Министерство образования и науки Российской Федерации Федеральное агентство по образованию Федеральное государственное бюджетное образовательное учреждение высшего образования «Вятский государственный университет»

Факультет автоматики и вычислительной техники

Кафедра электронных вычислительных машин

Лабораторная работа №8

по курсу «Разработка программных систем»

Выполнил студент группы ИВТ-31\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Птахова А.М/

Проверил доцент кафедры ЭВМ \_\_\_\_\_\_\_\_\_\_\_\_\_\_/ Чистяков Г.А./

Киров 2023

1. Задание

В ходе выполнения лабораторной работы необходимо решить следующие задачи:

- Согласовать тематику с преподавателем;

- Разработать структуру серверного и клиентского приложений;

- Реализовать приложения;

- Продемонстрировать работу приложений.

2. Листинг программы

**server.py**

from xmlrpc.server import SimpleXMLRPCServer

from primerandom import get\_random\_array, get\_random\_number

def main():

server = SimpleXMLRPCServer(("localhost", 8080))

server.register\_function(get\_random\_number)

server.register\_function(get\_random\_array)

server.serve\_forever()

if \_\_name\_\_ == '\_\_main\_\_':

main()

**app.py**

import tkinter as tk

from tkinter import messagebox

from xmlrpc.client import ServerProxy

class ReadOnlyText(tk.Text):

def \_\_init\_\_(self, \*args, \*\*kwargs):

super().\_\_init\_\_(\*args, \*\*kwargs)

self['state'] = tk.DISABLED

def set\_text(self, text: str):

self['state'] = tk.NORMAL

self.delete('0.0', tk.END)

self.insert('0.0', text)

self['state'] = tk.DISABLED

class ArrayFrame(tk.Frame):

def \_\_init\_\_(self, server, parent=None):

super().\_\_init\_\_(master=parent)

self.\_server = server

self.\_build\_ui()

def \_build\_ui(self):

params\_frame = tk.Frame(self)

array\_len\_label = tk.Label(params\_frame, text='Длина массива')

array\_len\_label.pack(side=tk.LEFT)

self.\_array\_len\_var = tk.StringVar()

array\_len\_entry = tk.Entry(params\_frame, textvariable=self.\_array\_len\_var, width=20)

array\_len\_entry.pack(side=tk.LEFT)

gen\_array\_button = tk.Button(params\_frame, text='Выдать список', command=self.\_gen\_array)

gen\_array\_button.pack(side=tk.LEFT)

params\_frame.pack(side=tk.TOP)

array\_frame = tk.Frame(self)

scrollbar = tk.Scrollbar(array\_frame)

scrollbar.pack(side=tk.RIGHT, fill=tk.Y)

self.\_array\_text = ReadOnlyText(array\_frame, yscrollcommand=scrollbar.set)

scrollbar['command'] = self.\_array\_text.yview

self.\_array\_text.pack(side=tk.LEFT)

array\_frame.pack(side=tk.TOP)

def \_gen\_array(self):

length = self.\_array\_len\_var.get()

if length.strip().isdecimal() and 1 <= int(length) <= 100000:

array = self.\_server.get\_random\_array(int(length))

self.\_array\_text.set\_text(self.\_dump\_json(array))

else:

messagebox.showerror('Ошибка', 'Длина массива должна быть в диапазоне [1; 100000]')

@staticmethod

def \_dump\_json(array)->str:

mod\_str = ''

for arr in array:

mod\_str = mod\_str + str(arr)+'\n'

return mod\_str

class NumberFrame(tk.Frame):

def \_\_init\_\_(self, server, parent=None):

super().\_\_init\_\_(master=parent)

self.\_server = server

self.\_build\_ui()

def \_build\_ui(self):

gen\_number\_button = tk.Button(self, text='Выдать простое случайное число', command=self.\_gen\_number)

gen\_number\_button.pack(side=tk.LEFT)

self.\_number\_var = tk.StringVar()

number\_entry = tk.Entry(self, textvariable=self.\_number\_var, state='readonly')

number\_entry.pack(side=tk.LEFT)

def \_gen\_number(self):

self.\_number\_var.set(self.\_server.get\_random\_number())

class Application(tk.Tk):

def \_\_init\_\_(self):

super().\_\_init\_\_()

self.\_server = ServerProxy('http://127.0.0.1:8080')

self.\_build\_ui()

def \_build\_ui(self):

self.title('Генератор простых случайных чисел')

self.\_number\_frame = NumberFrame(self.\_server, self)

self.\_number\_frame.pack(side=tk.TOP)

self.\_array\_frame = ArrayFrame(self.\_server, self)

self.\_array\_frame.pack(side=tk.TOP)

if \_\_name\_\_ == '\_\_main\_\_':

app = Application()

app.mainloop()

3. Вывод

В ходе выполнения лабораторной работы были изучены основы возможности протокола XML-RPC. Разработано серверное приложение для генерации простых случайных чисел, для взаимодействия с которым использовался протокол. На основе предыдущей лабораторной работы разработано клиентское приложение, передающее выполнение вычислительных действий серверному приложению. Для организации взаимодействия между приложениями был использован модуль xmlrpc.