Министерство образования и науки Республики Казахстан

Северо-Казахстанский университет им. М. Козыбаева

Проектная работа 2

На тему:

«Программирование виджетов с помощью tkinter и разработка простейшей

ИС»

Группа №4

|  |  |
| --- | --- |
| Выполнили студенты  группы АПО-19 | Шамсудинов В.В.  Сахаров А.В.  Буряк Р.С.  Сыздыков Р.С. |
| Научный руководитель  доцент, PhD | Астапенко Н.В. |

Петропавловск, 2022

ОГЛАВЛЕНИЕ

[Введение 3](#_Toc97107715)

[1. Постановка задачи 4](#_Toc97107716)

[2. Распределение работ 4](#_Toc97107717)

[3. Архитектура программы 5](#_Toc97107718)

[4. Описание используемых алгоритмов 6](#_Toc97107719)

[5. Тестирование и анализ результатов 6](#_Toc97107720)

[Заключение 8](#_Toc97107721)

[Используемая литература 9](#_Toc97107722)

[ПРИЛОЖЕНИЕ А 9](#_Toc97107723)

## 

## Введение

Цель: Создание виджетов с помощью tkinter и разработка простейшей ИС. Разработка программы для скоро помощи больницы.

Оборудование: Visual Studio Code, GitHub.

Описание предметной области.

Лечебное учреждение (Код, Название, Краткое название, Адрес, Контактные

телефоны) оказывает скорую медицинскую помощь населению. В учреждении

имеется штат сотрудников, о которых необходимо хранит следующие сведения:

- табельный номер;

- ФИО; дата рождения, пол;

- должность, дата начала работы в данной должности, дата окончания,

ставка.

Работа в учреждении круглосуточная – сотрудники работают по 24 часа с

последующими выходными днями. Необходимо знать, в какой смене и бригаде

работает тот или иной сотрудник. Закрепление в бригаду осуществляется на

основании внутреннего приказа, имеющего номер и дату. В каждой позиции

приказа указывается, что конкретный сотрудник с даты 1 по дату 2 работает в

бригаде с заданным номером.

Необходимо вести учет сведений о выездах бригад на вызовы. Каждый вызов

определяется датой, временем выезда и адресом. Пациент, которому оказывается

помощь, может быть описан следующими данными ФИО, возраст (примерный),

первоначальный диагноз. Необходимо также знать ФИО и должности

сотрудников выехавшей на вызов бригады (включая водителя и диспетчера).

Необходимо также хранить небольшое текстовое описание принятых бригадой

мер. Если больной был госпитализирован, либо получил направление на

госпитализацию, также необходимо знать в какое учреждение он был направлен

(название, адрес). При возвращении бригады фиксируется время прибытия.

Необходимо осуществлять следующую обработку данных:

- на заданную дату список выездов всех бригад учреждения (номер

выезда, время, номер бригады, принятые меры);

- на заданную дату описание самого длительного выезда;

- на заданную дату список заданной бригады (табельный номер, ФИО,

должность).

## 1. Постановка задачи

На проектный период перед командой были поставлены следующие задачи:

1. Ознакомиться с Лабораторной работой №2. Изучить минимальные требования к проекту.
2. Выбрать тему проекта.
3. Распределить обязанности между членами группы.
4. Ознакомится с модулем tkinter.
5. Составить примерный план проекта.
6. Начать писать код проекта.
7. Реализация минимальных требований проекта.
8. Доработка проекта.
9. Обучение членов команды основным функциям проекта.
10. Обучение членов команды классам.
11. Конечная проверка проекта и проверка подготовленности членов команды.

Выполнено из поставленных задач:

1. Ознакомились с Лабораторной работой №2. Изучили минимальные требования к проекту. Ознакомились с предоставленным материалом.

2. Выбрали тему проекта. В качестве темы выбрали автоматизированную информационную систему «Скорая помощь» для больницы.

1. Распределили обязанности между членами группы.
2. Составили примерный план проекта. Представили примерный вид программы.
3. Написали работающий проект.
4. Реализация минимальных требований проекта. Реализовали еще один класс.
5. Доработка проекта. Доработали движения мыши. Доработали движение кота. Добавили объект, который двигается по математической формуле.
6. Обучили членов команды основным функциям проекта.
7. Проверили готовность проекта и проверили подготовленности членов команды.

## 2. Распределение работ

На первом этапе работы была поставлена задача выбора метода реализации требований проекта. После формирования общего видения и концепции продукта, мы приступили к распределению ролей и обязанностей.

Работа над проектом в составе группы должна иметь и содержать распределение работ, то есть распределение функциональных обязанностей между участниками группы.

Участниками группы являются:

* Алексей Сахаров – тимлид;
* Роман Буряк;
* Раимбек Сыздыков;
* Владислав Шамсутдинов;

Тимлид как роль в проекте занимается непосредственным распределением задач между участниками проекта. Распределение работ представлено в соответствии с таблицей 1.

|  |  |
| --- | --- |
| Участник | Задача |
| Алексей Сахаров | Создание архитектуры программы  Реализация участков кода |
| Роман Буряк | Создание архитектуры программы  Реализация участков кода |
| Раимбек Сыздыков | Знакомство с Git |
| Владислав Шамсутдинов | Знакомство с модулем Turtle |

Таблица 1. Главные задачи и их исполнитель.

## 3. Архитектура программы

Архитектура программы отображает как программа была реализована и из чего она состоит. Архитектура программы представлена в соответствии с рисунком 1.

|  |
| --- |
|  |

Рисунок 1. Архитектура программы.

Использование выбранной архитектуры проекта гарантирует работоспособность готового продукта.

## 4. Описание используемых алгоритмов

Программа будет состоять из следующих функциональных частей:

* Основное тело программы
* Класс, описывающий окно программы в котором происходят первые действия пользователя
* Класс, описывающий дочернее окно «о программе»

## 5. Тестирование и анализ результатов

При тестировании программы был произведен подробный анализ каждой подпрограммы.

При запуске программы появилось окно программы.

Окно программы представлено в соответствии с рисунком 2.

|  |
| --- |
| https://sun9-75.userapi.com/impg/AAGZ_OyQW_XbR5qphiy5j2vzLvcTJlupwfRBoQ/6PSPbJj1dM4.jpg?size=252x232&quality=96&sign=aba40782aa5df26d018b491549701e76&type=album |

Рисунок 2. Главное окно.

Главное окно появляется сразу после запуска программы.

После нажатия кнопки «О программе» появляется окно, показанное на рисунке 3.

|  |
| --- |
| https://sun9-1.userapi.com/impg/z9guwz9NnWa6J099xntQi9kybw9-qj3f7KrITA/m3AbCAcEjX4.jpg?size=402x252&quality=96&sign=31b04985695a1e9345f51b7ab827b197&type=album |

Рисунок 3. Окно о программе.

После нажатия кнопки «Вызовы» появляется окно, показанное на рисунке 3.

|  |
| --- |
| https://sun2.beeline-kz.userapi.com/impg/xsNbA8ksuFzF5skHnKsualNf_TJ94NLw0ynR-Q/fyUT9PWbSW8.jpg?size=602x632&quality=96&sign=885b2a5f6f0157b29361681718de2d1e&type=album |

Рисунок 4. Окно вызовы.

## Заключение

В ходе выполнения курсовой работы были изучены основы ООП модуля TKinter. на языке Python, а так же изучен порядок работы с GitHub. Написание программы способствовало закреплению теоретического материала членов группы на практике.

Был получен опыт работы в команде, с четким разделением обязанностей. Данный подход показал повышение в скорость разработки и отладки приложения. Каждый член команды отвечал за выполнение только своих задач, что гарантировало качество выполнения

## Используемая литература

1. GitHub Actions для автоматической проверки кода:

− https://www.youtube.com/watch?v=NijFSs03Pd4

2. Модуль tkinter.

− https://stepik.org/lesson/349987/step/1?unit=333841

− Рядченко, В.П. Программирование на языке высокого уровня Python: учебно-

методическое пособие / В.П. Рядченко, Л.М. Эльканова, Л.М. Шавтикова. –

Черкесск: БИЦ СевКавГГТА, 2018. –144с.

− Федоров Д.Ю. Основы программирования на примере языка Python: учебное

пособие. – СПб.: Питер, 2019. – 152 с.

3. Файл .csv

− Мэтиз Э. Изучаем Python. Программирование игр, визуализация данных,

веб-приложения. — СПб.: Питер, 2017. — 496 с.: ил. — (Серия «Библиотека

программиста»)

− Федоров Д.Ю. Основы программирования на примере языка Python: учебное

пособие. – СПб.: Питер, 2019. – 152 с.

## ПРИЛОЖЕНИЕ А

Код программы

import tkinter as tk

from tkinter import \*

#Создаём главное меню и объекты на нём

class Main(tk.Frame):

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

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

self.init\_main()

def init\_main(self):

toolbar = tk.Frame(bg='#d7d8e0', bd=2)

toolbar.pack(side=tk.TOP, fill=tk.X)

# Не мог сделать

btn\_open\_dialog = tk.Button(toolbar, text='О программе', bg='#d7d8e0', compound=tk.TOP, command = self.vizov\_inf)

btn\_open\_dialog.pack(side=tk.LEFT)

btn\_svedenya = tk.Button(text='Список вызовов\nвсех бригад', compound=tk.TOP, width = 16, command = self.v\_inf)

btn\_svedenya.pack()

btn\_vizov = tk.Button(text='Сведения о самом\nдлительном вызове', compound=tk.TOP, width = 16)

btn\_vizov.pack()

btn\_brigada = tk.Button(text='Сведения о бригаде\nпо дате', compound=tk.TOP, width = 16)

btn\_brigada.pack()

btn\_ex = tk.Button(text='Выход', compound=tk.TOP, width = 16)

btn\_ex.pack()

def vizov\_inf(self):

info()

def v\_inf(self):

viz\_info()

#Создаём форму на которой будет информация о программе

class info(tk.Toplevel):

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

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

self.init\_info()

def init\_info(self):

self.title('Информация о программе')

self.geometry('400x220')

self.resizable(False, False)

label\_name = tk.Label(self, text="Скорая помощь", font =('Times New Roman', 25))

label\_name.place(x=100, y=10)

label\_info = tk.Label(self,text="Программа была разработана 4-ой минигруппой,\n с использованием python и Tkinter \n для контроля бригад скорой помощи")

label\_info.place(x=70, y=70)

#Создаём форму где будет храниться информация о вызовах

class viz\_info(tk.Toplevel):

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

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

self.v\_info()

def v\_info(self):

f = open('baza.txt', 'r')

line = f.read()

self.title('Вызовы')

self.geometry('600x600')

self.resizable(False, False)

label\_name = tk.Label(self, text="Введите дату")

label\_name.place(x=10, y=6)

text\_info = tk.Entry(self)

text\_info.place(x=100, y=10)

btn\_ent = tk.Button(self,text='Ввод', compound=tk.TOP, width=16)

btn\_ent.place(x=260, y=10)

box = Listbox(self, selectmode = EXTENDED,width = 100)

box.place(x=0, y=40)

#Создаём главное меню, задаём его размеры и выводим компоненты

if \_\_name\_\_ == "\_\_main\_\_":

root = tk.Tk()

main = Main(root)

main.pack()

root.title("Скорая помощь")

root.geometry("250x200")

root.mainloop()