

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ
ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Вятский государственный университет»
Факультет автоматики и вычислительной техники
Кафедра электронных вычислительных машин

Лабораторная работа №7 по курсу
«Разработка программных систем»

Выполнил студент группы ИВТ-32 _____ /Рзаев А. Э./
Проверил доцент кафедры ЭВМ _____ /Чистяков Г. А./

Киров 2017

1 Задание

В ходе выполнения лабораторной работы необходимо решить следующие задачи:

- Согласовать тематику разработки с преподавателем.
- Разработать структуру приложения.
- Разработать графический интерфейс пользователя.
- Реализовать приложение.
- Продемонстрировать работу приложения.

2 Листинг программы

Листинг программы приведен в приложении А.

3 Вывод

В ходе выполнения лабораторной работы был изучен основной функционал виджетов библиотеки tkinter (Frame, Entry, Label, Text, Root); на основе tk.Text разработан собственный виджет с поддержкой режима "только чтение". С использованием библиотек tkinter и json разработано приложение с графическим интерфейсом пользователя, предназначенное для создания списков случайных чисел, представляемых в форматах JSON и XML (на выбор пользователя).

Приложение А
(обязательное)
Листинг программы

lab7.pyw

```
import tkinter as tk
from tkinter import messagebox

import random

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, parent=None):
        super().__init__(master=parent)
        self._build_ui()

    def _build_ui(self):
        params_frame = tk.Frame(self)
        array_len_label = tk.Label(params_frame, text='Length of array')
        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='Get array', command=self._gen_array)
        gen_array_button.pack(side=tk.LEFT)
        params_frame.pack(side=tk.TOP)

        type_frame = tk.Frame(self)
        type_label = tk.Label(type_frame, text='Data format:')
        type_label.pack(side=tk.LEFT)

        self._type_var = tk.StringVar()
        type_json = tk.Radiobutton(type_frame, text='JSON', variable=self._type_var, value='JSON')
        type_json.select()
        type_json.pack(side=tk.LEFT)

        type_xml = tk.Radiobutton(type_frame, text='XML', variable=self._type_var, value='XML')
        type_xml.pack(side=tk.LEFT)

        type_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 = primerandom.get_random_array(int(length))
            if self._type_var.get() == 'JSON':
                self._array_text.set_text(self._dump_json(array))
            else:
                self._array_text.set_text(self._dump_xml(array))
        else:
            messagebox.showerror('Error', 'Length of array must be in [1; 100000]')

    @staticmethod
    def _dump_json(array):
        from json import dumps
        return dumps(array, indent=2)

    @staticmethod
    def _dump_xml(array):
        return '<?xml version="1.0" ?>\n' + \
            '<array>\n' + \
            '\n'.join('    <item>{}\n</item>'.format(v) for v in array) + \
            '\n</array>'

class NumberFrame(tk.Frame):
    def __init__(self, parent=None):
        super().__init__(master=parent)
        self._build_ui()

    def _build_ui(self):
        gen_number_button = tk.Button(self, text='Get prime random number',
                                       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(primerandom.get_random_number())

class Application(tk.Tk):
    def __init__(self):
        super().__init__()
        self._build_ui()

    def _build_ui(self):
        self.title('Prime random number generator')
        self._number_frame = NumberFrame(self)
        self._number_frame.pack(side=tk.TOP)
        self._array_frame = ArrayFrame(self)
        self._array_frame.pack(side=tk.TOP)

if __name__ == '__main__':
    app = Application()
    app.mainloop()

```