Федеральное государственное образовательное учреждение

высшего профессионального образования

«**КАЛИНИНГРАДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**»

Кафедра систем управления и вычислительной техники

|  |  |
| --- | --- |
| Курсовая работа  допущена к защите  руководитель | Курсовая работа защищена  с оценкой …………………  руководитель |

**«Антивирус»**

Курсовая работа по дисциплине

**“Высокоуровневые технологии программирования”**

Пояснительная записка

53.09.03.03.03

Нормконтроллер:

доцент, к.т.н.,

Высоцкий Л.Г.

Проект выполнил:

студент группы 19-ИЭ-1

Богданов М. Д.

Калининград

2022

**Содержание**

[Аннотация 3](#_Toc95331028)

[Обоснование выбора языка программирования 3](#_Toc95331029)

[Описание модулей программы и их взаимосвязь 7](#_Toc95331030)

[Вид программного интерфейса 8](#_Toc95331031)

[Блок-схемы модулей 9](#_Toc95331032)

[Листинг программы 10](#_Toc95331033)

[Вид результатов выполнения программы 13](#_Toc95331034)

[Руководство пользователя 15](#_Toc95331035)

[Заключение 16](#_Toc95331036)

[Список использованной литературы 18](#_Toc95331037)

# Аннотация

Представленная курсовая работа состоит из введения, подготовки к разработке программы, самой разработки, справочных данных, заключения и списка литературных источников.

В первой части работы обосновывается выбор языка программирования. В следующей части детально производится подготовка к разработке программы, затем сам листинг, скриншоты работы и руководство пользователя.

Работа состоит из 18 страниц, содержит 8 литературных источников, 1 таблицу и 5 рисунков.

# Обоснование выбора языка программирования

Был выбран язык разработки Python, по следующим причинам:

Высокая скорость разработки.

По сравнению с компилирующими или строго типизированными языками, такими как С, C++ и Java, Python во много раз повышает производительность труда разработчика. Объем программного кода на языке Python обычно составляет треть или даже пятую часть эквивалентного программногокода на языке C++ или Java. Это означает меньший объем ввода с клавиатуры, меньшее количество времени на отладку и меньший объем трудозатрат на сопровождение. Кроме того, программы на языке Python запускаются сразу же, минуя длительные этапы компиляции и связывания, необходимые в некоторых других языках программирования, что еще больше увеличивает производительность труда программиста.

Переносимость программ.

Большая часть программ на языке Python выполняется без изменений на всех основных платформах. Перенос программного кода из операционной системы Linux в Windows обычно заключается в простом копировании файлов программ с одной машины на другую. Более того, Python предоставляет массу возможностей по созданию переносимых графических интерфейсов, программ доступа к базам данных, веб-приложений и многих других типов программ. Даже интерфейсы операционных систем, включая способ запуска программ и обработку каталогов, в языке Python реализованы переносимым способом.

Библиотеки поддержки. В составе Python поставляется большое число собранных и переносимых функциональных возможностей, известных как стандартная библиотека. Эта библиотека предоставляет массу возможностей, востребованных в прикладных программах, начиная от поиска текста по шаблону и заканчивая сетевыми функциями. Кроме того, Python допускает расширение как за счет ваших собственных библиотек, так и за счет библиотек, созданных сторонними разработчиками. Из числа сторонних разработок можно назвать инструменты создания веб-сайтов, программирование математических вычислений, доступ к последовательному порту, разработку игровых программ и многое другое. Например, расширение NumPy позиционируется как свободный и более мощный эквивалент системы программирования математических вычислений Mathlab.

Интеграция компонентов.

Сценарии Python легко могут взаимодействовать с другими частями приложения благодаря различным механизмам интеграции. Эта интеграция позволяет использовать Python для настройки и расширения функциональных возможностей программных продуктов. На сегодняшний день программный код на языке Python имеет возможность вызывать функции из библиотек на языке C/C++, сам вызываться из программ, написанных на языке C/C++, интегрироваться с программными компонентами на языке Java, взаимодействовать с такими платформами, как СОМ и .NET, и производить обмен данными через последовательный порт или по сети с помощью таких протоколов, как SOAP, XML-RPC и CORBA. Python - не обособленный инструмент.

Качество программного обеспечения.

По своей природе Python имеет простой, удобочитаемый синтаксис и ясную модель программирования. Согласно лозунгу, выдвинутому на недавней конференции по языку Python, основное его преимущество состоит в том, что Python «каждому по плечу» - характеристики языка взаимодействуют ограниченным числом непротиворечивых способов и естественно вытекают из небольшого круга базовых концепций. Это делает язык простым в освоении, понимании и запоминании. На практике программистам, использующим язык Python, почти не приходится прибегать к справочным руководствам - это непротиворечивая система, на выходе которой, к удивлению многих, получается профессиональный программный код. Философия Python по сути диктует использование минималистского подхода. Это означает, что даже при наличии нескольких вариантов решения задачи в этом языке обычно существует всего один очевидный путь, небольшое число менее очевидных альтернатив и несколько взаимосвязанных вариантов организации взаимодействий. Более того, Python не принимает решения за вас, когда порядок взаимодействий неочевиден - предпочтение отдается явному описанию, а не «волшебству». В терминах Python явное лучше неявного, а простое лучше сложного.1 Помимо философии Python обладает такими возможностями, как модульное и объектно-ориентированное программирование, что естественно упрощает возможность многократного использования программного кода. Поскольку качество находится в центре внимания самого языка Python, оно также находится в центре внимания программистов.

Высокая скорость разработки

Во время бума развития Интернета во второй половине 1990-х годов было сложно найти достаточное число программистов для реализации программных проектов - от разработчиков требовалось писать программы со скоростью развития Интернета. Теперь, в эпоху экономического спада, картина изменилась. Сегодня от программистов требуется умение решать те же задачи меньшим числом сотрудников. В обоих этих случаях Python блистал как инструмент, позволяющий программистам получать большую отдачу при меньших усилиях. Он изначально оптимизирован для достижения высокой скорости разработки - простой синтаксис, динамическая типизация, отсутствие этапа компиляции и встроенные инструментальные средства позволяют программистам создавать программы за меньшее время, чем при использовании некоторых других инструментов. В результате Python увеличивает производительность труда разработчика во много раз по сравнению с традиционными языками программирования. Это значительное преимущество, которое с успехом может использоваться как во время бума, так и во время спада, а также во время любого промежуточного этапа развития индустрии программного обеспечения.

**Постановка задачи**

Данная программа запоминает содержимое указанной папки и длину файлов в ней. При запуске она определяет появление новых файлов в данной папке, удаление существовавших и изменение размера у исполняемых файлов. Выводится информация об изменениях. Количество анализируемых папок не ограничено.

# Описание модулей программы и их взаимосвязь

Описание модулей программы представлено в таблице 1.

Табл. 1 – Список модулей и их описание

|  |  |  |
| --- | --- | --- |
| Модули | Чем представлены модули | Описание модулей |
| Unit 1 | Методы конструктора tkinter | Создание окна программы и расстановка виджетов. |
| Unit 2 | Функция Folderpath() | Выбор проверяемой папки. |
| Unit 3 | Функция Start() | При нажатии на кнопку, вызывающую данную функцию производится проверка существования указанной в поле entry папки. При успехе вызывается функция ExistingFile(), иначе функция NewFile(). |
| Unit 4 | Функция NewFile() | Добавление содержимого папки и размера файлов внутри неё в файл logs.txt. |
| Unit 5 | Функция ExistingFile() | Производится поиск папки в файле logs.txt и запускается функция DifferencesCheck() |
| Unit 6 | Функция DifferencesCheck() | Вывод изменений размера содержимого в папке, а также добавления и/или удаления файлов в ней. |

# Вид программного интерфейса

На рисунке 1 показан вид программного интерфейса программы

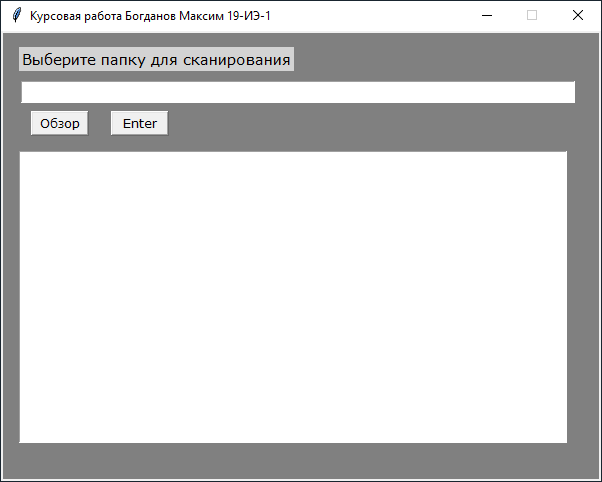


Рис. 1 - вид программного интерфейса

Для расположения необходимых виджетов и дальнейшего функционировании программы необходимо:

* Окно Tkinter
* Расположенные на нём виджеты

# Блок-схемы модулей

На рисунке 2 показана блок-схема модулей программы



Рис. 2 – Блок-схема модулей программы

# Листинг программы

from tkinter import \* # модуль графического интерфейса

import tkinter.filedialog # модуль файлового диалога

import os # модуль для взаимодействия с файловой системой

import ast # модуль, используемый для преобразования строки в список

def Folderpath(): # функция выбора директории

entry.delete(0, END)

entry.insert(0, tkinter.filedialog.askdirectory())

def DifferencesCheck(array): # функция вывода изменений в директории

#ray - новые данные ### #array - данные в базе

ray=os.listdir(path=entry.get())

for i in range(len(ray)):

ray[i]=[ray[i]] # преобразование 1мерного в 2мерный массив

ray[i].append(os.path.getsize(entry.get()+"/"+ray[i][0])) # размер файла в байтах

array=ast.literal\_eval(array) # преобразование строки из базы в список

text['state'] = "normal"

text.insert(END, "Выполняется проверка папки " + entry.get() + "\n")

text.insert(END, "Изменение размеров содержимого: ")

b=0

for i in ray:#поиск изменённых файлов

for j in array:

if i[0]==j[0] and i[1]!=j[1]: #сравнение размеров файлов в байтах

b=1

text.insert(END,"\n " + i[0] + ": " + str(j[1]) + " b → " + str(i[1]) + " b", "tag\_blue\_text")

if b==0:

text.insert(END,"не выявлено\n", "tag\_green\_text")

names1=[] # новый список с именами файлов из базы

names2=[] # новый список с именами проверяемых файлов

for i in array:

names1.append(i[0])

for i in ray:

names2.append(i[0])

text.insert(END, "\nИзменение содержимного: ")

plus\_names = list(set(names2) - set(names1)) # новые файлы

minus\_names = list(set(names1) - set(names2)) # удалённые файлы

if plus\_names == [] and minus\_names==[]:

text.insert(END,"не выявлено\n", "tag\_green\_text")

else:

for i in plus\_names:

text.insert(END, "\n + " + i, "tag\_blue\_text")

for i in minus\_names:

text.insert(END, "\n - " + i, "tag\_blue\_text")

text.insert(END, "\n")

text['state'] = "disabled"

def ExistingFile(): # функция поиска папки в базе

b=0

f = open('logs.txt', 'r')

for line in f:

if b:

DifferencesCheck(line)

break

if line==entry.get()+"\n": #имя папки есть в базе

b=1

f.close()

def NewFile(): # функция добавления папки в базу

f = open('logs.txt', 'a') # запись текущего содержимого папки

arr=os.listdir(path=entry.get()) # список файлов в папке

for i in range(len(arr)):

arr[i]=[arr[i]] # преобразование 1мерного в 2мерный список

arr[i].append(os.path.getsize(entry.get()+"/"+arr[i][0])) # размер файла в байтах

f.write("\n\n"+ entry.get() + "\n" + str(arr)) # запись информации

f.close()

text['state'] = "normal"

text.insert(END, 'Папка добавлена в базу\n')

text['state'] = "disabled"

def Start(): # функция проверки папки

f = open('logs.txt', 'a')

if os.path.exists(entry.get()):

f = open('logs.txt', 'r')

b=0

for line in f: # проверка папки на содержание в базе

if line == entry.get()+"\n":

b = 1

break

if b:

ExistingFile()

else:

NewFile()

else:

text['state'] = "normal"

text.insert(END, 'Путь к папке указан некорректно\n', 'tag\_red\_text')

text['state'] = "disabled"

f.close()

root = Tk()

root.resizable(0, 0)

root.geometry('600x450')

root.title('Курсовая работа Богданов Максим 19-ИЭ-1')

c = Canvas(width = 596, height = 446, bg = 'grey') # канва

c.place(x = 0, y = 0)

entry = Entry(width = 55, font = "Verdana 11") # поле ввода

entry.place(x = 20, y = 50)

text = Text(width = 68, height = 18, font = "Verdana 10", state='disabled') # поле вывода информации

text.place(x = 18, y = 120)

text.tag\_config('tag\_red\_text', foreground='red')

text.tag\_config('tag\_green\_text', foreground='green')

text.tag\_config('tag\_blue\_text', foreground='blue')

search = Button(width = 6, text = "Обзор", font = "Verdana 9", command = Folderpath) # выбор папки

search.place(x = 30, y = 80)

enter = Button(width = 6, text = "Enter", font = "Verdana 9", command = Start) # начать сканирование

enter.place(x = 110, y =80)

label = Label(text = "Выберите папку для сканирования", bg = "lightgrey", font = "Verdana 11") # метка

label.place(x = 18, y = 16)

root.mainloop()

# Вид результатов выполнения программы

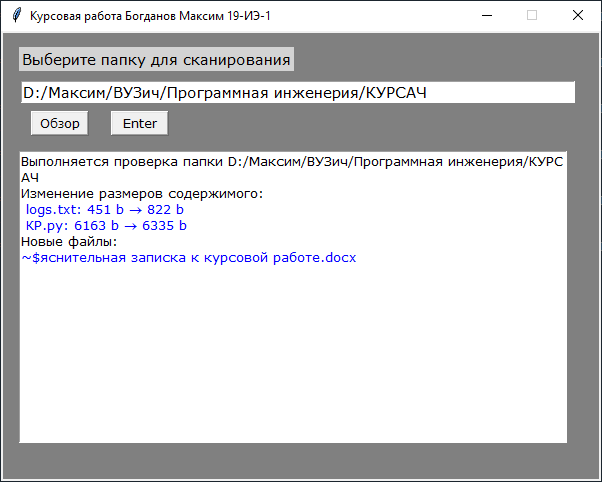


Рис 3.1 – скриншот работы программы

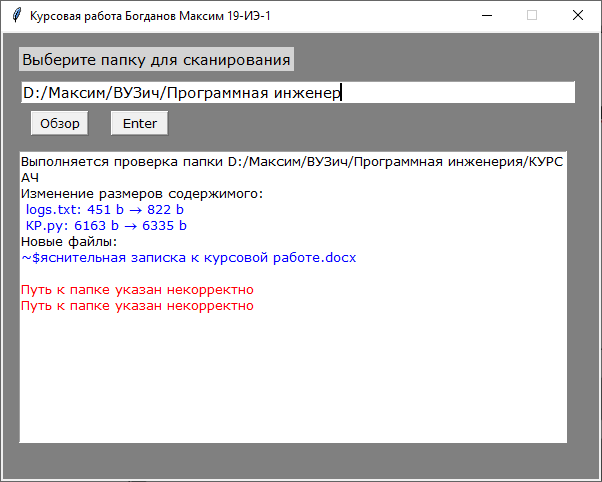


Рис 3.2 – скриншот работы программы

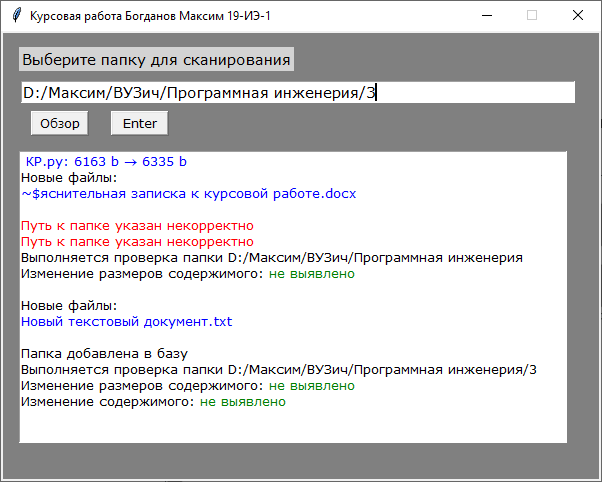


Рис 3.3 – скриншот работы программы

# Руководство пользователя

1. *Общие сведения о программе*.

Наименование: “Антивирус”.

Программа запоминает содержимое указанной папки и длину файлов в ней. При запуске она определяет появление новых файлов в данной папке, удаление существовавших и изменение размера у исполняемых файлов. Выводится информация об изменениях.

Количество анализируемых папок не ограничено.

В коде реализованы методы создания графического интерфейса, работы с файлами, преобразования строк, использования файлового диалога.

Программа будет полезна для повышения безопасности и контроля за файлами в системе компьютера, а также изучения влияния различных программ на файлы на диске.

1. *Установка*.

Для установки данного ПО имеются следующие требования к программному обеспечению:

* Windows - 64-битная x86, 32-битная x86; MacOS - 64-битная x86; Linux - 64-битная x86, 64-битная Power8 / Power9
* Установленная среда разработки Python версии 3.0 и выше

1. *Запуск.*

Для запуска данного ПО существуют следующие требования к техническому обеспечению:

* RAM: 1+ ГБ
* CPU: 2+ ядер

1. *Инструкция по работе.*

Программа имеет 2 режима работы:

* Добавление папки в базу для дальнейшего анализа.
* Анализ папки на наличие изменений в размере исполняемых файлов, наличия новых файлов или отсутствия существовавших.

1. *Сообщения пользователю.*

Если в поле вывода высвечено сообщение “Путь к папке указан некорректно”, то пользователю необходимо проверить указанный в поле ввода путь, выбрать существующий и продолжить работу.

# Заключение

В результате проделанной работы была достигнута поставленная цель, а именно, разработана программа “Антивирус”.

Теоретическая часть помогла избежать лишних ошибок и сыграла немалую роль в понимании основных процессов и структур

В ходе работы мною были укреплены знания и умения в написании программ на языке Python с использованием библиотеки создания графических интерфейсов Tkinter. Были задействованы такие умения как:

1. Создание и обработка входных данных через интерфейс Tkinter
2. Работа с файловой системой

Эти умения будут полезны при последующем написании последующих курсовых, практических лабораторных, а также дипломной работы.

Главное достоинство Python – простота, в которой я убедился, достигая цели – создавая программу, и о которой говорится в послании разработчиков – философии Python. В большей степени к упрощению написания программ приводит отсутствие необходимости объявлять переменные, тип которых определяется по ходу написания кода автоматически, а также удобная система отступов, что придает коду компактность и структурированность.

Безусловно, как и любой язык программирования, Python имеет некоторые свои недостатки. Но преимущества, которые он предоставляет при создании программного обеспечения, гораздо существеннее и не оказывают особого влияния на его функциональность и практичность. В случае же критичности некоторых недостатков при решении некоторых специфических задач, универсальность и гибкость Python позволяет обходить эти недостатки без ущерба для решаемой задачи. Таким образом, можно с большой уверенностью утверждать, что Рython подходит для решения подавляющего большинства повседневных задач, будь то подключение к сети интернет, чтение-отправка электронной почты, резервное копирование, либо же какая-нибудь игрушка. Язык программирования Python практически не имеет никаких ограничений или запретов на использование, поэтому также может свободно использоваться при создании крупных проектов. К примеру, Рython интенсивно применяется многими IT-гигантами, такими как, например, Google и Yandex. А универсальность и простота и Рython делают его одним из лидеров среди языков программирования как для профессионалов, так и для тех, кто только начинает пробовать свои силы в программировании.

Подводя итоги, можно сделать вывод, что Python вполне может подойти для изучения как начинающим программистам, так разработчикам с большим опытом.

# Список использованной литературы

1. <https://habr.com/ru/post/133337/>
2. <http://python-3.ru/page/pochemu-programmisty-ispolzujut-python>
3. <https://ru.stackoverflow.com/questions/519365/Как-конвертировать-string-в-list>
4. <https://pythonworld.ru/moduli/modul-os.html>
5. <https://pythonworld.ru/tipy-dannyx-v-python/fajly-rabota-s-fajlami.html>
6. <https://tech-wiki.online/ru/python-get-file-details.html>
7. <https://younglinux.info/tkinter/widget>
8. <https://pythonbasics.org/tkinter-filedialog/>