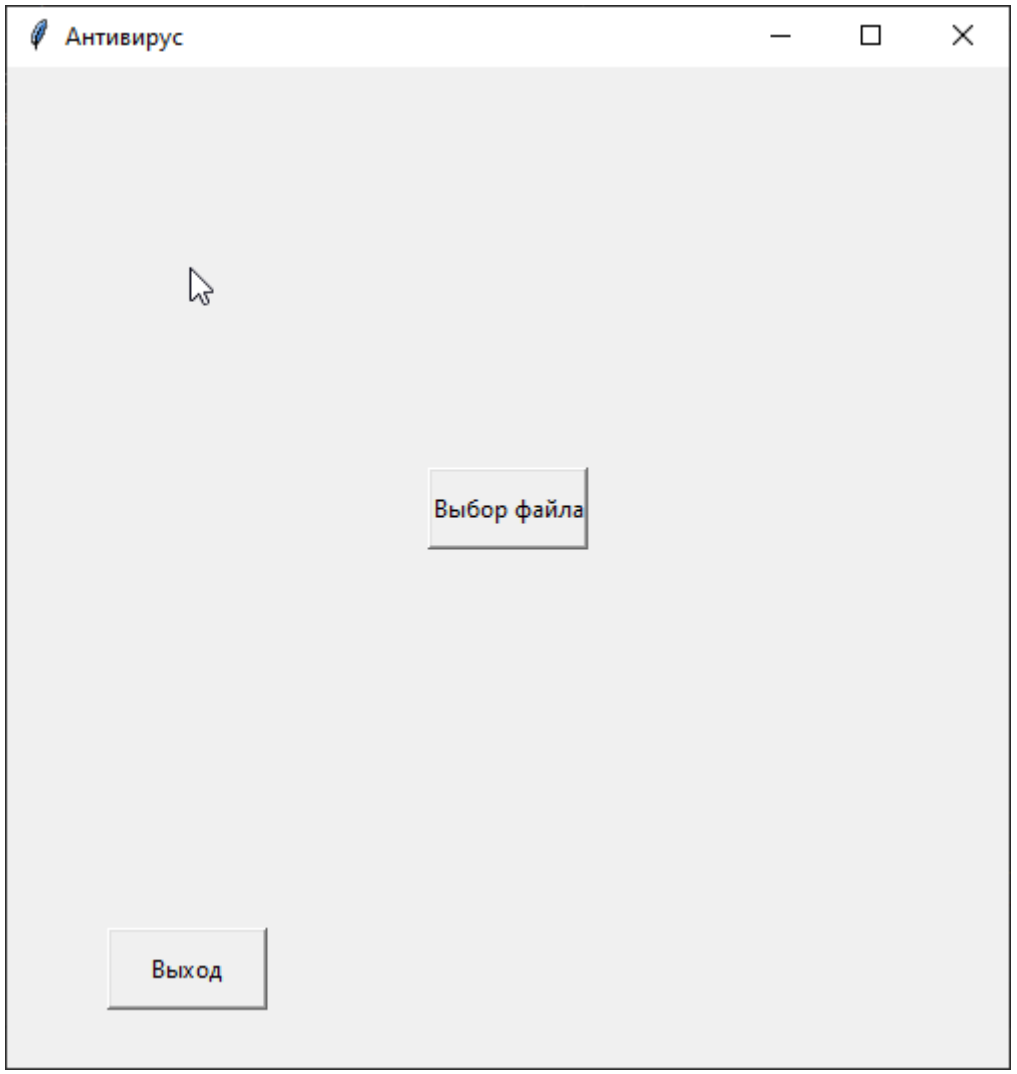
3. Из-за динамической типизации выше вероятность ошибки при запуске, нужно больше тестов.
4. Не подходит для работы с памятью на низком уровне.

ПРАКТИЧЕСКАЯ ЧАСТЬ

Глава 1. Создание антивируса

1.1 Методы и функции, используемые в создании

Конструктор __init__ создаёт окно:




```
def __init__(self, width, height, title="Antivirus"):
    self.root = Tk()
    self.root.title(title)
    self.root.geometry(f'{width}x{height}+200+200')
```

код конструктора __init_

За кнопки на данном окне отвечают функция create_menu:

```
Button(self.root, height=2, width=10, text="Выход",
command=self.close).place(x=50, y=430)

Button(self.root, text="Выбор файла", height=2, width=10,
command=browsefiles).pack(anchor='center',

pady=200)
```

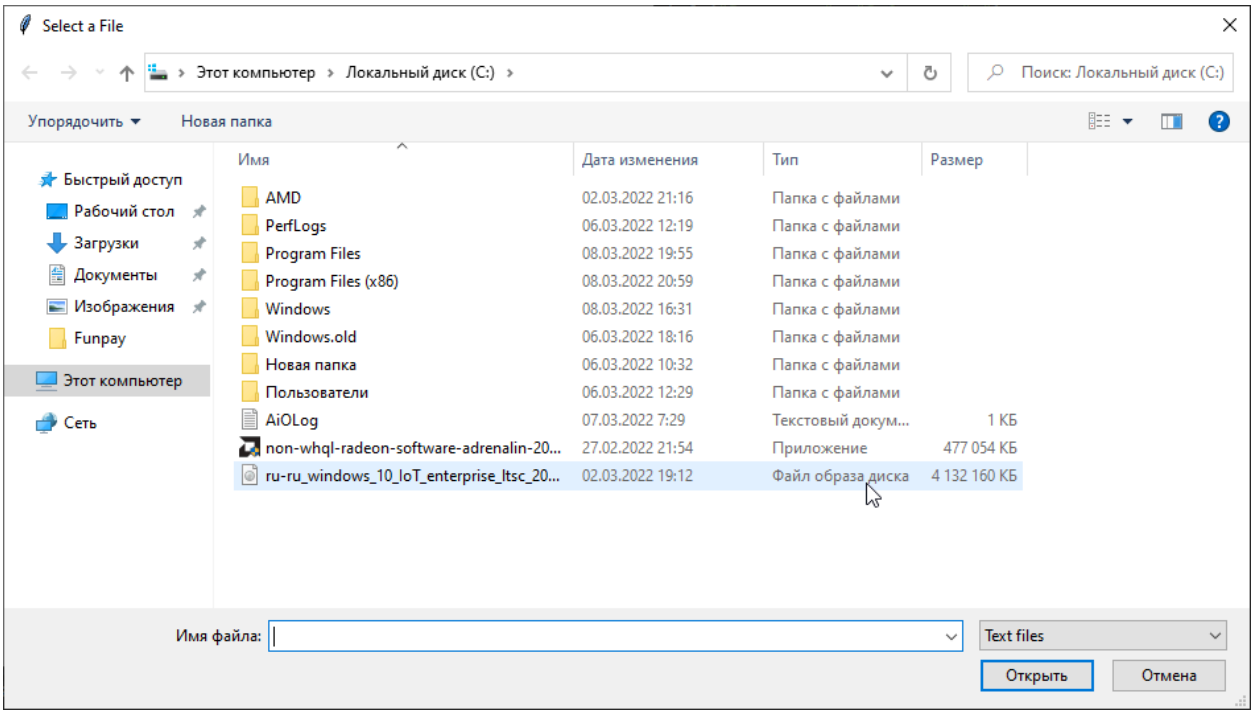
код функции create_menu

Кнопка ’Выход’ с помощью функции close спрашивает – закрыть ли программу, если да то закрывает, если нет, то программа продолжает работать:

```
def close(self):
    if messagebox.askyesno('Выход', 'Вы уверены?'):
        self.root.destroy()
```

код функции close

Кнопка ’Выбор файла’ с помощью функции browsefiles позволяет выбрать файл и проверить его на наличие вирусов с помощью трёх этапов:



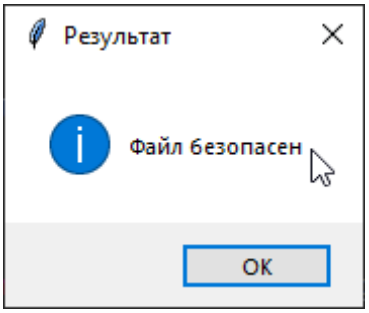

```
def browsefiles():
    filename = filedialog.askopenfilename(initialdir="/",
                                          title="Select a File",
                                          filetypes=(("Text files",
                                                      "*.*"),
                                                     ("all files",
                                                      "*.*")))


    scan_sha256(filename)
    scan_md5(filename)
    scan(filename)
```

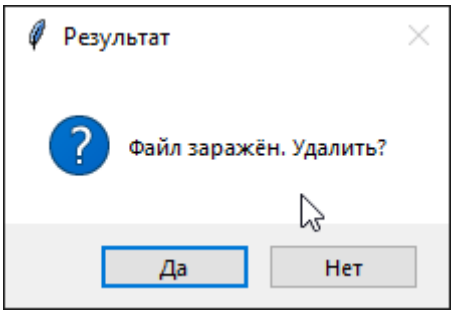
код функции browsefiles

Проверка выполняется с помощью функции scan, вызывающей результат проверки, если файл безопасен, то вызовется окно, что файл безопасен, если заражён, то вызовется окно с вопросом об удалении файла, если ответить да, то файл удалится, если нет, то программа продолжит работу:

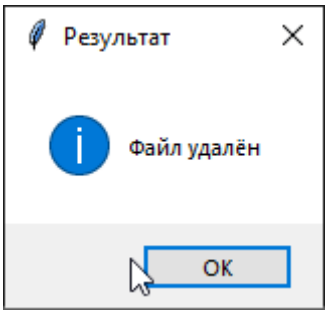
1 Вариант – Файл безопасен



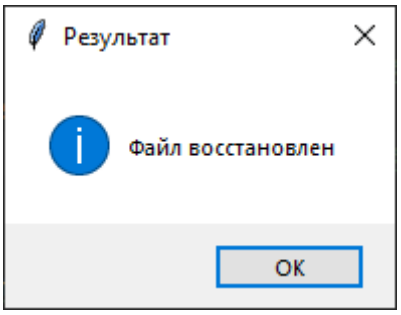
2 Вариант – Файл заражён



Да:



Нет:




```
def scan(file):
    a = [0, 0, 0]
    virus_found = False

    with open(file, "rb") as f:
        byte = f.read()
        readable_hash = hashlib.sha1(byte).hexdigest()

        print("Хэш SHA1: " + readable_hash)

        with open('SHA1 HASHES.json', 'r') as f:
            dataset = json.loads(f.read())

            for index, item in enumerate(dataset["data"]):
                if str(item['hash']) == str(readable_hash):
                    virus_found = True

            f.close()

    if not virus_found:
        a[0] = 1
    else:
        print('Файл заражён - SHA1')

    virus_found = False

    with open(file, "rb") as f:
        byte = f.read()
        readable_hash = hashlib.md5(byte).hexdigest()

        print("Хэш MD5: " + readable_hash)

        with open("MD5 Virus Hashes.txt", 'r') as f:
            lines = [line.rstrip() for line in f]
            for line in lines:
                if str(readable_hash) == str(line.split(";")[0]):
                    virus_found = True

            f.close()

    if not virus_found:
        a[1] = 1
    else:
        print('Файл заражён - MD5')

    virus_found = False

    with open(file, "rb") as f:
```

```
        byte = f.read()
        readable_hash = hashlib.sha256(byte).hexdigest()

        print("Хэш SHA256: " + readable_hash)

        with open("SHA256.txt", 'r') as f:
            lines = [line.rstrip() for line in f]
            for line in lines:
                if str(readable_hash) == str(line.split(";")[0]):
                    virus_found = True

            f.close()

    if not virus_found:
        a[2] = 1
    else:
        print('Файл заражён - SHA256')

    if a == [1, 1, 1]:
        messagebox.showinfo("Результат", "Файл безопасен")
    else:
        if messagebox.askyesno("Результат", "Файл заражён. Удалить?"):
            os.remove(file)
            messagebox.showinfo("Результат", "Файл удалён")
        else:
            messagebox.showinfo("Результат", "Файл восстановлен")
```

код функции scan

метод window.run () запускает программу

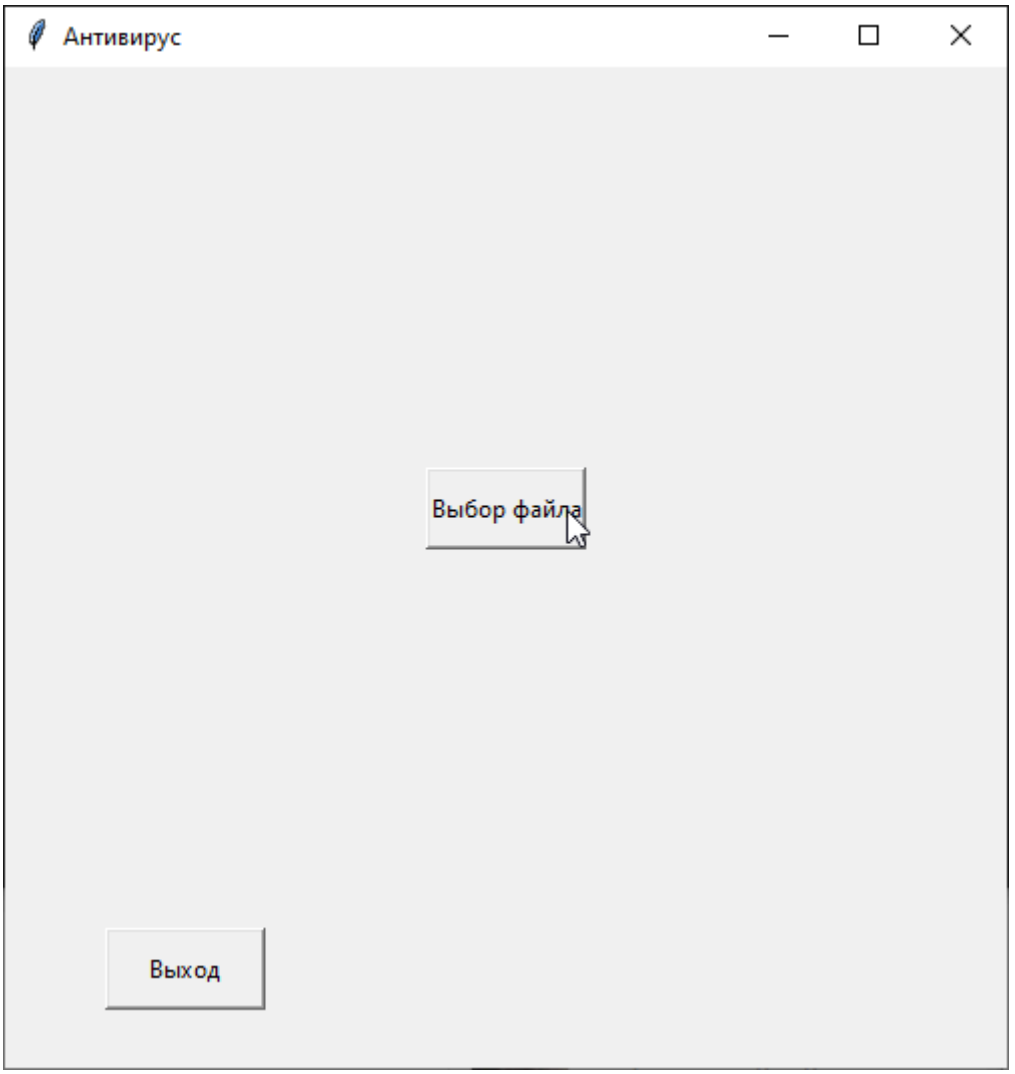
```
window.run()
```

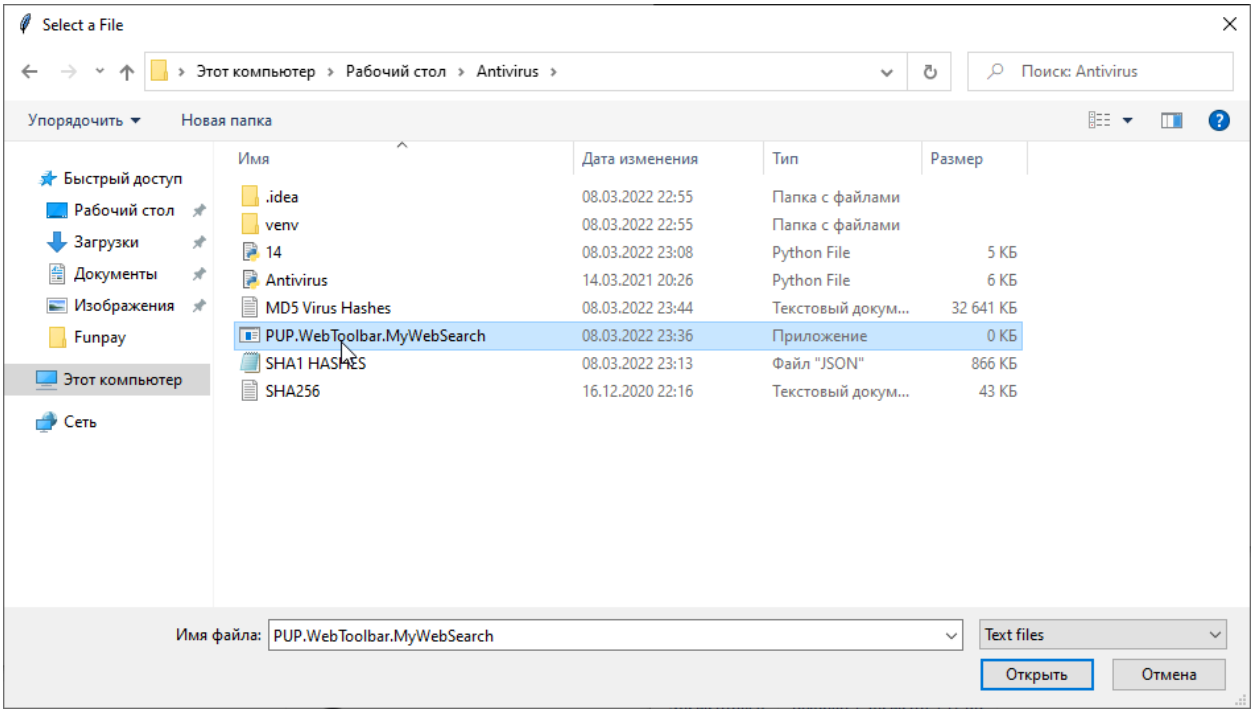
1.2 Проверка работы антивируса

Проверим антивирус на скачанном вирусе - PUP.WebToolbar.MyWebSearch.

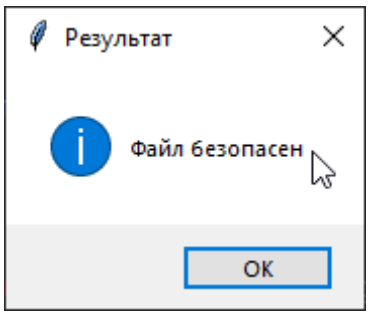
Его hash находится в одном из списков с хэшами вирусов.

Запускаем программу и выбираем файл

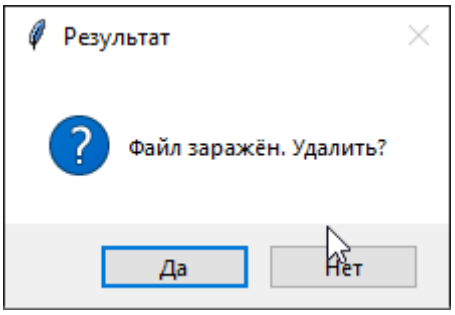

Антивирус
Выбор файла
Выход


Select a File
Этот компьютер › Рабочий стол › Antivirus ›
Поиск: Antivirus
Упорядочить
Новая папка
Быстрый доступ
Рабочий стол
Загрузки
Документы
Изображения
Funpay
Этот компьютер
Сеть
Имя
Дата изменения
Тип
Размер
.idea 08.03.2022 22:55 Папка с файлами
venv 08.03.2022 22:55 Папка с файлами
14 08.03.2022 23:08 Python File 5 КБ
Antivirus 14.03.2021 20:26 Python File 6 КБ
MD5 Virus Hashes 08.03.2022 23:44 Текстовый докум... 32 641 КБ
PUP.WebToolbar.MyWebSearch 08.03.2022 23:36 Приложение 0 КБ
SHA1 HASHES 08.03.2022 23:13 Файл "JSON" 866 КБ
SHA256 16.12.2020 22:16 Текстовый докум... 43 КБ
Имя файла: PUP.WebToolbar.MyWebSearch
Text files
Открыть
Отмена

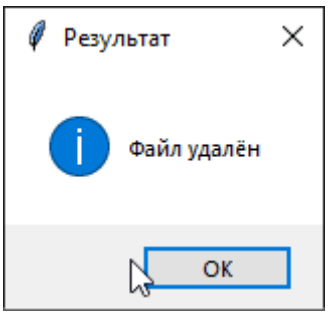
1. Файл безопасен:



2. Файл заражён



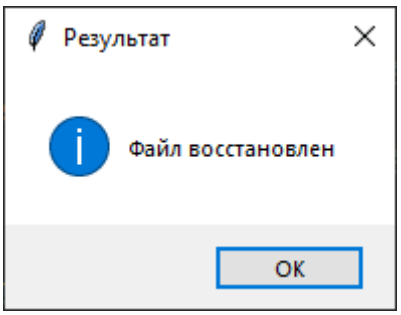
Да:



Имя	Дата изменения	Тип	Размер
.idea	08.03.2022 22:55	Папка с файлами	
venv	08.03.2022 22:55	Папка с файлами	
14	08.03.2022 23:08	Python File	5 КБ
Antivirus	14.03.2021 20:26	Python File	6 КБ
MD5 Virus Hashes	08.03.2022 23:44	Текстовый докум...	32 641 КБ
SHA1 HASHES	08.03.2022 23:13	Файл "JSON"	866 КБ
SHA256	16.12.2020 22:16	Текстовый докум...	43 КБ

Проверка и удаление файла прошла успешно, программа работает.

Нет:



Имя	Дата изменения	Тип	Размер
.idea	09.03.2022 21:54	Папка с файлами	
venv	08.03.2022 22:55	Папка с файлами	
14	09.03.2022 22:23	Python File	4 КБ
Antivirus	14.03.2021 20:26	Python File	6 КБ
MD5 Virus Hashes	08.03.2022 23:44	Текстовый докум...	32 641 КБ
PUP.WebToolbar.MyWebSearch	08.03.2022 23:46	Приложение	0 КБ
SHA1 HASHES	08.03.2022 23:13	Файл "JSON"	866 КБ
SHA256	16.12.2020 22:16	Текстовый докум...	43 КБ

Файл не удалился, программа продолжила работу, проверка прошла успешно.

ЗАКЛЮЧЕНИЕ

В результате проектно-исследовательской работы был создан антивирус на базе языка программирования Python, выполняющий проверку файла на вирусы и его удаление, если он заражён.

Целью данной работы является создание антивируса, способного обезопасить компьютер от вирусов. Данная цель была достигнута, так как антивирус способен сканировать файлы на вирусы.

Выполненные задачи:

1. Ознакомился с программой ’Антивирус’
2. Ознакомился с языком Python
3. Написал антивирус на языке программирования Python
4. Проверил работу программы и включил его в работу ПК.

Таким образом, в рамках работы цель и задачи проектно-исследовательской работы были выполнены.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. https://clck.ru/TiKjV
2. https://studbooks.net/1174018/informatika/antivirusnye_programmy
3. https://habr.com/ru/company/ruvds/blog/418823/
4. https://ru.wikipedia.org/wiki/Высокоуровневый_язык_программирования
5. https://ru.wikipedia.org/wiki/История_языка_программирования
6. https://skillbox.ru/media/code/kratkaya-istoriya-python/
7. https://docs.python.org/3/library/tkinter.html
8. https://docs.python.org/3/library/os.html
9. https://www.programiz.com/python-programming/methods/built-in/hash
10. https://docs.python.org/3/library/json.html
11.

ПРИЛОЖЕНИЯ

Код.

```
from tkinter import *
from tkinter import messagebox
from tkinter import filedialog
import os
import json
import hashlib


def scan(file):
    a = [0, 0, 0]
    virus_found = False

    with open(file, "rb") as f:
        byte = f.read()
        readable_hash = hashlib.sha1(byte).hexdigest()

        print("Хэш SHA1: " + readable_hash)

        with open('SHA1 HASHES.json', 'r') as f:
            dataset = json.loads(f.read())

            for index, item in enumerate(dataset["data"]):
                if str(item['hash']) == str(readable_hash):
                    virus_found = True

            f.close()

    if not virus_found:
        a[0] = 1
    else:
        print('Файл заражён - SHA1')

    virus_found = False

    with open(file, "rb") as f:
        byte = f.read()
        readable_hash = hashlib.md5(byte).hexdigest()

        print("Хэш MD5: " + readable_hash)

        with open("MD5 Virus Hashes.txt", 'r') as f:
            lines = [line.rstrip() for line in f]
            for line in lines:
                if str(readable_hash) == str(line.split(";")[0]):
                    virus_found = True

            f.close()

    if not virus_found:
        a[1] = 1
    else:
        print('Файл заражён - MD5')

    virus_found = False
```

```
    with open(file, "rb") as f:
        byte = f.read()
        readable_hash = hashlib.sha256(byte).hexdigest()

        print("Хэш SHA256: " + readable_hash)

        with open("SHA256.txt", 'r') as f:
            lines = [line.rstrip() for line in f]
            for line in lines:
                if str(readable_hash) == str(line.split(";")[0]):
                    virus_found = True

            f.close()

    if not virus_found:
        a[2] = 1
    else:
        print('Файл заражён - SHA256')

    if a == [1, 1, 1]:
        messagebox.showinfo("Результат", "Файл безопасен")
    else:
        if messagebox.askyesno("Результат", "Файл заражён. Удалить?"):
            os.remove(file)
            messagebox.showinfo("Результат", "Файл удалён")
        else:
            messagebox.showinfo("Результат", "Файл восстановлен")


def browsefiles():
    filename = filedialog.askopenfilename(initialdir="/",
                                          title="Выбор файла",
                                          filetypes=(("Text files",
                                                      "*.*"),
                                                     ("all files",
                                                      "*.*")))

    scan(filename)


class Main:
    def __init__(self, width, height, title="Antivirus"):
        self.root = Tk()
        self.root.title(title)
        self.root.geometry(f'{width}x{height}+200+200')

    def close(self):
        if messagebox.askyesno('Выход', 'Вы уверены?'):
            self.root.destroy()

    def run(self):
        self.create_menu()
        self.root.mainloop()

    def create_menu(self):
        Button(self.root, height=2, width=10, text="Выход",
command=self.close).place(x=50, y=430)

        Button(self.root, text="Выбор файла", height=2, width=10,
```

```
command=browsefiles).pack(anchor='center',

pady=200)


if __name__ == '__main__':
    window = Main(500, 500, "Антивирус")
    window.run()
```