



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНЖЕНЕРНЫХ ТЕХНОЛОГИЙ»

Факультет управления и информатики в технологических системах
Кафедра информационной безопасности
Направление подготовки (специальность) 10.05.03 Информационная безопасность
автоматизированных систем

Отчет
По технологиям и методам программирования.
Практическая работа №10
наименование (вид) дисциплины

Выполнил студент гр. УБ-21
Заварзин В.Д.
(ф.и.о.)

Проверил:

Доцент каф. ИБ Маслов А.А._____

(оценка)

(подпись)

(дата)

Формулировка:

Создать приложение с графическим интерфейсом со следующими параметрами:

1. Название приложения: ФИО автора.
2. Создать в приложении 3 вкладки с отступом друг от друга (равномерно распределить по окну)
 - Первая вкладка: простой калькулятор для двух чисел (между ними выпадающий список $+$ $-$ $*$ $/$).
 - Вторая вкладка: сделать три чекбокса
Первый
Второй
Третий
И кнопку, при нажатии на нее в зависимости от выбора, выводить всплывающее окно с надписью (например: *вы выбрали первый вариант*).
 - Третья вкладка: работа с текстом, можно текст загрузить файла по кнопке из созданного вами меню.

Решение

```
from tkinter import *
from tkinter import messagebox, filedialog, scrolledtext
from tkinter.ttk import *

def clicked():
    c1 = ch1.get()
    c2 = ch2.get()
    sign = combo.get()
    if sign != '':
        if c1.isdigit() and c2.isdigit():
            try:
                string = str(c1 + sign + c2)
                out = str(string + '=' + str(eval(string)))
                lbl.configure(text=out)
            except ZeroDivisionError:
                messagebox.showerror('Математическая ошибка!', 'Деление на ноль невозможно!')
        else:
            messagebox.showerror('Внимание!', 'Ячейки не должны быть пустыми и не должны содержать посторонние символы')
    else:
        lbl.configure(text='Выберите действие')

win = Tk()

win.title("Заварзин Владислав Дмитриевич")
win.geometry('475x450')

tab_control = Notebook(win)
tab1 = Frame(tab_control)
tab_control.add(tab1, text='Калькулятор')
tab2 = Frame(tab_control)
tab_control.add(tab2, text='Чекбокс')
tab3 = Frame(tab_control)
tab_control.add(tab3, text='Текст')
```

```
'''***** Работа с калькулятором *****'''

ch1 = Entry(tab1)
ch1.grid(column=0, row=0)

combo = Combobox(tab1)
combo.grid(column=1, row=0)
combo['values'] = ('+', '-', '*', '/')

ch2 = Entry(tab1)
ch2.grid(column=2, row=0)

lbl = Label(tab1, text='0')
lbl.grid(column=1, row=1)

btn = Button(tab1, text='Рассчитать', command=clicked)
btn.grid(column=3, row=0)
```

```

'''***** Работа с Checkbutton *****'''
def check():
    var = ''
    if var1.get():
        if var:
            var += ', '
        var += var1.get()
    if var2.get():
        if var:
            var += ', '
        var += var2.get()
    if var3.get():
        if var:
            var += ', '
        var += var3.get()

    if var:
        messagebox.showinfo('Сообщение', 'Вы выбрали ' + var + ' вариант')
    else:
        messagebox.showinfo('Сообщение', 'Вы ничего не выбрали')

var1 = StringVar()
var1.set('')
var2 = StringVar()
var2.set('')
var3 = StringVar()
var3.set('')
check1 = Checkbutton(tab2, onvalue='первый', offvalue='', variable=var1, text='Первый')
check1.grid(column=0, row=0)
check2 = Checkbutton(tab2, onvalue='второй', offvalue='', variable=var2, text='Второй')
check2.grid(column=1, row=0)
check3 = Checkbutton(tab2, onvalue='третий', offvalue='', variable=var3, text='Третий')
check3.grid(column=2, row=0)

btn_check = Button(tab2, text='Check', command=check)
btn_check.grid(column=3, row=0)

```

```

'''***** Работа с text *****'''
def file_import():
    file = filedialog.askopenfilename(filetypes=(("Text files", " *.txt"), ("all files", " *.* ")))
    text_box.delete('1.0', END)
    with open(file) as open_file:
        lines = open_file.readlines()
        for line in lines:
            text_box.insert(END, f'{line.strip()}\n')

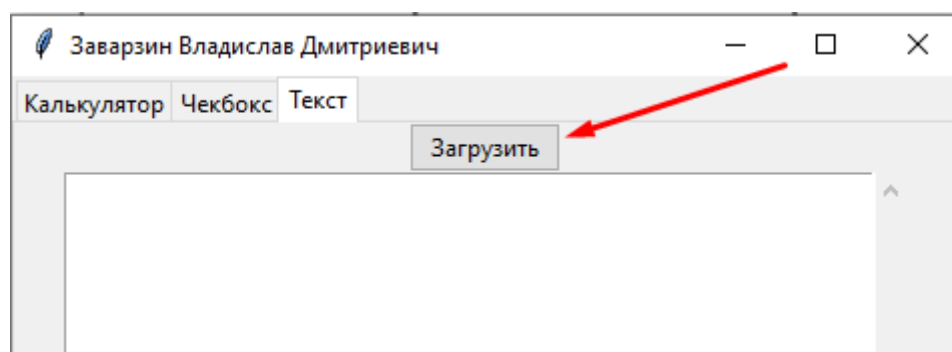
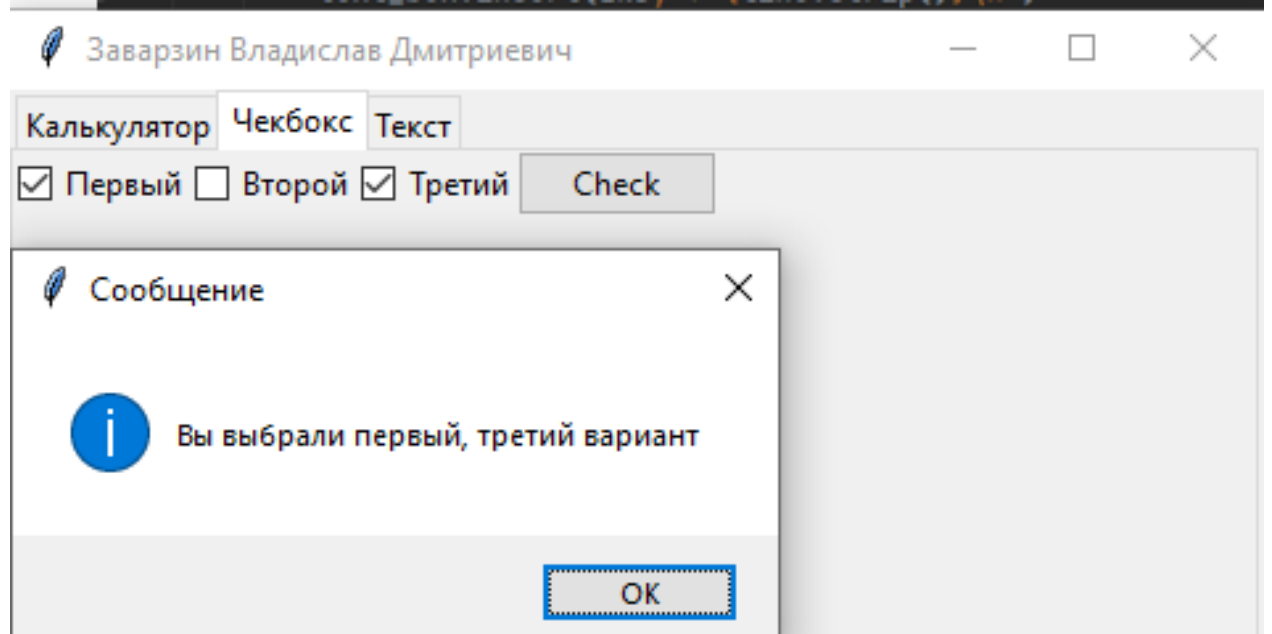
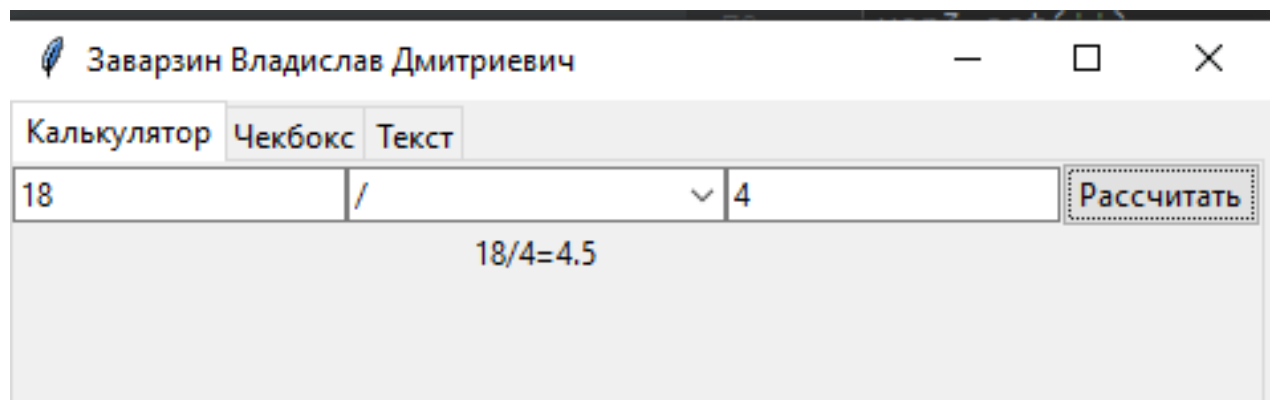
btn_import = Button(tab3, text='Загрузить', command=file_import)
btn_import.pack()
text_box = scrolledtext.ScrolledText(tab3, width=50, height=20)
text_box.pack()

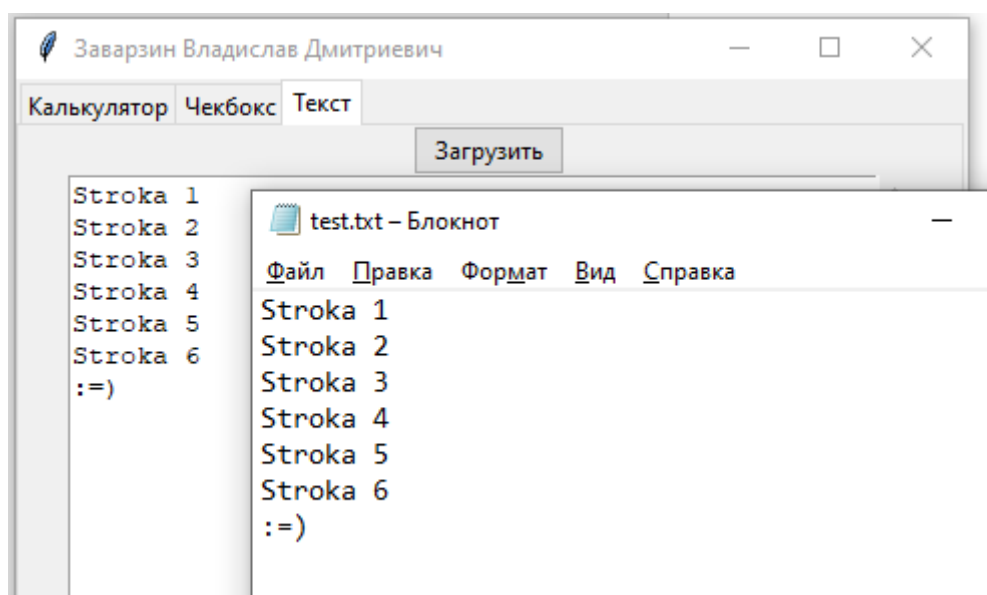
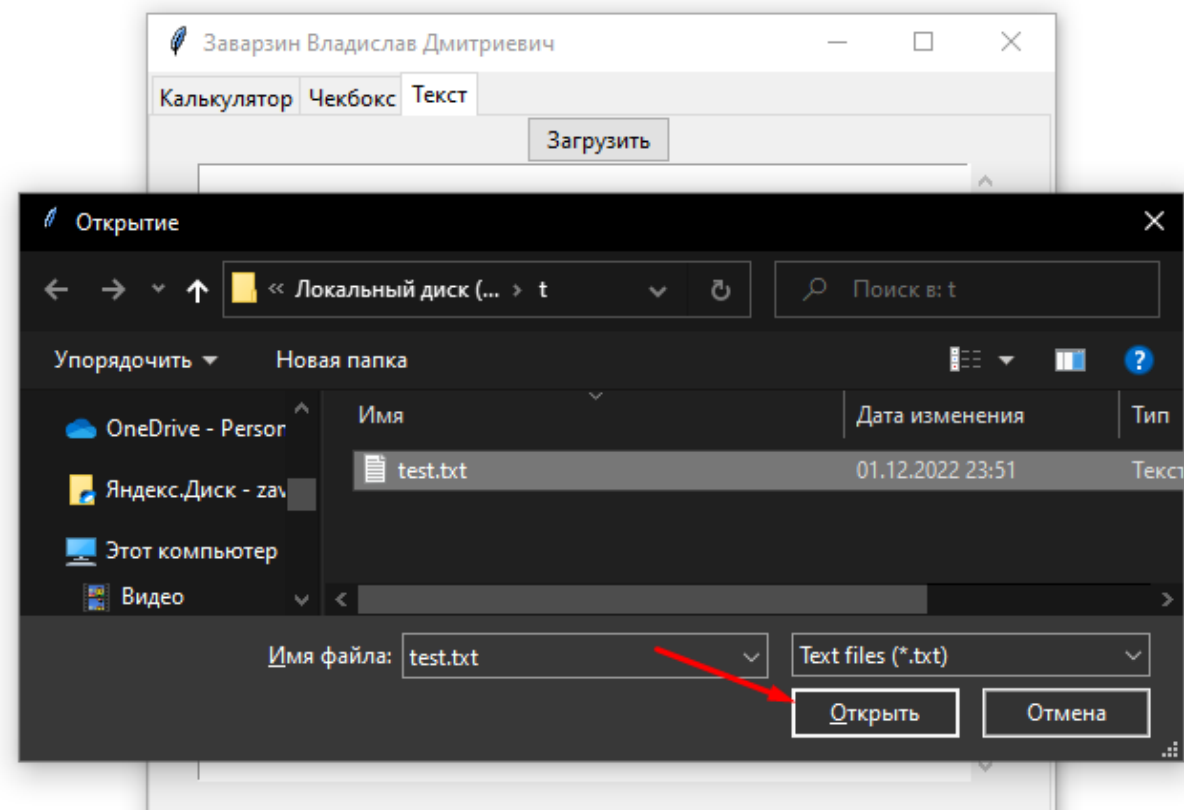
tab_control.pack(expand=1, fill='both')

win.mainloop()

```

(рис.1-4)





Пример работы программы (рис.5-9)

Код программы:

```
from tkinter import *
from tkinter import messagebox, filedialog, scrolledtext
from tkinter.ttk import *

def clicked():
    c1 = ch1.get()
    c2 = ch2.get()
    sign = combo.get()
    if sign != '':
        if c1.isdigit() and c2.isdigit():
            try:
                string = str(c1 + sign + c2)
                out = str(string + '=' + str(eval(string)))
                lbl.configure(text=out)
            except ZeroDivisionError:
                messagebox.showerror('Математическая ошибка!', 'Деление на ноль невозможно!')
            else:
                messagebox.showerror('Внимание!', 'Ячейки не должны быть пустыми и не должны содержать посторонние символы')
        else:
            lbl.configure(text='Выберите действие')

win = Tk()

win.title("Заварзин Владислав Дмитриевич")
win.geometry('475x450')

tab_control = Notebook(win)
tab1 = Frame(tab_control)
tab_control.add(tab1, text='Калькулятор')
tab2 = Frame(tab_control)
tab_control.add(tab2, text='Чекбокс')
tab3 = Frame(tab_control)
tab_control.add(tab3, text='Текст')
'''***** Работа с калькулятором *****'''
ch1 = Entry(tab1)
ch1.grid(column=0, row=0)

combo = Combobox(tab1)
combo.grid(column=1, row=0)
combo['values'] = ('+', '-', '*', '/')

ch2 = Entry(tab1)
ch2.grid(column=2, row=0)

lbl = Label(tab1, text='0')
lbl.grid(column=1, row=1)

btn = Button(tab1, text='Рассчитать', command=clicked)
btn.grid(column=3, row=0)
'''***** Работа с Checkbutton *****'''
def check():
    var = ''
    if var1.get():
        if var:
            var += ', '
        var += var1.get()
    if var2.get():
```

```

        if var:
            var += ', '
        var += var2.get()
    if var3.get():
        if var:
            var += ', '
        var += var3.get()

    if var:
        messagebox.showinfo('Сообщение', 'Вы выбрали ' + var + ' вариант')
    else:
        messagebox.showinfo('Сообщение', 'Вы ничего не выбрали')

var1 = StringVar()
var1.set('')
var2 = StringVar()
var2.set('')
var3 = StringVar()
var3.set('')
check1 = Checkbutton(tab2, onvalue='первый', offvalue='', variable=var1,
text='Первый')
check1.grid(column=0, row=0)
check2 = Checkbutton(tab2, onvalue='второй', offvalue='', variable=var2,
text='Второй')
check2.grid(column=1, row=0)
check3 = Checkbutton(tab2, onvalue='третий', offvalue='', variable=var3,
text='Третий')
check3.grid(column=2, row=0)

btn_check = Button(tab2, text='Check', command=check)
btn_check.grid(column=3, row=0)
'''*****Работа с
text*****'''
def file_import():
    file = filedialog.askopenfilename(filetypes=(("Text files", " *.txt"),
("all files", " *.* ")))
    text_box.delete('1.0', END)
    with open(file) as open_file:
        lines = open_file.readlines()
        for line in lines:
            text_box.insert(END, f'{line.strip()}\n')

btn_import = Button(tab3, text='Загрузить', command=file_import)
btn_import.pack()
text_box = scrolledtext.ScrolledText(tab3, width=50, height=20)
text_box.pack()

tab_control.pack(expand=1, fill='both')

win.mainloop()

```

Вывод:

Выполнив данную лабораторную работу я узнал о такой библиотеке как **Tkinter**, познакомился с такими элементами, как: вкладка, кнопка, диалоговое окно, CheckButton, текстовое поле и др.; научился выводить окно и отображать на нём все элементы. Главным объектом при выполнении всех заданий послужил такой элемент, как “Button” – это кнопка внутри которой есть такой параметр, как “command”, которому нужно было присваивать функции, которые выполнялись после нажатия. В свою очередь эти функции и

обрабатывали данные полученные от элементов, будь то состояние “`CheckButton`” или какое-то значение в текстовом поле. Также было использовано множество других параметров и методов. Такой метод, как “`grid`” использовался, чтобы расположить элементы элементы внутри окна в том порядке, что мне нужен был, как например в “калькуляторе”, я располагал элементы в одной строке “`row=0`” но в “колонках” друг за другом “`column=0..3`”. И наконец часто использовался такое метод, как “`text`”, которому присваивалось значение текста отвечающее за отображения текста будь то в заголовке элемента или прям на нём самом. Получив знания я применил их на практике создав три вкладки, в каждой выполнив задание.

При выполнении лабораторной работы использовалась среда программирования ***PyCharm*** (см. рис.1-9). На рис. представлен исходный код программы и пример его работы.



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНЖЕНЕРНЫХ ТЕХНОЛОГИЙ»

Факультет управления и информатики в технологических системах
Кафедра информационной безопасности
Направление подготовки (специальность) 10.05.03 Информационная безопасность
автоматизированных систем

Отчет
По технологиям и методам программирования.
Практическая работа №11

наименование (вид) дисциплины

Выполнил студент гр. УБ-21
Заварзин В.Д.
(ф.и.о.)

Проверил:

Доцент каф. ИБ Маслов А.А._____

(оценка)

(подпись)

(дата)

Формулировка:

Задание: даны самые популярные репозитории на github <https://habr.com/ru/post/453444/>, по последней цифре зачетки получить JSON для вашего варианта .

Программа с графическим интерфейсом вводим в поле имя репозитория и по нажатию кнопки получаем результат.

Необходимо получить в новый файл следующую информацию:

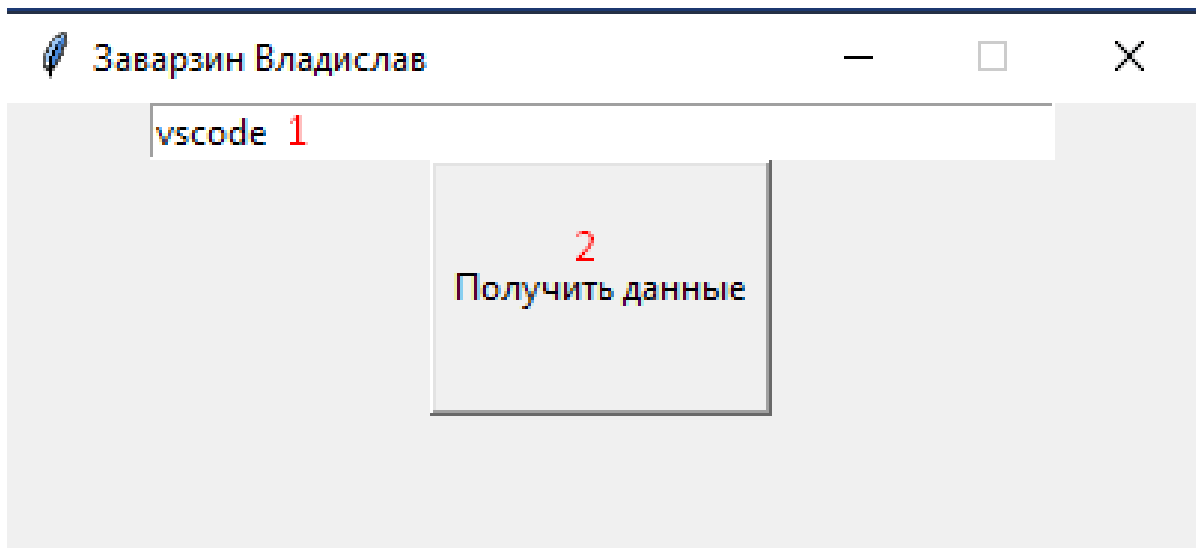
```
'company': None,  
'created_at': '2015-08-03T17:55:43Z',  
'email': None,  
'id': 13629408,  
'name': 'Kubernetes',  
'url': 'https://api.github.com/users/kubernetes'}
```

Все прикрепить одним архивом.

Решение

```
import traceback  
from tkinter import *  
import requests  
  
def data_site():  
    try:  
        url = 'https://api.github.com/users/' + input_text.get()  
        site = requests.get(url).json()  
        company = site['company']  
        created_at = site['created_at']  
        email = site['email']  
        id = site['id']  
        name = site['name']  
        with open('export', 'w') as f:  
            f.write('company: ' + str(company) + '\n')  
            f.write('created_at: ' + str(created_at) + '\n')  
            f.write('email: ' + str(email) + '\n')  
            f.write('id: ' + str(id) + '\n')  
            f.write('name: ' + str(name) + '\n')  
            f.write('url: ' + str(url) + '\n')  
        print('Done!')  
        '''Вывод ошибки в случае чего'''  
    except Exception as e:  
        print('Error')  
        print(traceback.format_exc())  
  
window = Tk()  
window.title('Заварзин Владислав')  
window.geometry('400x150+0+0')  
window.resizable(False, False)  
'''Расположение окна посреди экрана'''  
x = (window.winfo_screenwidth() - window.winfo_reqwidth()) / 2.35  
y = (window.winfo_screenheight() - window.winfo_reqheight()) / 2.35  
window.wm_geometry("%d+%d" % (x, y))  
  
input_text = Entry(window, width=50)  
input_text.pack(anchor=N)  
  
btn = Button(window, text='Получить данные', width=15, height=5, command=data_site).pack()  
  
window.mainloop()
```

(рис.1)



1 – вписываем название репозитория по варианту

2 – генерируем файл с данными, которые нам надо получить по заданию

Пример работы программы (рис.2)

A screenshot of a Python script output. The output is displayed in a dark-themed window with two tabs: "main.py" and "export". The "export" tab is active, showing the following JSON data:

```
1 company: None
2 created_at: 2014-07-11T17:01:00Z
3 email: None
4 id: 8137302
5 name: Victor Salazar
6 url: https://api.github.com/users/vscode|
```

Выходные данные (рис.3)

Код программы:

```
import traceback
from tkinter import *
import requests
def data_site():
    try:
        url = 'https://api.github.com/users/' + input_text.get()
        site = requests.get(url).json()
        company = site['company']
        created_at = site['created_at']
        email = site['email']
        id = site['id']
        name = site['name']
        with open('export', 'w') as f:
            f.write('company: ' + str(company) + '\n')
            f.write('created at: ' + str(created_at) + '\n')
```

```

        f.write('email: ' + str(email) + '\n')
        f.write('id: ' + str(id) + '\n')
        f.write('name: ' + str(name) + '\n')
        f.write('url: ' + str(url) + '\n')
    print('Done!')
    '''Вывод ошибки в случае чего'''
except Exception as e:
    print('Error')
    print(traceback.format_exc())

window = Tk()
window.title('Заварзин Владислав')
window.geometry('400x150+0+0')
window.resizable(False, False)
'''Расположение окна посреди экрана'''
x = (window.winfo_screenwidth() - window.winfo_reqwidth()) / 2.35
y = (window.winfo_screenheight() - window.winfo_reqheight()) / 2.35
window.wm_geometry("+%d+%d" % (x, y))

input_text = Entry(window, width=50)
input_text.pack(anchor=N)

btn = Button(window, text='Получить данные', width=15, height=5,
command=data_site).pack()

window.mainloop()

```

Вывод:

Выполнив данную лабораторную работу я узнал о такой библиотеке, как “requests”. При помощи данной библиотеки, я создал запрос для сайт и после обработал данные, которые он мне вернул, путём собирания нужных при помощи переменных. Получив знания я выполнил практическую работу и научился работать с запросами.

При выполнении лабораторной работы использовалась среда программирования *PyCharm* (см. рис.1-3). На рис. представлен исходный код программы и пример его работы.



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНЖЕНЕРНЫХ ТЕХНОЛОГИЙ»

Факультет управления и информатики в технологических системах
Кафедра информационной безопасности
Направление подготовки (специальность) 10.05.03 Информационная безопасность
автоматизированных систем

Отчет
По технологиям и методам программирования.
Практическая работа №12

наименование (вид) дисциплины

Выполнил студент гр. УБ-21
Заварзин В.Д.
(ф.и.о.)

Проверил:

Доцент каф. ИБ Маслов А.А._____

(оценка)

(подпись)

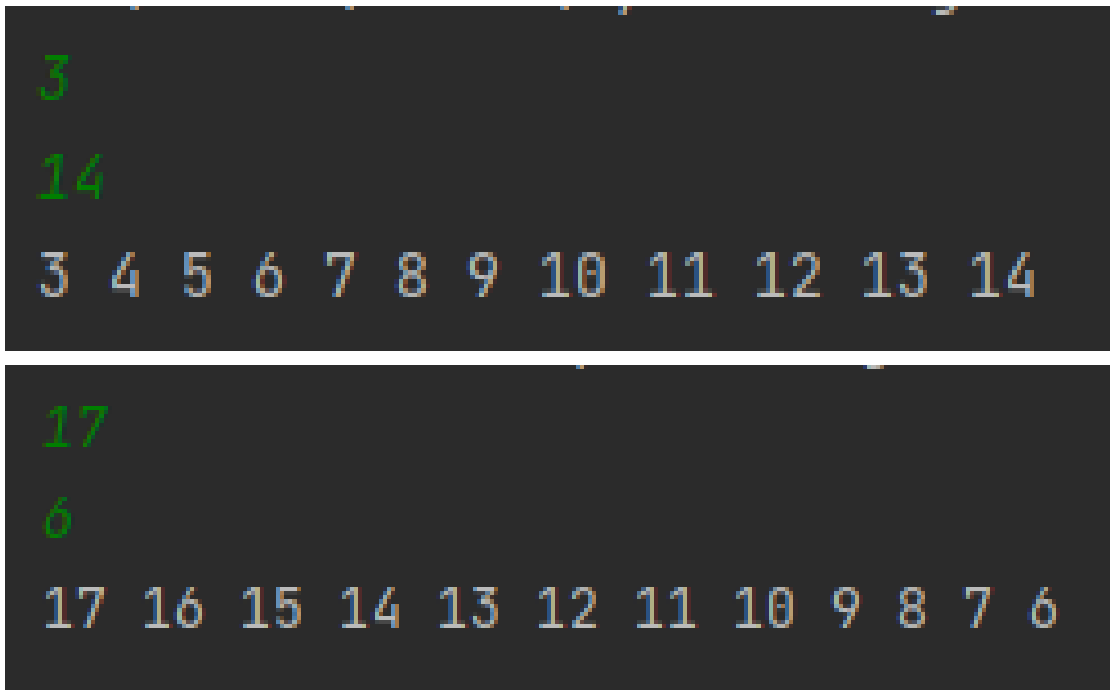
(дата)

Формулировка:

7. Даны два целых числа A и B (каждое в отдельной строке). Выведите все числа от A до B включительно, в порядке возрастания, если $A < B$, или в порядке убывания в противном случае.

Задание 1

```
def rec(a, b):  
    if a < b:  
        print(a, end=' ')  
        rec(a+1, b)  
    elif b < a:  
        print(a, end=' ')  
        rec(a-1, b)  
    if a==b:  
        print(a)  
  
a = int(input())  
b = int(input())  
rec(a, b)
```



Программа и пример её работы (рис.1-3)

Код программы:

```
def rec(a, b):  
    if a < b:  
        print(a, end=' ')  
        rec(a+1, b)  
    elif b < a:  
        print(a, end=' ')  
        rec(a-1, b)  
    if a==b:  
        print(a)  
a = int(input())  
b = int(input())  
rec(a, b)
```

Задание 2

Формулировка:

1. Вводим последовательность натуральных чисел (одно число в строке), завершающаяся числом 0. Определите наибольшее значение числа в этой последовательности. В этой задаче нельзя использовать глобальные переменные и передавать какие-либо параметры в рекурсивную функцию. Функция получает данные, считывая их с клавиатуры. Функция возвращает единственное значение: максимум считанной последовательности. Гарантируется, что последовательность содержит хотя бы одно число (кроме нуля).


```
def posl():
    n = int(input())
    if (n == 0):
        return 0
    else:
        return max(n, posl())
print(posl())
```

1
4
6
9
2
5
4
0
9

Программа и пример её работы (рис. 4-5)

Код программы:

```
def posl():
    n = int(input())
    if (n == 0):
        return 0
    else:
        return max(n, posl())
print(posl())
```

Вывод:

Выполнив данную практическую работу я узнал о таком приёме программирования, как “рекурсия”. Разобравшись с её значением, я использовал её в решении задач при работе.

При выполнении лабораторной работы использовалась среда программирования *PyCharm* (см. рис.1-5). На рис. представлен исходный код программы и пример его работы.