

Практическая работа 2

Показатель безотказности системы

Для начала посмотрим код:

```
import flet as ft
import matplotlib
matplotlib.use('Agg')
import matplotlib.pyplot as plt
import os

def main(page: ft.Page):
    page.title = 'Показатели безотказности системы'
    page.horizontal_alignment = ft.CrossAxisAlignment.CENTER

    result1 = ft.Text("", size=18)
    result2 = ft.Text("", size=18)
    result3 = ft.Text("", size=18)
    history_text = ft.Text('История расчетов:', size=14, weight=ft.FontWeight.BOLD)
    history = []

    def add_history(text):
        history.append(text)
        history_text.value = 'История расчетов:\n' + '\n'.join(history)
        page.update()

    def open_chart_window(times, title='Время до отказа'):
        total_time = sum(times)
        num_failures = len(times)
        mtbf = total_time / num_failures

        fig, ax = plt.subplots()
        x = range(1, len(times) + 1)
        colors = ['green' if t >= mtbf else 'red' for t in times]
        ax.bar(x, times, color=colors, alpha=0.7, label='Время до отказа')
        ax.axhline(y=mtbf, color='blue', linestyle='--', linewidth=2, label=f'MTBF = {mtbf:.2f} ч')
        ax.set_xlabel('Номер отказа')
        ax.set_ylabel('Время (часы)')
        ax.set_title(title)
        ax.legend()
        ax.grid(True, alpha=0.3)
        plt.savefig('chart.png', dpi=100, bbox_inches='tight')
        plt.close()

    calc_text = ft.Text(
```

```

f'Расчет МТБФ:\n'
f'Сумма времени работы: {total_time} ч\n'
f'Количество отказов: {num_failures}\n'
f'МТБФ = {total_time} / {num_failures} = {mtbf:.2f} ч',
size=14
)

dlg = ft.AlertDialog(
    title=ft.Text('График'),
    content=ft.Column([
        ft.Image(src='chart.png', width=500, height=350),
        calc_text
    ])
)
page.open(dlg)

def calc_task1(e):
    times = [185, 342, 268, 220, 96, 102]
    t0 = sum(times) / len(times)
    result1.value = f'Задание 1: T0 = {t0:.2f} часов'
    add_history(f'Задание 1: T0 = {t0:.2f} часов')
    open_chart_window(times, 'Время до отказа и МТБФ')

def calc_task2(e):
    data = [
        (358, 4),
        (385, 3),
        (400, 2)
    ]

    rows = []
    times = []
    for i, (t, n) in enumerate(data):
        mtbf_i = t / n
        times.append(mtbf_i)
        rows.append(ft.DataRow(cells=[
            ft.DataCell(ft.Text(f'Система {i+1}')),
            ft.DataCell(ft.Text(f'{t}')),
            ft.DataCell(ft.Text(f'{n}')),
            ft.DataCell(ft.Text(f'{mtbf_i:.2f}'))
        ]))

    table = ft.DataTable(columns=[
        ft.DataColumn(ft.Text('Система')),
        ft.DataColumn(ft.Text('Время работы')),

```

```

ft.DataColumn(ft.Text('Кол-во отказов')),
ft.DataColumn(ft.Text('MTBF-системы'))
], rows=rows)

```

```

total_t = sum(d[0] for d in data)
total_n = sum(d[1] for d in data)
t0 = total_t / total_n
result2.value = f'Задание 2: T0 = {t0:.2f} часов'
add_history(f'Задание 2: T0 = {t0:.2f} часов')

```

```

dlg = ft.AlertDialog(
    title=ft.Text('Задание 2'),
    content=ft.Column([
        table,
        ft.Text(f'Общее время: {total_t} ч, Всего отказов: {total_n}'),
        ft.Text(f'MTBF = {total_t}/{total_n} = {t0:.2f} ч'),
        ft.Button('Показать график', on_click=lambda e:
open_chart_window(times, 'MTBF систем'))
    ])
)
page.open(dlg)

```

```

def calc_task3(e):
    t01, tv1 = 24, 16
    t02, tv2 = 400, 32

```

```

rows = [
    ft.DataRow(cells=[
        ft.DataCell(ft.Text('1')),
        ft.DataCell(ft.Text(f'{t01}')),
        ft.DataCell(ft.Text(f'{tv1}')),
        ft.DataCell(ft.Text(f'{t02}')),
        ft.DataCell(ft.Text(f'{tv2}'))
    ])
]

```

```

table = ft.DataTable(columns=[
    ft.DataColumn(ft.Text('Номер')),
    ft.DataColumn(ft.Text('t01, час')),
    ft.DataColumn(ft.Text('tb1, час')),
    ft.DataColumn(ft.Text('t02, час')),
    ft.DataColumn(ft.Text('tb2, час'))
], rows=rows)

```

```

k1 = t01 / (t01 + tv1)

```

```
k2 = t02 / (t02 + tv2)
better = 'Система 1' if k1 > k2 else 'Система 2'
```

```
calc_info = ft.Text(
    f'Расчет коэффициента готовности:\n'
    f'KГ1 = {t01} / ({t01} + {tv1}) = {k1:.4f}\n'
    f'KГ2 = {t02} / ({t02} + {tv2}) = {k2:.4f}\n'
    f'Лучше: {better}',
    size=14
)
```

```
result3.value = f'Задание 3: KГ1 = {k1:.4f}, KГ2 = {k2:.4f}. Лучше: {better}'
add_history(f'Задание 3: KГ1= {k1:.4f}, KГ2= {k2:.4f}, Лучше: {better}')
```

```
dlg = ft.AlertDialog(
    title=ft.Text('Задание 3'),
    content=ft.Column([
        table,
        calc_info
    ])
)
page.open(dlg)
```

```
page.add(
    ft.Text('Задание 1: Средняя наработка на отказ', size=20,
weight=ft.FontWeight.BOLD),
    ft.Text('6 отказов: 185, 342, 268, 220, 96, 102 часа'),
    ft.Button('Рассчитать', on_click=calc_task1),
    result1,
    ft.Divider(),
    ft.Text('Задание 2: Нарботка на отказ для нескольких систем', size=20,
weight=ft.FontWeight.BOLD),
    ft.Text('t1=358, n1=4; t2=385, n2=3; t3=400, n3=2'),
    ft.Button('Рассчитать', on_click=calc_task2),
    result2,
    ft.Divider(),
    ft.Text('Задание 3: Коэффициент готовности', size=20,
weight=ft.FontWeight.BOLD),
    ft.Text('Система 1: t0=24, tv=16; Система 2: t0=400, tv=32'),
    ft.Button('Рассчитать', on_click=calc_task3),
    result3,
    ft.Divider(),
    history_text
)
```

```
if __name__ == "__main__":  
    ft.app(target=main)
```

Теперь посмотрим на работу приложения на рисунке 1:

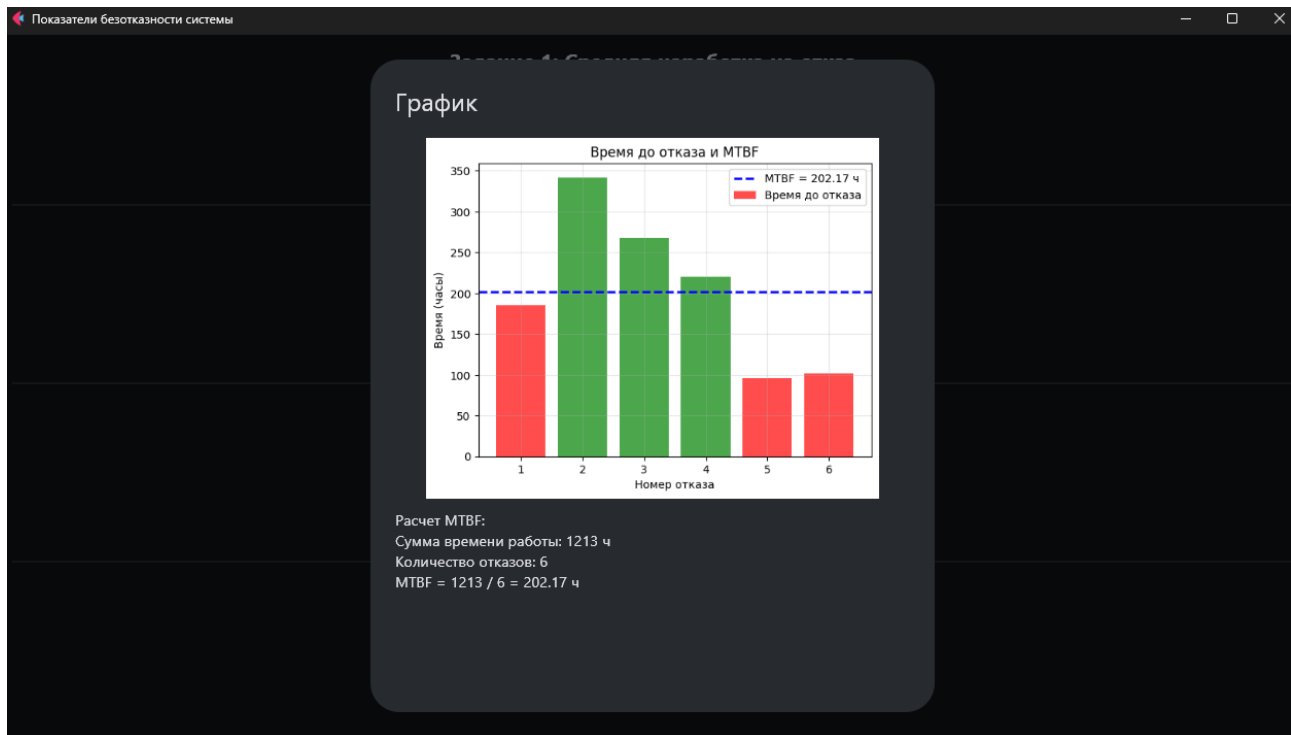


Рисунок 1 — Демонстрация работы