­­МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

**«Вятский государственный университет»**

**(ФГБОУ ВО «ВятГУ»)**

Факультет автоматики и вычислительной техники

Кафедра электронных вычислительных машин

Отчет по лабораторной работе №8 дисциплины

«Разработка программных систем»

Вариант 2

Выполнил студент группы ИВТб-31\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Жеребцов К. А./

Проверил преподаватель\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Чистяков Г.А. /

Киров 2023

1. Постановка задачи

В соответствии с выбранной тематикой разработать клиентское

приложение, делегирующее выполнение вычислительных действий

серверному приложению, и серверное приложение, выполняющее

вычисления клиента. Для организации взаимосвязи между клиентом и

сервером использовать протокол XML-RPC.

Для выполнения лабораторной работы необходимо решить следующие

задачи.

 Согласовать тематику разработки с преподавателем.

 Разработать структуры клиентского и серверного приложений.

 Реализовать приложения.

 Продемонстрировать работу приложения.

2. Ход работы

2.1 Модуль server.py

from xmlrpc.server import SimpleXMLRPCServer  
  
  
with SimpleXMLRPCServer(('localhost', 8080)) as server:  
  
 server.register\_function(lambda main\_num, sec\_num: main\_num \* sec\_num, 'multiply\_num')  
 server.register\_function(lambda main\_den, sec\_den: main\_den \* sec\_den, 'multiply\_den')  
  
 server.register\_function(lambda main\_num, sec\_den: main\_num \* sec\_den, 'divide\_num')  
 server.register\_function(lambda main\_den, sec\_num: main\_den \* sec\_num, 'divide\_den')  
  
 server.register\_function(lambda main\_num, sec\_num: main\_num + sec\_num, 'sum\_num')  
 server.register\_function(lambda main\_num, sec\_num: main\_num - sec\_num, 'sub\_num')  
  
 server.register\_function(lambda main\_num, sec\_den: main\_num \* sec\_den, 'num\_main')  
 server.register\_function(lambda main\_den, sec\_num: sec\_num \* main\_den, 'num\_sec')  
  
 server.serve\_forever()

2.2 Модуль client.py

import xmlrpc.client  
import fractions  
from tkinter import \*  
from tkinter.ttk import Combobox  
import re  
  
def clicked():  
 lbl\_hint.configure(text= "Введите дроби", foreground='black')  
 if (txt\_num1.get() != "") & (txt\_den1.get() != "") & (txt\_num2.get() != "") & (txt\_den2.get() != ""):  
 try:  
 num = txt\_num1.get()  
 num = re.sub('[^0-9-]', '', num)  
 int(num)  
 except:  
 lbl\_hint.configure(text="Неверный числитель 1!", foreground='red')  
  
 try:  
 den = txt\_den1.get()  
 den = re.sub('[^0-9-]', '', den)  
 int(den)  
 except:  
 lbl\_hint.configure(text="Неверный знаменатель 1!", foreground='red')  
  
 # num = txt\_num1.get()  
 # den = txt\_den1.get()  
 if int(den) != 0:  
 main = fractions.Fraction(int(num), int(den))  
 else:  
 lbl\_hint.configure(text="Знаменатель = 0!", foreground='red')  
  
  
 try:  
 num = txt\_num2.get()  
 num = re.sub('[^0-9-]', '', num)  
 int(num)  
 except:  
 lbl\_hint.configure(text="Неверный числитель 2!", foreground='red')  
  
 try:  
 den = txt\_den2.get()  
 den = re.sub('[^0-9-]', '', den)  
 int(den)  
 except:  
 lbl\_hint.configure(text="Неверный знаменатель 2!", foreground='red')  
  
 # num = txt\_num2.get()  
 # den = txt\_den2.get()  
 if int(den) != 0:  
 sec = fractions.Fraction(int(num), int(den))  
 else:  
 lbl\_hint.configure(text="Знаменатель = 0!", foreground='red')  
  
 if combo.get() == "Сложение":  
 if main.denominator == sec.denominator:  
 num = s.sum\_num(main.numerator, sec.numerator)  
 den = main.denominator  
 res = fractions.Fraction(num, den)  
 elif main.denominator != sec.denominator:  
 num1 = s.num\_main(main.numerator, sec.denominator)  
 num2 = s.num\_sec(main.denominator, sec.numerator)  
 den = s.multiply\_den(main.denominator, sec.denominator)  
 res = fractions.Fraction(num1 + num2, den)  
 if combo.get() == "Вычитание":  
 if main.denominator == sec.denominator:  
 num = s.sub\_num(main.numerator, sec.numerator)  
 den = main.denominator  
 res = fractions.Fraction(num, den)  
 elif main.denominator != sec.denominator:  
 num1 = s.num\_main(main.numerator, sec.denominator)  
 num2 = s.num\_sec(main.denominator, sec.numerator)  
 den = s.multiply\_den(main.denominator, sec.denominator)  
 res = fractions.Fraction(num1 - num2, den)  
 if combo.get() == "Умножение":  
 num = s.multiply\_num(main.numerator, sec.numerator)  
 den = s.multiply\_den(main.denominator, sec.denominator)  
 res = fractions.Fraction(num, den)  
 if combo.get() == "Деление":  
 if sec.numerator != 0:  
 num = s.divide\_num(main.numerator, sec.denominator)  
 den = s.divide\_den(main.denominator, sec.numerator)  
 res = fractions.Fraction(num, den)  
 elif sec.numerator == 0:  
 lbl\_hint.configure(text= "Деление на 0!", foreground='red')  
 else:  
 lbl\_hint.configure(foreground='red')  
  
 try:  
 string = str(res.numerator) + "/" + str(res.denominator)  
 lbl\_res1.configure(text=string)  
 except:  
 lbl\_hint.configure(text="Деление на 0!", foreground='red')  
  
  
if \_\_name\_\_ == '\_\_main\_\_':  
  
 s = xmlrpc.client.ServerProxy('http://localhost:8080')  
  
 window = Tk()  
 window.title("Lab7")  
 window.geometry('600x150')  
  
 lbl\_num1 = Label(window, text="1 числитель")  
 lbl\_num1.grid(column=0, row=0)  
 lbl\_den1 = Label(window, text="1 знаменатель")  
 lbl\_den1.grid(column=0, row=1)  
  
 lbl\_num2 = Label(window, text="2 числитель")  
 lbl\_num2.grid(column=0, row=2)  
 lbl\_den2 = Label(window, text="2 знаменатель")  
 lbl\_den2.grid(column=0, row=3)  
  
 lbl\_res = Label(window, text="Результат")  
 lbl\_res.grid(column=0, row=4)  
  
 lbl\_hint = Label(window, text="Введите дроби")  
 lbl\_hint.grid(column=1, row=5)  
  
 txt\_num1 = Entry(window, width=50)  
 txt\_num1.grid(column=1, row=0)  
 txt\_den1 = Entry(window, width=50)  
 txt\_den1.grid(column=1, row=1)  
  
 txt\_num2 = Entry(window, width=50)  
 txt\_num2.grid(column=1, row=2)  
 txt\_den2 = Entry(window, width=50)  
 txt\_den2.grid(column=1, row=3)  
  
 lbl\_res1 = Label(window, width=50)  
 lbl\_res1.grid(column=1, row=4)  
  
 combo = Combobox(window, width=20, state="readonly")  
 combo['values'] = ('Сложение', 'Вычитание', 'Умножение', 'Деление')  
 combo.current(0)  
 combo.grid(column=2, row=2)  
  
 btn = Button(window, width=20, text="Вычислить!", command=clicked)  
 btn.grid(column=2, row=0)  
 window.mainloop()

3. Экранные формы

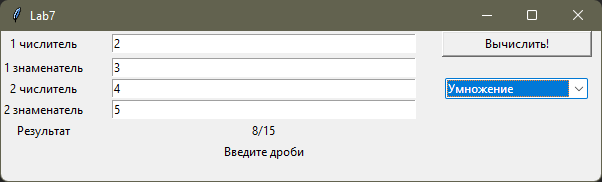


Рисунок 1 – Окно приложения

Вывод

В результате выполнения лабораторной работы были получены навыки работы с протоколом XML-RPC, а также навыки применения этого протокола для организации серверного взаимодействия.