2.

Разработать графический интерфейс для ввода интервала значений Х для   
построения графиков функций (см. лабораторная работа 5, п.1). Для выбора функции   
использовать Combobox.

Code:

from tkinter import \*

from tkinter.ttk import Combobox

import matplotlib.pyplot as plt

import numpy as np

def BuildGraph(xmin, xmax, funct):

    x = np.arange(xmin, xmax, round((xmax-xmin)/100, 2))

    match funct:

        case "x": y = x

        case 'x^2': y = x\*\*2

        case "x\*\*2 - 8\*x + 15": y = x\*\*2 - 8\*x + 15

        case '(x\*\*2 - 6\*x + 10)\*(x>=1) + (x + 2)\*(x<1)': y = (x\*\*2 - 6\*x + 10)\*(x>=1) + (x + 2)\*(x<1)

    plt.plot(x,y)

    plt.show()

    print (xmin, xmax, funct)

def clicked():

    try:

        xmin = int(xmin\_.get())

        xmax = int(xmax\_.get())

    except: xmin, xmax = 0, 0

    funct = funct\_.get()

    BuildGraph(xmin, xmax, funct)

# Window manager

window = Tk()

xmin\_ = Entry(window, width = 10)

xmin\_.grid(column=1, row=0)

xmax\_ = Entry(window, width = 10)

xmax\_.grid(column=2, row=0)

funct\_ = Combobox(window, values=('x', 'x^2', 'x\*\*2 - 8\*x + 15', '(x\*\*2 - 6\*x + 10)\*(x>=1) + (x + 2)\*(x<1)'))

funct\_.current(0)

funct\_.grid(column = 3, row=0)

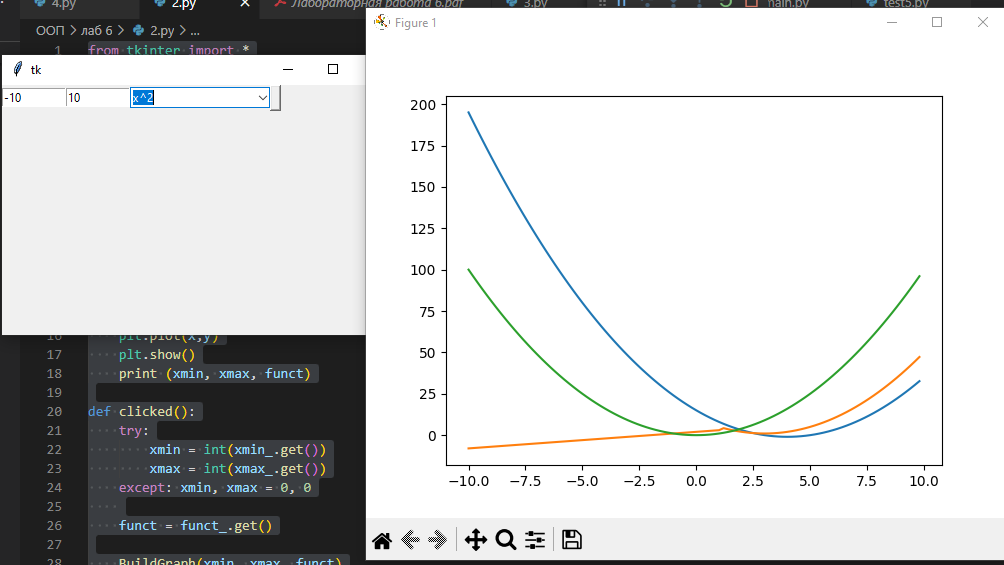
xmin\_btn = Button(window, command=clicked)

xmin\_btn.grid(column=4, row=0)

window.geometry('400x250')

window.mainloop()

result:



3. Разработать форму для изменения данных в XML-файле, созданного в лабораторной   
работе No3, п.1. Поиск абитуриента осуществлять по фамилии или шифру   
абитуриента (использовать Radiobutton). Для выбора вступительного(ых)   
испытания(й), по которому(ым) нужно внести изменения, используйте Checkbutton.   
Для подтверждения корректности внесенных пользователем данных используйте   
MessageBox.

Code:

import xml.etree.ElementTree as ET

from tkinter import \*

from tkinter.ttk import Radiobutton

from tkinter.ttk import Checkbutton

tree = ET.parse('лаб 6/1.xml')

root = tree.getroot()

def clicked():

    for i in entryes:

        try:

            if 100 <= int(i.get()) >= 0:

                lbl.configure(text='invalid score')

                return

        except: pass

    for enrolle in root.findall(f"enrolle[name='{students\_names[selected.get()]}']"):

        for i in range(len(exam\_names)):

            try:

                enrolle.find(exam\_names[i]).text

            except:

                new\_exam = ET.Element(exam\_names[i])

                root.find(f"enrolle[name='{students\_names[selected.get()]}']").append(new\_exam)

                tree.write('лаб 6/1.xml', encoding="UTF-8")

            if checkbuttons\_state[i].get(): enrolle.find(exam\_names[i]).text = entryes[i].get()

    tree.write('лаб 6/1.xml', encoding="UTF-8")

    lbl.configure(text='done')

def get\_names():

    global students\_names

    global exam\_names

    students\_names = list({elem.findtext('name') for elem in root.iter()})

    students\_names.remove(None)

    students\_names.sort()

    exam\_names = list({elem.tag for elem in root.findall('enrolle/\*')})

    exam\_names.remove('name')

    exam\_names.sort()

# window manager

window = Tk()

get\_names()

radiobuttons = []

selected = IntVar()

for i in range(len(students\_names)):

    radiobuttons.append(Radiobutton(window, text = students\_names[i], value = i, variable=selected))

    radiobuttons[i].grid(column = 0, row = i)

checkbuttons\_state = []

checkbuttons = []

entryes = []

for i in range(len(exam\_names)):

    checkbuttons\_state.append(BooleanVar())

    checkbuttons.append(Checkbutton(window, text=exam\_names[i], var = checkbuttons\_state[i]))

    checkbuttons[i].grid(column = 1, row = i)

    entryes.append(Entry(window, width = 10))

    entryes[i].grid(column = 2, row = i)

submit\_btn = Button(window, command=clicked, width=10)

submit\_btn.grid(column=0, row=len(students\_names))

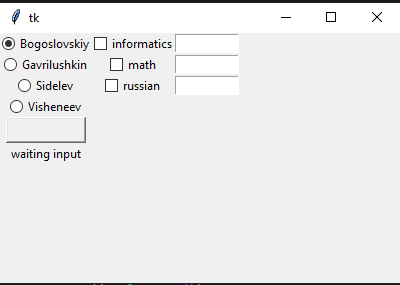
lbl = Label(window,text='waiting input')

lbl.grid(column=0, row=len(students\_names)+1)

window.geometry('400x250')

window.mainloop()

result:



4. Разработать форму для добавления новых данных в XML-файл (лабораторная   
работа No3, п.1)

Code:

import xml.etree.ElementTree as ET

from tkinter import \*

from tkinter.ttk import Radiobutton

from tkinter.ttk import Checkbutton

tree = ET.parse('лаб 6/1.xml')

root = tree.getroot()

def clicked():

    if student\_name.get() in students\_names:

        lbl.configure(text='student already exists')

        return

    for i in entryes:

        try:

            if 100 <= int(i.get()) >= 0:

                lbl.configure(text='invalid score')

                return

        except: pass

    new\_student = ET.Element('enrolle')

    new\_student.set('id', str(len(students\_names) + 1))

    ET.SubElement(new\_student, 'name').text = student\_name.get()

    for i in range(len(exam\_names)):

        if checkbuttons\_state[i].get():

            print("works")

            ET.SubElement(new\_student, exam\_names[i]).text = entryes[i].get()

    root.append(new\_student)

    tree.write('лаб 6/1.xml', encoding="UTF-8")

    get\_names()

    lbl.configure(text='done')

def get\_names():

    global students\_names

    global exam\_names

    students\_names = list({elem.findtext('name') for elem in root.iter()})

    students\_names.remove(None)

    students\_names.sort()

    exam\_names = list({elem.tag for elem in root.findall('enrolle/\*')})

    exam\_names.remove('name')

    exam\_names.sort()

# Window manager

window = Tk()

get\_names()

student\_name = Entry(window, width = 10)

student\_name.grid(column=1, row = 0)

student\_name\_lbl = Label(window, text = 'student\_name').grid(column=0, row=0)

checkbuttons\_state = []

checkbuttons = []

entryes = []

for i in range(len(exam\_names)):

    checkbuttons\_state.append(BooleanVar())

    checkbuttons.append(Checkbutton(window, text=exam\_names[i], var = checkbuttons\_state[i]))

    checkbuttons[i].grid(column = 0, row = i + 2)

    entryes.append(Entry(window, width = 10))

    entryes[i].grid(column = 1, row = i + 2)

submit\_btn = Button(window, command=clicked, width=10)

submit\_btn.grid(column=0, row=len(exam\_names)+2)

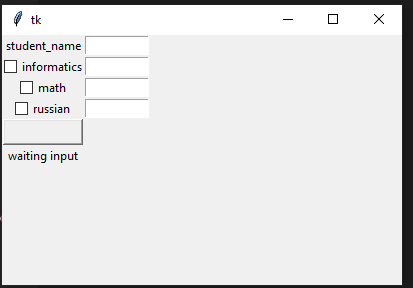
lbl = Label (window, text='waiting input')

lbl.grid(column=0, row=len(exam\_names)+3)

window.geometry('400x250')

window.mainloop()

result:



xml for 3 and 4:

<enrolles>

    <enrolle id="1">

        <name>Gavrilushkin</name>

        <math>74</math>

        <russian>78</russian>

        <informatics>123</informatics>

    </enrolle>

    <enrolle id="2">

        <name>Sidelev</name>

        <math>0</math>

        <russian>95</russian>

        <informatics>50</informatics>

    </enrolle>

    <enrolle id="3">

        <name>Visheneev</name>

        <math>50</math>

        <russian>50</russian>

        <informatics>50</informatics>

    </enrolle>

    <enrolle id="4">

        <name>Bogoslovskiy</name>

        <math>33</math>

        <russian>24</russian>

        <informatics>123</informatics>

    </enrolle>

</enrolles>