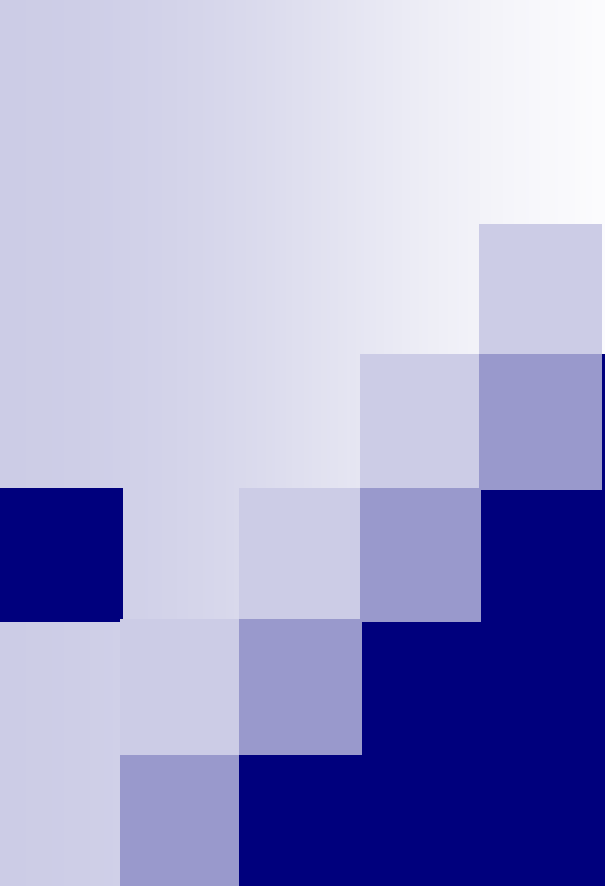




Технологии и опыт разработки интерактивных электронных учебников в процессе подготовки к занятиям

Очков Валерий Федорович,
проф., кафедра ТОТ,
Тихонов Антон Иванович,
проф. кафедра ФТЭМК,
НИУ «МЭИ», Москва

Подход к проведению занятий

- Сюжеты – постановки задач
- **Совместное решение задач**
- Нужна поддержка:
 - ☐ Формулировка задач
 - ☐ Решение задач
 - ☐ Интерпретация результатов
- Побочный результат – раздел интерактивного учебного пособия

Нужна поддержка:

- Форматированного текста
- Математических формул
- Изображений и видео
- **Интерактивных приложений**
- **Программной анимации**
- **Интеграции в интерактивное учебное пособие**



Основные требования

- Простота
- Приемлемая трудоемкость
- Поддержка **локальности**
- **Неотторжение** студентами

Реализация в экосистеме Python



marimo – не только украшение аквариума

- Вычислительный документ (аналог JN, JL)
- Реактивность
- Интеграция в Markdown
- Блокнот «чистый» Python
- Простые средства работы с данными (SQL, СУБД, фреймы данных, визуализация)
- Простой пользовательский интерфейс для интерактивных приложений
- **Простая интеграция блокнотов с помощью Flask, FastAPI**

Формулы с помощью latex и markdown

```
1 # Формулы с помощью `latex` и `markdown`
```

markdown

```
1 import marimo as mo
```

Разложение экспоненты $f(x) = e^x$ может представлено так:

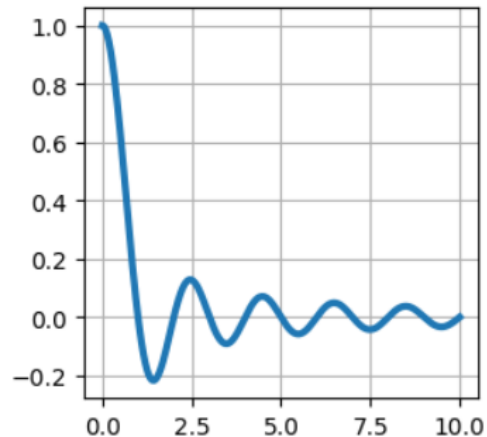
$$f(x) = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots$$

```
1  mo.md(  
2      r'''  
3      Разложение экспоненты  $f(x) = e^x$  может представлено так:  
4  
5      \[  
6      f(x) = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \ldots.  
7      \]  
8      '''  
9  )  
10
```

Блокнот marimo – встраивание в Markdown

```
1 import marimo as mo
2 import numpy as np
3 import matplotlib.pyplot as plt
```

Выводим график matplotlib с помощью Markdown



```
1 fig = plt.figure(figsize=(3,3))
2 t = np.linspace(0,10, 1000)
3 plt.plot(t, np.sinc(t), lw=3)
4 plt.grid()
5 mo.md(
6     f"""
7     ### Выводим график matplotlib с помощью Markdown
8
9     {mo.as_html(fig)}
10    """
11 )
```


Видео

20000000

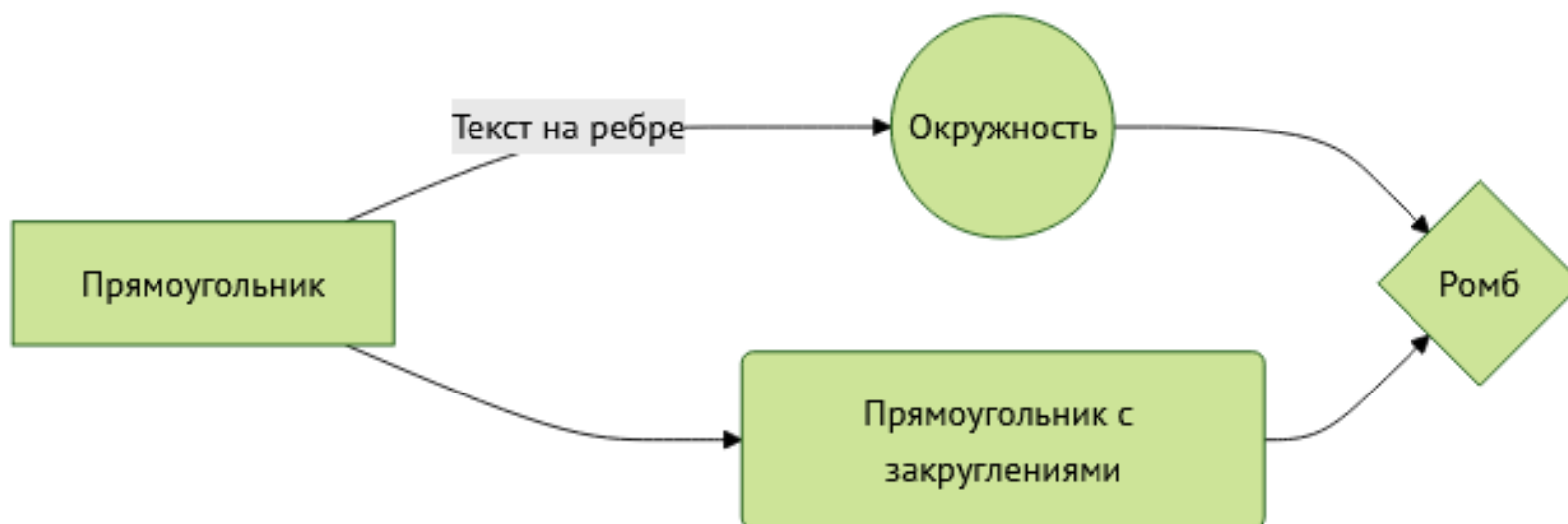
```
1 os.environ.get("MARIMO_OUTPUT_MAX_BYTES")
```



```
1 v with open("attr.mp4", mode="rb") as f:  
2     fileContent = f.read()  
3 v     v = mo.video(src=fileContent, controls=True,  
4         autoplay=True, width="100%")  
5 v
```

175

Диаграммы Mermaid



```
1 diagram = '''
2 graph LR
3     A[Прямоугольник] -- Текст на ребре --> B((Окружность))
4     A --> C[Прямоугольник с закруглениями]
5     B --> D{Ромб}
6     C --> D
7     ...
8 mo.mermaid(diagram)
```

Диаграмма Ганта



```
1 gantt = ''
2 gantt
3     section Раздел
4     Завершено :done,    des1, 2025-01-06,2025-01-08
5     Выполняется :active, des2, 2025-01-07, 3d
6     Параллельно 1 :      des3, after des1, 1d
7     Параллельно 2 :      des4, after des1, 1d
8     Параллельно 3 :      des5, after des3, 1d
9     Параллельно 4 :      des6, after des4, 1d
10    ...
11 mo.mermaid(gantt)
```

Работа с данными

Make
str
unique: 38
nulls: 0

Model
str
unique: 425
nulls: 0

Cylinders
f64

3 12

Weight
f64

1 850 7 190

MPG_City
f64

10

Acura	MDX	6	4451	17
Acura	RSX Type S 2dr	4	2778	24
Acura	TSX 4dr	4	3230	22
Acura	TL 4dr	6	3575	20
Acura	3.5 RL 4dr	6	3880	18

428 rows, 5 columns

Page 1 of 86

Download

```
1 -- Загрузка и создание таблицы в оперативной памяти
2 v CREATE OR REPLACE TABLE cars as
3   FROM 'https://datasets.marimo.app/cars.csv';
4   SELECT Make, Model, Cylinders, Weight, MPG_City from cars;
5
```

Output variable:

☐ Hide output | sql 

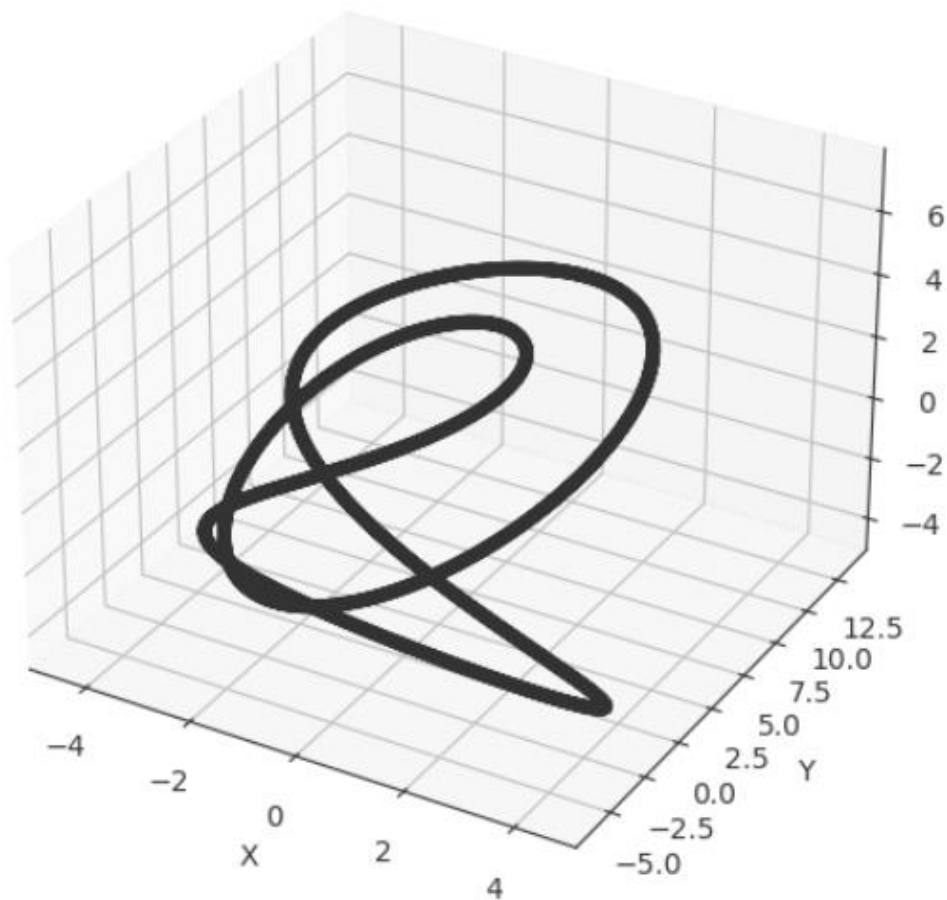
Вращаем трехмерную кривую вокруг осей

Ось Z: -60°

Ось Y: 0°

Ось X: 30°

☒ Оси координат



Рисуем кривые и поверхности в 3d

Мёбиус

Выбор фигуры

Мёбиус

Поворот вокруг оси

z

y

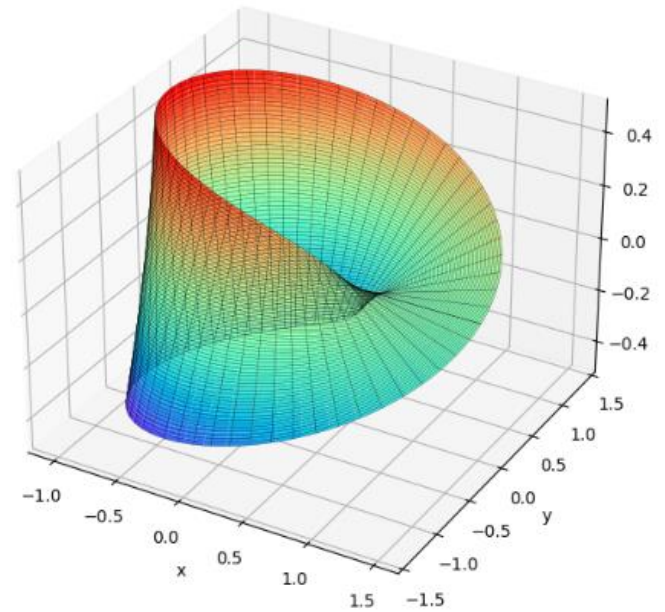
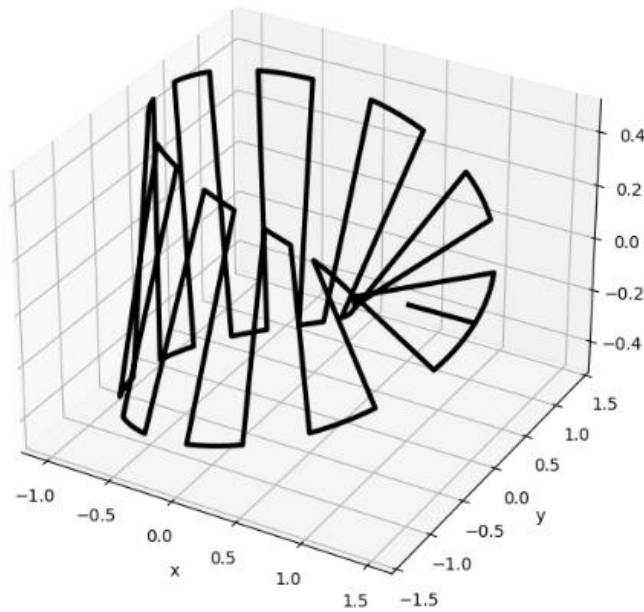
x

Цвета, оцифровка, сетка

палитра: rainbow

оцифровка

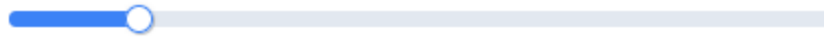
шаг сетки



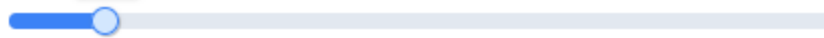
Строим гистограммы

Данные:

Объем выборки:



Число рядов гистограммы:



Распределение:

- ☐ Нормальное
- ☐ Равномерное
- ☐ Экспоненциальное
- ☒ Бимодальное

Рисунок

Ширина рисунка:



Высота рисунка:



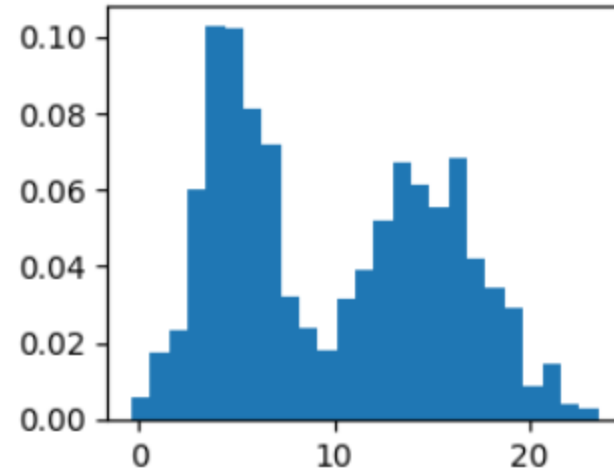
Характеристики распределения:

Среднее: 9.990

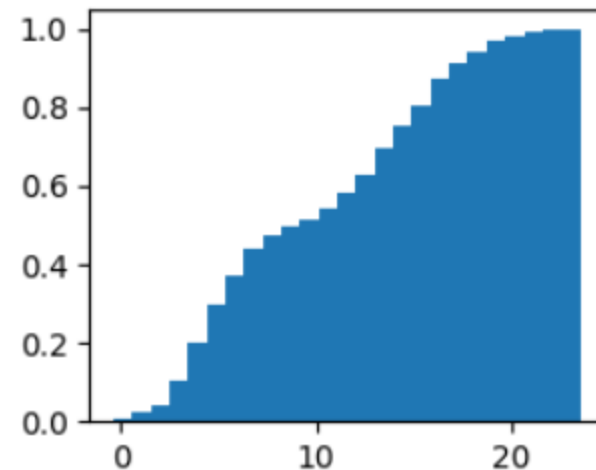
Медиана: 9.417

СКО: 5.590

Плотность распределения



Кумулятивная функция распределения



Лабораторная работа

Диэлектрическая спектроскопия

Описание

Лабораторный стенд

Задание

Подбор параметров релаксаторов

Выбор материала:

Материал 1

Число релаксаторов: ☐ 1 ☐ 2 ☒ 3 ☐ 4

☒ Оптимизация

▼ [3 Items

0: a1: 1

1: a2: 1

2: a3: 1

1

▼ [3 Items

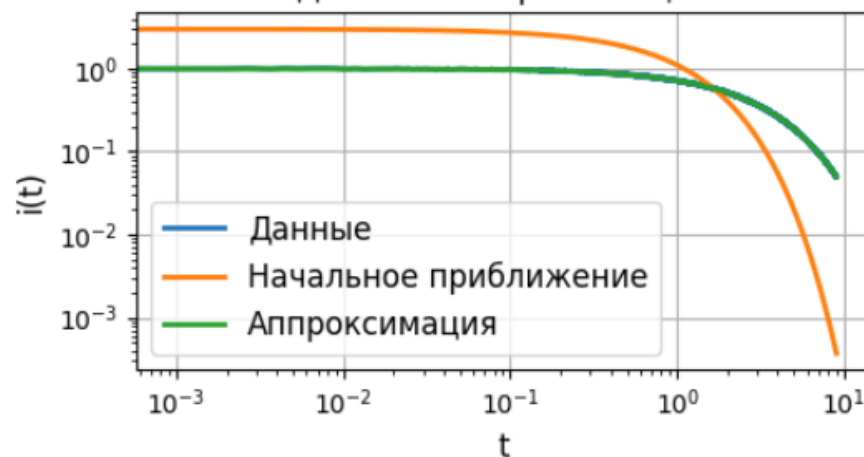
0: t1: 1

1: t2: 1

2: t3: 1

1

Данные и аппроксимация



Погрешности: max_error: 2.576e-02, std_error: 4.117e-03

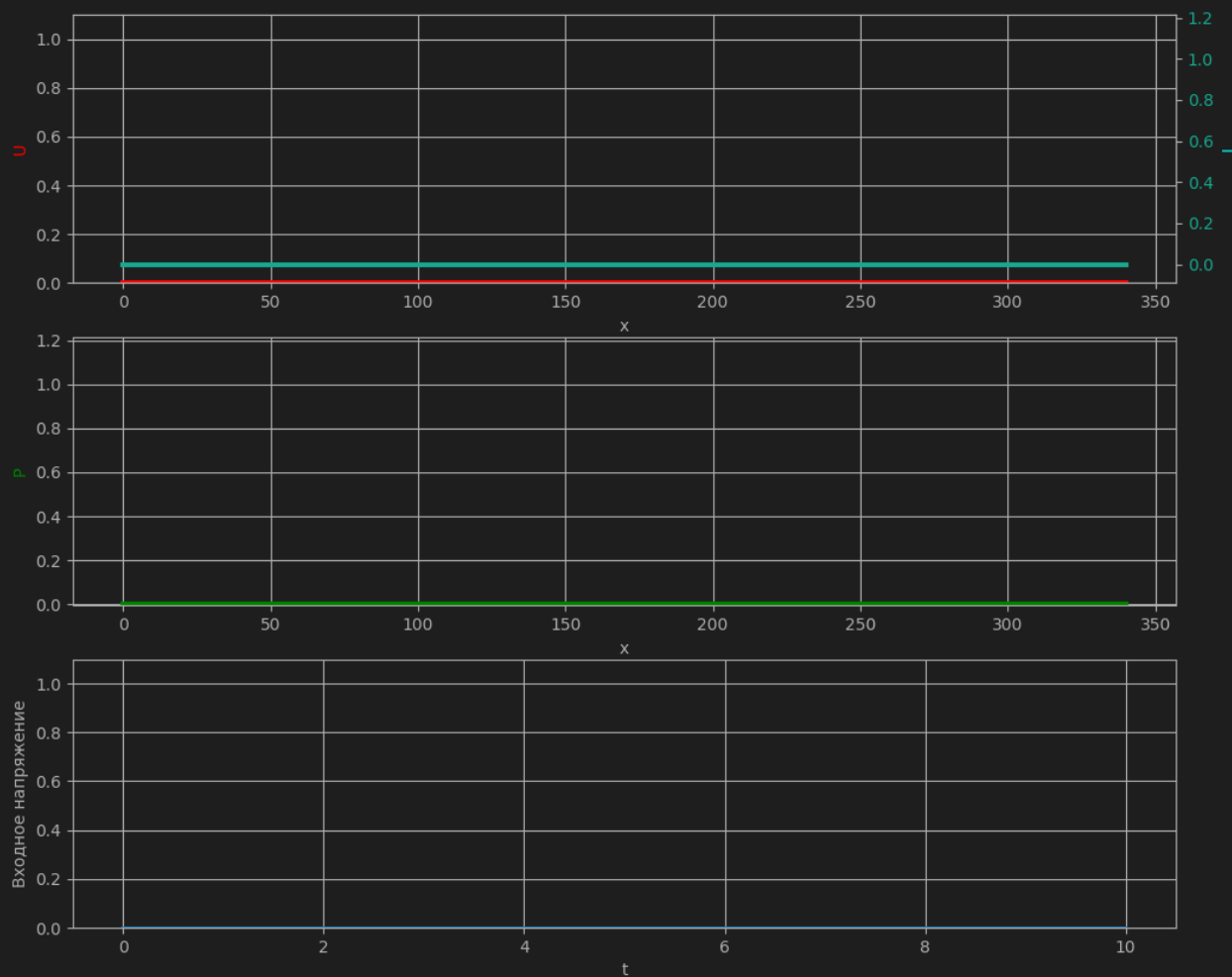
Параметры релаксаторов

☐	a	t
☐	1.713e-01	6.866e-02
☐	-1.734e-01	6.763e-02
☐	1.000e+00	2.999e+00

🔍 📄 📄

Download ▼

Распространение импульсов в витой паре



Интеграция блокнотов в электронный учебник

Запускаем блокноты marimo из веб-приложения FastAPI

Доступные блокноты marimo:

1. Пример 01

Первый блокнот marimo

Щелкните мышью, чтобы
открыть приложение

2. Пример 02

Ячейка Python блокнота
marimo

Щелкните мышью, чтобы
открыть приложение

3. Пример 03

Ячейка Markdown
блокнота marimo

Щелкните мышью, чтобы
открыть приложение

4. Пример 04

Ячейка SQL блокнота
marimoa

Щелкните мышью, чтобы
открыть приложение

5. Пример 05

Замечания в Markdown

Щелкните мышью, чтобы
открыть приложение

6. Пример 06

Исходный текст в
Markdown

Щелкните мышью, чтобы
открыть приложение

7. Пример 07

Мультимедиа в Markdown

Щелкните мышью, чтобы
открыть приложение

8. Пример 08

Диаграммы

Щелкните мышью, чтобы
открыть приложение

9. Пример 09

Визуализация matplotlib

Щелкните мышью, чтобы
открыть приложение

10. Пример 10

Визуализация plotly

Щелкните мышью, чтобы
открыть приложение

11. Пример 11

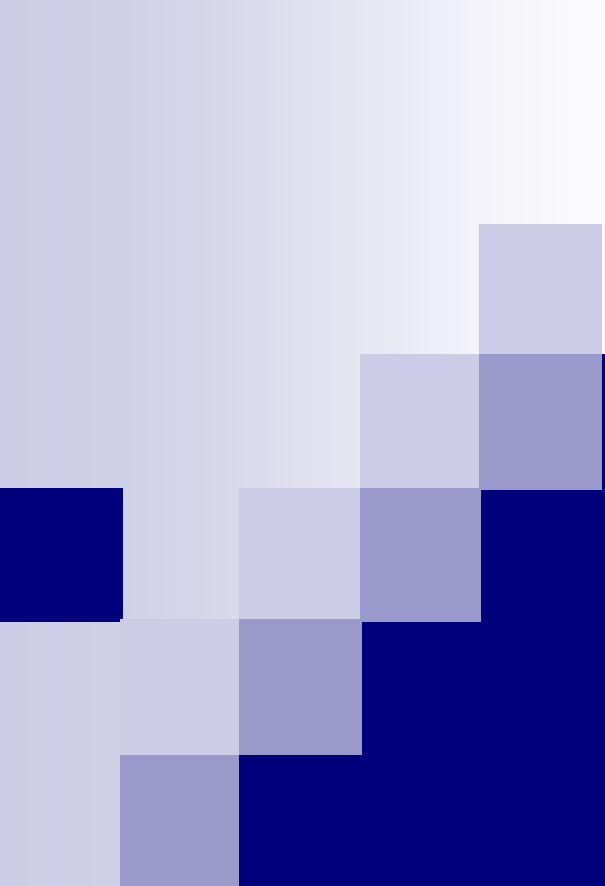
Реактивные переменные

Щелкните мышью, чтобы
открыть приложение

12. Пример 12

Пользовательский
интерфейс

Щелкните мышью, чтобы
открыть приложение



Спасибо за внимание!