

# Программирование графики

Учебная практика

мехмат, III курс, кафедра ИВЭ

Введение

Пример

Проблемы

Обзор

picture

MetaPost 1

MetaPost 2

PSTricks

PGF/TikZ

pgffor

Примеры

Параметры 1

Параметры 2

Пример 1

Пример 2

Этапы

Пример 1

Команды 1

Команды 2

Пути

Вершины

Области

# Введение

# Пример растрового изображения

Введение

**Пример**

Проблемы

Обзор

picture

MetaPost 1

MetaPost 2

PSTricks

PGF/TikZ

pgffor

Примеры

Параметры 1

Параметры 2

Пример 1

Пример 2

Этапы

Пример 1

Команды 1

Команды 2

Пути

Вершины

Области

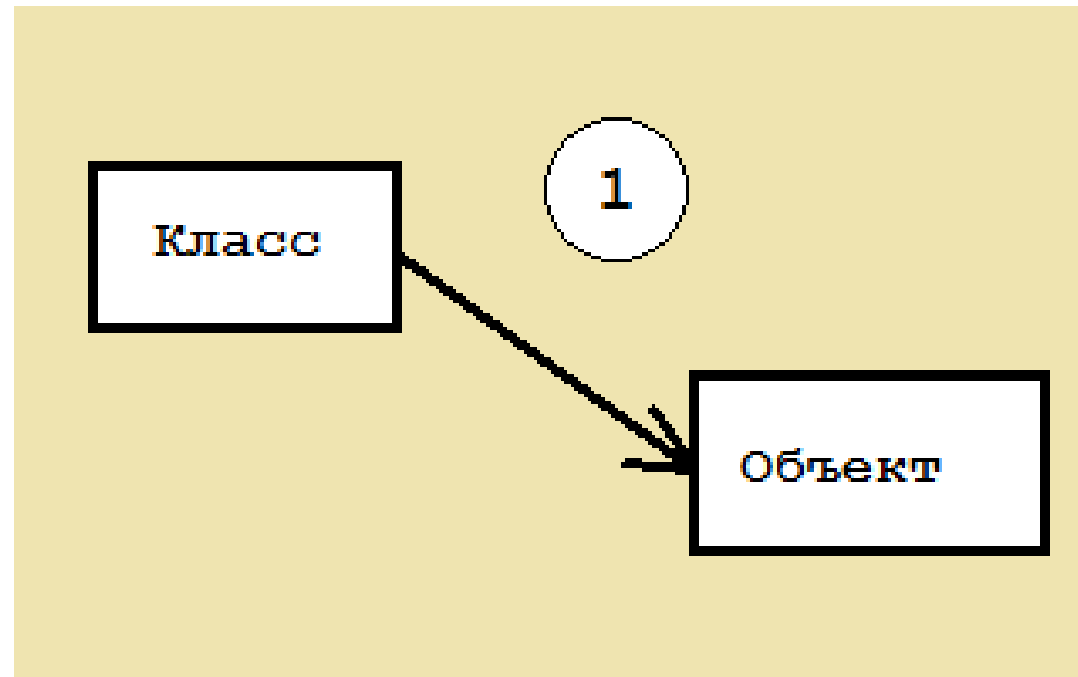


Рис. 1: пример диаграммы в формате растрового изображения

# Проблемы растровых изображений

Введение

Пример

**Проблемы**

Обзор

picture

MetaPost 1

MetaPost 2

PSTricks

PGF/TikZ

pgffor

Примеры

Параметры 1

Параметры 2

Пример 1

Пример 2

Этапы

Пример 1

Команды 1

Команды 2

Пути

Вершины

Области

- ✓ Разрешение лазерных принтеров (тем более, профессиональных устройств печати) гораздо выше современных экранов.
- ✓ Требованием многих издательств при приёме материалов к печати является представление всех сложных рисунков в векторном формате.
- ✓ Растровый способ хранения изображений хорошо подходит для представления фотографий, снимков экранов и т. д., но совершенно не подходит для представления рисованных диаграмм ввиду плохого поведения при масштабировании и выводе на печать (пикселизация, размытие контуров).
- ✓ Графические визуальные редакторы доставляют сложности при необходимости создания сложных диаграмм и редактирования уже существующих.

# Обзор основных графических средств $\LaTeX$

Введение

Пример

Проблемы

**Обзор**

picture

MetaPost 1

MetaPost 2

PSTricks

PGF/TikZ

pgffor

Примеры

Параметры 1

Параметры 2

Пример 1

Пример 2

Этапы

Пример 1

Команды 1

Команды 2

Пути

Вершины

Области

В качестве альтернативного подхода выступает описание диаграмм на некотором языке программирования.

- ✓ Стандартные средства  $\LaTeX$  (окружение `picture`).
- ✓ Внешние инструменты (METAPOST).
- ✓ Набор команд PSTricks.
- ✓ Пакет PGF/TikZ .
- ✓ ...

# Стандартное окружение `picture`

- Введение
- Пример
- Проблемы
- Обзор
- picture**
- MetaPost 1
- MetaPost 2
- PSTricks
- PGF/TikZ
- pgffor
- Примеры
- Параметры 1
- Параметры 2
- Пример 1
- Пример 2
- Этапы
- Пример 1
- Команды 1
- Команды 2
- Пути
- Вершины
- Области

- ✓ Использует встроенные средства  $\TeX$  для построения изображений, портируемо.
- ✓ Результирующий документ имеет малый объём.
- ✓ Позволяет использовать те же шрифты в диаграммах, что и в остальном тексте.
- ✗ Имеет крайне ограниченные возможности (ограниченный набор радиусов окружностей и т. д.)
- ✗ Средства автоматизации очень ограничены (команда `\multiput` и т. д.)

# Инструмент METAPOST

Введение

Пример

Проблемы

Обзор

picture

**MetaPost 1**

MetaPost 2

PSTricks

PGF/TikZ

pgffor

Примеры

Параметры 1

Параметры 2

Пример 1

Пример 2

Этапы

Пример 1

Команды 1

Команды 2

Пути

Вершины

Области

Предоставляет возможности описания на сложном языке программирования нескольких векторных рисунков в одном файле исходного кода. После компиляции рисунки можно включать в документ  $\text{\LaTeX}$  при помощи команды `\includegraphics`.

- ✓ Предоставляет сложный язык программирования с богатыми алгоритмическими возможностями.
- ✓ Существуют библиотеки для решения типичных задач (рисование графиков, трёхмерные изображения и т. д.)

# Инструмент METAPOST (окончание)

Введение

Пример

Проблемы

Обзор

picture

MetaPost 1

**MetaPost 2**

PSTricks

PGF/TikZ

pgffor

Примеры

Параметры 1

Параметры 2

Пример 1

Пример 2

Этапы

Пример 1

Команды 1

Команды 2

Пути

Вершины

Области

- ✘ Имеет недостаточно высокий уровень для удобства использования.
- ✘ Усложняет процесс компиляции документа (использование дополнительных инструментов).
- ✘ Использование шрифтов, одинаковых с остальным текстом документа, возможно, хотя и связано со сложностями (два прохода инструмента и т. д.)
- ✘ В настоящее время дальнейшее развитие ведётся очень медленно.



# Набор PSTricks

Введение

Пример

Проблемы

Обзор

picture

MetaPost 1

MetaPost 2

**PSTricks**

PGF/TikZ

pgffor

Примеры

Параметры 1

Параметры 2

Пример 1

Пример 2

Этапы

Пример 1

Команды 1

Команды 2

Пути

Вершины

Области

Представляет собой набор макросов  $\TeX$ , реализующих графические возможности средствами языка PostScript. Реализуется при помощи команд `\special`, переносящих команды PostScript через DVI-файл.

- ✓ Имеет богатые возможности для представления графики.
- ✓ Существует множество библиотек для решения задач из областей математики, информатики, физики и т. д.
- ✓ Не требует внешних инструментов.
- ✓ Описание графики выполняется в том же файле, что и описание документа.
- ✓ Не имеет проблем с использованием тех же шрифтов, что и в остальном документе.
- ✗ Требует компиляции в формат PS, следовательно, несовместим с механизмом pdf $\TeX$ .

# Пакет PGF/TikZ

Введение

Пример

Проблемы

Обзор

picture

MetaPost 1

MetaPost 2

PSTricks

**PGF/TikZ**

pgffor

Примеры

Параметры 1

Параметры 2

Пример 1

Пример 2

Этапы

Пример 1

Команды 1

Команды 2

Пути

Вершины

Области

Как и PSTricks, полностью реализован средствами  $\TeX$ . Представляет собой набор из трёх уровней абстракции:

**Системный уровень:** реализует минимальный набор примитивных операций для следующего слоя. Абстрагирует различия в разных драйверах (dvips, pdf $\TeX$  и т. д.) Набор реализуемых операций минимален для упрощения портирования на другие драйверы.

**Базовый уровень, PGF:** реализует более сложные операции на основе примитивов системного уровня. Например, рисование окружности одной командой, реализованное при помощи кривых Безье системного уровня.

**Пользовательский уровень, TikZ:** работает с понятиями более высокого уровня абстракции: вершины, фигуры, сложные системы координат.

# Пакет pgffor

Введение

Пример

Проблемы

Обзор

picture

MetaPost 1

MetaPost 2

PSTricks

PGF/TikZ

**pgffor**

Примеры

Параметры 1

Параметры 2

Пример 1

Пример 2

Этапы

Пример 1

Команды 1

Команды 2

Пути

Вершины

Области

- ✓ Один из пакетов  $\text{\LaTeX}$ , предоставляющих возможности организации циклов.
- ✓ Является частью набора утилит пакета PGF/TikZ, однако может использоваться также отдельно от него.

**`\foreach`** *⟨переменные⟩* [*⟨параметры⟩*] in *⟨список⟩* *⟨код⟩*

Повторяет *⟨код⟩*, внутри которого могут содержаться переменные в форме команд  $\text{\TeX}$ . *⟨переменные⟩* разделяются символами «/», *⟨список⟩* состоит из последовательности групп значений, разделённых запятыми, каждая группа разделена символами «/». Вместо одной из групп может встречаться многоточие.

Пример

```
\foreach \i / \k in {1/a, 2/b, 3/c} {(\i: \k) }
```

# Примеры использования команды \foreach

## Пример

```
\foreach \i in {1, ..., 4} {\i--} \\
\foreach \i in {4, ..., 1} {\i\ $\rightarrow$ } \\
\foreach \i in {a, ..., d} {\i, } \\
\foreach \i in {1, 1.5, ..., 5.8} {-\i-} \\ [1ex]
$\foreach \i / \k in {1/a, 2/b, 3/c} {\dfrac {\i} {\k}}$
```

1-2-3-4-

4 → 3 → 2 → 1 →

a, b, c, d,

-1--1.5--2--2.5--3--3.5--4--4.5--5--5.5-

$$\frac{1}{a} \frac{2}{b} \frac{3}{c}$$

Введение

Пример

Проблемы

Обзор

picture

MetaPost 1

MetaPost 2

PSTricks

PGF/TikZ

pgffor

**Примеры**

Параметры 1

Параметры 2

Пример 1

Пример 2

Этапы

Пример 1

Команды 1

Команды 2

Пути

Вершины

Области

# Основные параметры команды `\foreach`

Введение

Пример

Проблемы

Обзор

picture

MetaPost 1

MetaPost 2

PSTricks

PGF/TikZ

pgffor

Примеры

**Параметры 1**

Параметры 2

Пример 1

Пример 2

Этапы

Пример 1

Команды 1

Команды 2

Пути

Вершины

Области

**remember** =  $\langle \text{переменная} \rangle$  as  $\langle \text{команда} \rangle$  ( $\langle \text{initially} \langle \text{значение} \rangle \rangle$ )

Позволяет запоминать предыдущее значение переменной в другом макросе с заданным начальным значением (по умолчанию 0).

## Пример

```
\foreach \i %  
  [remember = \i as \j (initially 1)] %  
  in {2, 3, 4} %  
  {(\i, \j) }
```

(2, 1) (3, 2) (4, 3)

# Основные параметры команды `\foreach` (окончание)

Введение

Пример

Проблемы

Обзор

picture

MetaPost 1

MetaPost 2

PSTricks

PGF/TikZ

pgffor

Примеры

Параметры 1

**Параметры 2**

Пример 1

Пример 2

Этапы

Пример 1

Команды 1

Команды 2

Пути

Вершины

Области

**count** = `<команда> from <значение>`

Определяет новую переменную-счётчик, увеличивающуюся на каждой итерации, начиная с заданного значения (по умолчанию 1).

**evaluate** = `<переменная> as <команда> using <выражение>`

Позволяет вычислять новые переменные (`<команда>`) из значений уже существующих (`<переменная>`). По умолчанию если `<команда>` не указана, результат записывается в исходную переменную. По умолчанию если `<выражение>` не указано, оно берётся из значения самой переменной.

# Пример использования ключа evaluate

## Пример

```
\foreach \i [evaluate = \i as \result] %  
  in {1 + 2, 10 * 3, 2^3}%  
  {$\i = \result \quad}$%
```

---

$1 + 2 = 3.0$     $10 * 3 = 30.0$     $2^3 = 8.0$

Введение

Пример

Проблемы

Обзор

picture

MetaPost 1

MetaPost 2

PSTricks

PGF/TikZ

pgffor

Примеры

Параметры 1

Параметры 2

**Пример 1**

Пример 2

Этапы

Пример 1

Команды 1

Команды 2

Пути

Вершины

Области

# Пример использования ключей `count` и `evaluate`

## Пример

```
\foreach \letter %  
[ %  
  count = \i, %  
  evaluate = \i as \wave using 363 + (451 * \i) / 16  
] %  
  in {H, e, l, l, o, \ , f, u, n, n, y, \ , %  
    w, o, r, l, d}%  
{%  
  \textcolor [wave] {\wave} {\letter}%  
}%
```

Hello funny world

[Введение](#)

[Пример](#)

[Проблемы](#)

[Обзор](#)

[picture](#)

[MetaPost 1](#)

[MetaPost 2](#)

[PSTricks](#)

[PGF/TikZ](#)

[pgffor](#)

[Примеры](#)

[Параметры 1](#)

[Параметры 2](#)

[Пример 1](#)

**[Пример 2](#)**

[Этапы](#)

[Пример 1](#)

[Команды 1](#)

[Команды 2](#)

[Пути](#)

[Вершины](#)

[Области](#)



# Основные этапы работы с пакетом TikZ

Введение

Пример

Проблемы

Обзор

picture

MetaPost 1

MetaPost 2

PSTricks

PGF/TikZ

pgffor

Примеры

Параметры 1

Параметры 2

Пример 1

Пример 2

**Этапы**

Пример 1

Команды 1

Команды 2

Пути

Вершины

Области

1. Подключение пакета. В большинстве случаев достаточно команды:

```
\usepackage {tikz}
```

2. Необязательно: загрузка в преамбуле вспомогательных библиотек TikZ при помощи команды `\usetikzlibrary`.
3. Необязательно: установка в любом месте глобальных настроек при помощи команды `\tikzset`.
4. Описание изображений в тексте при помощи команды `\tikz` или в отдельном абзаце в окружении `tikzpicture`, которое, как правило, помещается внутри окружения `figure`.

# Пример документа с командой \tikz

Введение

Пример

Проблемы

Обзор

picture

MetaPost 1

MetaPost 2

PSTricks

PGF/TikZ

pgffor

Примеры

Параметры 1

Параметры 2

Пример 1

Пример 2

Этапы

**Пример 1**

Команды 1

Команды 2

Пути

Вершины

Области

```
\documentclass {article}

\usepackage {tikz}

\begin {document}
  A picture on a paragraph:
  \tikz \fill [magenta] (0, 0) circle [radius = 1ex];

  \begin {figure}
    \centering
    \begin {tikzpicture}
      \draw [thick, green!40!black] %
        (0, 0) -- +(0:1cm) -- +(60:1cm) -- cycle;
    \end {tikzpicture}
    \caption {a picture in a figure}
  \end {figure}
\end {document}
```

# Основные команды пакета

Введение

Пример

Проблемы

Обзор

picture

MetaPost 1

MetaPost 2

PSTricks

PGF/TikZ

pgffor

Примеры

Параметры 1

Параметры 2

Пример 1

Пример 2

Этапы

Пример 1

**Команды 1**

Команды 2

Пути

Вершины

Области

**\usetikzlibrary** {⟨список\_библиотек⟩}

Загружает заданные библиотеки, список разделяется запятыми.

Пример

```
\usetikzlibrary {arrows.meta, calc}
```

**\tikzset** {⟨параметры⟩}

Устанавливает заданные параметры глобально.

Пример

```
\tikzset {every picture/.style = thick}
```

# Основные команды пакета (окончание)

Введение

Пример

Проблемы

Обзор

picture

MetaPost 1

MetaPost 2

PSTricks

PGF/TikZ

pgffor

Примеры

Параметры 1

Параметры 2

Пример 1

Пример 2

Этапы

Пример 1

Команды 1

**Команды 2**

Пути

Вершины

Области

**\tikz** [*⟨параметры⟩*] *⟨команды\_путей⟩*

Добавляет внутри текста рисунок. Если команд несколько, они должны находиться в фигурных скобках.

**\begin{tikzpicture}** [*⟨параметры⟩*]  
*⟨содержимое\_окружения⟩*  
**\end{tikzpicture}**

Все остальные команды TikZ должны находиться внутри этого окружения. Исполняет заданные команды, вычисляет границы изображения (могут быть переопределены) и выводит его в прямоугольнике соответствующих размеров.

*Замечание:* все команды рисования должны заканчиваться символом «;».



Введение

**Пути**

Координаты

Относительные 1

Относительные 2

Пути

Операции

Основные 1

Основные 2

rectangle

circle

arc

grid

to

foreach

node

Действия 1

Действия 2

Действия 3

Действия 4

Вершины

Области

# Координаты и пути

# Задание координат

Введение

Пути

**Координаты**

Относительные 1

Относительные 2

Пути

Операции

Основные 1

Основные 2

rectangle

circle

arc

grid

to

foreach

node

Действия 1

Действия 2

Действия 3

Действия 4

Вершины

Области

**Вершина:** специальный элемент изображения, помещаемый в заданных координатах и, возможно, имеющий различные свойства: имя, фигуру, размеры, внутренний текст, цвет фона и т. д.

## Системы координат

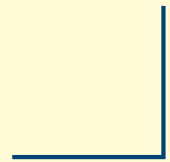
- ✓ Декартовы координаты:  $(1, 1.5)$ . Размерность по умолчанию: см, координата  $x$  направлена вправо,  $y$  — вверх.
- ✓ Трёхмерные декартовы координаты:  $(10\text{mm}, 14\text{mm}, 2\text{ex})$ .
- ✓ Полярные координаты:  $(60:2\text{cm})$ . Угол задаётся в градусах.
- ✓ Вершинные координаты — задаются относительно некоторой точки вершины.
- ✓ ...

# Относительные координаты

- ✓ Если перед координатой стоят символы «++», она определяется заданным смещением относительно предыдущей координаты. Следующая относительная координата будет вычисляться относительно текущей.

## Пример

```
\tikz %  
  \draw (0, 0) -- ++(1cm, 0) -- ++(0, 1cm);
```



Введение

Пути

Координаты

**Относительные 1**

Относительные 2

Пути

Операции

Основные 1

Основные 2

rectangle

circle

arc

grid

to

foreach

node

Действия 1

Действия 2

Действия 3

Действия 4

Вершины

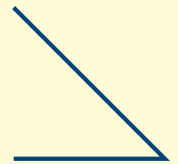
Области

# Относительные координаты (окончание)

- ✓ Символ «+» действует аналогично, но следующая относительная координата будет вычисляться относительно предыдущей, а не текущей координаты.

## Пример

```
\tikz %  
  \draw (0, 0) -- +(1cm, 0) -- ++(0, 1cm);
```



Введение

Пути

Координаты

Относительные 1

**Относительные 2**

Пути

Операции

Основные 1

Основные 2

rectangle

circle

arc

grid

to

foreach

node

Действия 1

Действия 2

Действия 3

Действия 4

Вершины

Области



# Задание путей

Введение

Пути

Координаты

Относительные 1

Относительные 2

**Пути**

Операции

Основные 1

Основные 2

rectangle

circle

arc

grid

to

foreach

node

Действия 1

Действия 2

Действия 3

Действия 4

Вершины

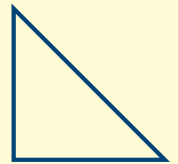
Области

**Путь:** последовательность координат, соединённых линиями.

В простейшем случае синтаксис определения путей в TikZ аналогичен METAPOST.

Пример

```
\tikz %  
  \draw (0, 0) -- (0, 1) -- (1, 0) -- cycle;
```



# Операции путей

Введение

Пути

Координаты

Относительные 1

Относительные 2

Пути

**Операции**

Основные 1

Основные 2

rectangle

circle

arc

grid

to

foreach

node

Действия 1

Действия 2

Действия 3

Действия 4

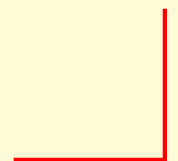
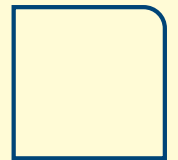
Вершины

Области

Путь представляет собой последовательность операций, между которыми могут находиться настройки локальных параметров (в квадратных скобках) и фигурные скобки, задающие локальные группы, в пределах которых действуют настройки. Некоторые настройки (`color` и т. д.) действуют на весь путь.

## Пример

```
\tikz %
  \draw %
    (0, 0) -- (1, 0)
    { [rounded corners] -- (1, 1) } %
    -- (0, 1) -- cycle; \ \ [1em]
\tikz %
  \draw %
    (0, 0) -- (1, 0) %
    [color = red] -- ++(90:1);
```



# Основные операции путей

Введение

Пути

Координаты

Относительные 1

Относительные 2

Пути

Операции

**Основные 1**

Основные 2

rectangle

circle

arc

grid

to

foreach

node

Действия 1

Действия 2

Действия 3

Действия 4

Вершины

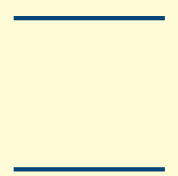
Области

**Перемещение:** («move-to») — перемещает текущую координату в заданную позицию. При помощи неё обычно задаются начала путей, также можно определять несвязные пути. Обозначается координатой.

**Отрезок:** («line-to») — определяет отрезок прямой от текущей координаты до заданной. Обозначается « -- *⟨координата⟩* ».

## Пример

```
\tikz %  
  \draw [thick] %  
    (0, 0) -- (1, 0) %  
    (0, 1) -- (1, 1);
```



# Основные операции путей (окончание)

Введение

Пути

Координаты

Относительные 1

Относительные 2

Пути

Операции

Основные 1

**Основные 2**

rectangle

circle

arc

grid

to

foreach

node

Действия 1

Действия 2

Действия 3

Действия 4

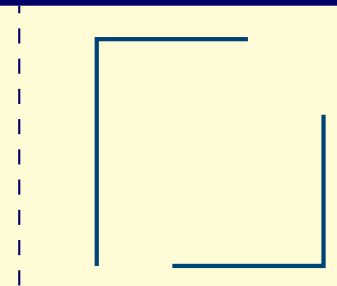
Вершины

Области

**Пара отрезков:** определяет два отрезка горизонтальной и вертикальной прямых (или наоборот). Обозначаются « -| или |- *⟨координата⟩* ».

Пример

```
\tikz %  
  \draw %  
    (0, 0) -| (1, 1) %  
    (-0.5, 0) |- (0.5, 1.5);
```



**Локальная группа:** («score») — уже рассматривалась, ограничивает часть операций пути фигурными скобками.

# Операция `rectangle`

Введение

Пути

Координаты

Относительные 1

Относительные 2

Пути

Операции

Основные 1

Основные 2

**rectangle**

circle

arc

grid

to

foreach

node

Действия 1

Действия 2

Действия 3

Действия 4

Вершины

Области

**rectangle** *⟨противоположный\_угол⟩*

Определяет прямоугольник с противоположными углами в текущей координате и заданной.

Пример

```
\tikz %  
  \draw %  
    (0, 0) rectangle +(1.5, 0.5) %  
    (0, 1) rectangle +(1.5, 0.5);
```



# Операция circle

Введение

Пути

Координаты

Относительные 1

Относительные 2

Пути

Операции

Основные 1

Основные 2

rectangle

**circle**

arc

grid

to

foreach

node

Действия 1

Действия 2

Действия 3

Действия 4

Вершины

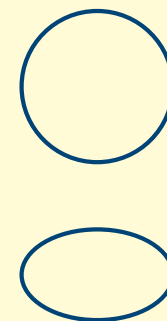
Области

**circle** [*⟨параметры⟩*]

Определяет эллипс с центром в текущей координате и заданным радиусом (радиусами).

## Пример

```
\tikz %
  \draw %
    (0, 0) circle [radius = 5mm]; \\\ [1em]
\tikz %
  \draw %
    (0, 0) circle %
    [ %
      x radius = 5mm, %
      y radius = 3mm %
    ];
```



# Операция arc

Введение

Пути

Координаты

Относительные 1

Относительные 2

Пути

Операции

Основные 1

Основные 2

rectangle

circle

**arc**

grid

to

foreach

node

Действия 1

Действия 2

Действия 3

Действия 4

Вершины

Области

**arc** [*⟨параметры⟩*]

Аналогично, определяет сектор эллипса между заданными углами.

## Пример

```
\tikz %  
  \draw %  
    (0, 0) %  
    arc %  
    [ %  
      radius = 0.5cm, %  
      start angle = 0, %  
      end angle = 120 %  
    ];
```



# Операция grid

Введение

Пути

Координаты

Относительные 1

Относительные 2

Пути

Операции

Основные 1

Основные 2

rectangle

circle

arc

**grid**

to

foreach

node

Действия 1

Действия 2

Действия 3

Действия 4

Вершины

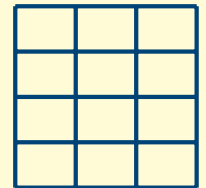
Области

**grid** [*⟨параметры⟩*] *⟨противоположный\_угол⟩*

Аналогично rectangle, определяет прямоугольную сетку.

## Пример

```
\tikz %  
  \draw %  
    (0, 0) %  
    grid [xstep = 4mm, ystep = 3mm] %  
    (1.2, 1.2);
```





# Операция to

Введение

Пути

Координаты

Относительные 1

Относительные 2

Пути

Операции

Основные 1

Основные 2

rectangle

circle

arc

grid

**to**

foreach

node

Действия 1

Действия 2

Действия 3

Действия 4

Вершины

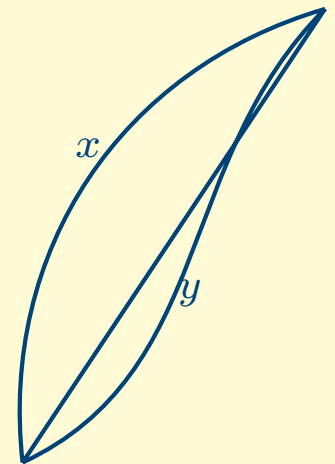
Области

**to** [*⟨параметры⟩*] *⟨вершины⟩* *⟨координата⟩*

Определяет прямую или дугу от текущей координаты до заданной, вдоль которой могут располагаться *⟨вершины⟩*.

## Пример

```
\tikz %  
  \draw %  
    (0, 0) %  
    to (2, 3) %  
    (0, 0) %  
    to [bend left = 40] %  
      node [above] {$x$} (2, 3) %  
    (0, 0) %  
    to [out = 25, in = -135] %  
      node [below] {$y$} (2, 3);
```



# Операция foreach

Введение

Пути

Координаты

Относительные 1

Относительные 2

Пути

Операции

Основные 1

Основные 2

rectangle

circle

arc

grid

to

**foreach**

node

Действия 1

Действия 2

Действия 3

Действия 4

Вершины

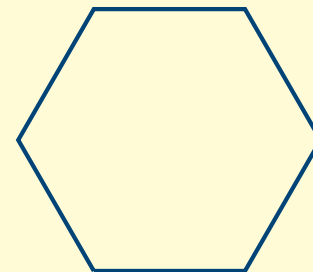
Области

**foreach**  $\langle$ переменные $\rangle$  [ $\langle$ параметры $\rangle$ ] in  $\{\langle$ команды $\rangle\}$

Действует аналогично команде \foreach.

## Пример

```
\tikz %  
  \draw %  
    (0, 0) %  
    foreach \i in %  
      {0, 60, ..., 300} %  
      { -- ++(\i:1)};
```



# Операция node

Введение

Пути

Координаты

Относительные 1

Относительные 2

Пути

Операции

Основные 1

Основные 2

rectangle

circle

arc

grid

to

foreach

**node**

Действия 1

Действия 2

Действия 3

Действия 4

Вершины

Области

**node** *<операторы foreach>* [*<параметры>*] (*<имя\_вершины>*)  
at *<координата>* {*<содержимое\_вершины>*}

Добавляет вершину в текущую координату пути или заданную. Вершины не являются частью пути. При помощи (вложенных) операторов foreach можно определять сразу серию вершин. Выражение at может явно задавать положение вершины.

## Пример

```
\tikz %  
  \path %  
    (0, 0) %  
    node [draw] {Data};
```



# Действия с путями

Введение

Пути

Координаты

Относительные 1

Относительные 2

Пути

Операции

Основные 1

Основные 2

rectangle

circle

arc

grid

to

foreach

node

Действия 1

Действия 2

Действия 3

Действия 4

Вершины

Области

**\path** [*⟨параметры⟩*] {⟨путь⟩}

Выполняет с путём действия, определённые в параметрах. Без указания действий ничего не делает с путём, однако учитывает его при вычислении границ рисунка. При этом возможные команды действий внутри пути будут исполняться, например, параметр `draw` у вершины.

Основные параметры действий команды:

**color** = ⟨цвет⟩

Определяет цвет рисования, заливки и текста для заданного пути.

# Действия с путями (продолжение)

Введение

Пути

Координаты

Относительные 1

Относительные 2

Пути

Операции

Основные 1

Основные 2

rectangle

circle

arc

grid

to

foreach

node

Действия 1

**Действия 2**

Действия 3

Действия 4

Вершины

Области

**draw** = *⟨цвет⟩*

Рисует заданный путь.

**line width** = *⟨размер⟩*

Определяет толщину линии. Определено несколько псевдонимов этого параметра, задающих стандартные значения: **semithick**, **thick**, **very thick** и т. д.

**dash pattern** = *⟨шаблон штриховки⟩*

Определяет способ штриховки линии (on 2pt off 3pt on... ) Определены псевдонимы для предопределённых способов: **solid**, **dotted**, **dashed** и т. д.

# Действия с путями (продолжение)

Введение

Пути

Координаты

Относительные 1

Относительные 2

Пути

Операции

Основные 1

Основные 2

rectangle

circle

arc

grid

to

foreach

node

Действия 1

Действия 2

**Действия 3**

Действия 4

Вершины

Области

**fill** =  $\langle \text{цвет} \rangle$

Выполняет заливку заданного пути, зацикливая его при необходимости.

**shade**

Рисует тень для указанного пути.

*Замечание:* также определены различные команды-псевдонимы: `\draw [...]` вместо `\path [draw = ...]`, `\fill`, `\shadedraw` и т. д.



# Действия с путями (продолжение)

Введение

Пути

Координаты

Относительные 1

Относительные 2

Пути

Операции

Основные 1

Основные 2

rectangle

circle

arc

grid

to

foreach

node

Действия 1

Действия 2

Действия 3

Действия 4

Вершины

Области

**arrows** =  $\langle \text{начало} \rangle - \langle \text{конец} \rangle$

Рисует стрелки в начале/конце пути.  $\langle \text{начало} \rangle$  и  $\langle \text{конец} \rangle$  определяют вид стрелок (могут быть пустыми). Большое количество дополнительных видов стрелок определено в библиотеке `arrows.meta`. Конструкция «`arrows =`» может быть опущена.

## Пример

```
\tikz %
  \draw [Circle-Stealth] %
    (0, 0) -- (1.5, 0); \ \ [2em]
\tikz %
  \draw [{Diamond [open] . %
    Circle [open]}-{Stealth [red]}]%
    (0, 0) -- (1.5, 0);
```



Введение

Пути

**Вершины**

Команды

Свойства 1

Свойства 2

Свойства 3

Свойства 4

Свойства 5

Метки

Дуги

Вдоль

Пример 1

Матрицы

Из вершин 1

Из вершин 2

Расстояния 1

Расстояния 2

Разделитель

Имена

Пример 2

Области

# Вершины и дуги



# Команды вершин

Введение

Пути

Вершины

**Команды**

Свойства 1

Свойства 2

Свойства 3

Свойства 4

Свойства 5

Метки

Дуги

Вдоль

Пример 1

Матрицы

Из вершин 1

Из вершин 2

Расстояния 1

Расстояния 2

Разделитель

Имена

Пример 2

Области

Таблица 1: команды вершин

Команда	Значение
<code>\node</code>	Аналогично <code>\path node</code>
<code>\coordinate</code>	Аналогично <code>\path coordinate</code> — пустая вершина для запоминания её координаты.

Основные свойства вершин:

**shape** =  $\langle \text{фигура} \rangle$

Определяет фигуру вершины. Доступны значения:

**rectangle**, **circle**, **coordinate**, другие фигуры определены в библиотеках.

# Основные свойства вершин

Введение

Пути

Вершины

Команды

Свойства 1

Свойства 2

Свойства 3

Свойства 4

Свойства 5

Метки

Дуги

Вдоль

Пример 1

Матрицы

Из вершин 1

Из вершин 2

Расстояния 1

Расстояния 2

Разделитель

Имена

Пример 2

Области

**inner sep** =  $\langle \text{размер} \rangle$

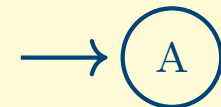
Расстояние от внутреннего текста до границы вершины.  
Аналогично, существуют ключи **inner xsep** и **inner ysep**.

**outer sep** =  $\langle \text{размер} \rangle$

Размер невидимой границы вокруг вершины.

Пример

```
\node %  
  [draw, shape = circle, outer sep = 1mm] %  
  (A) at (2, 0) {A};  
\draw [->] (1, 0) -- (A);
```



# Основные свойства вершин (продолжение)

Введение

Пути

Вершины

Команды

Свойства 1

**Свойства 2**

Свойства 3

Свойства 4

Свойства 5

Метки

Дуги

Вдоль

Пример 1

Матрицы

Из вершин 1

Из вершин 2

Расстояния 1

Расстояния 2

Разделитель

Имена

Пример 2

Области

**minimum size** =  $\langle \text{размер} \rangle$

Размер вершины зависит от содержимого. Но можно устанавливать его минимальный размер. Аналогично, существуют ключи **minimum width** и **minimum height**.

**text** =  $\langle \text{цвет} \rangle$

Цвет текста вершины.

**font** =  $\langle \text{команды шрифтов} \rangle$

Шрифт текста вершины.

**text width** =  $\langle \text{размер} \rangle$

Ширина области текста. В этом случае текст будет автоматически разбиваться на строки.

# Основные свойства вершин (продолжение)

- Введение
- Пути
- Вершины
- Команды
- Свойства 1
- Свойства 2
- Свойства 3**
- Свойства 4
- Свойства 5
- Метки
- Дуги
- Вдоль
- Пример 1
- Матрицы
- Из вершин 1
- Из вершин 2
- Расстояния 1
- Расстояния 2
- Разделитель
- Имена
- Пример 2
- Области

**align** =  $\langle \text{выравнивание} \rangle$

Горизонтальное выравнивание текста. Возможные значения: **left**, **center** и т. д.

**anchor** =  $\langle \text{имя\_якоря} \rangle$

Точка вершины, которая будет расположена в координате, определяемой как координата узла. Возможные значения: **center** (центр), **west** (запад), **south east** (юго-восток) и т. д.

*Замечание:* в качестве вершинных координат можно указывать конструкции вида  $\langle \text{имя\_вершины} \rangle . \langle \text{имя\_якоря} \rangle$ . Если имя якоря пропущено, используется либо **center**, либо для операций рисования линий — точка на границе вершины. △

# Основные свойства вершин (продолжение)

Введение

Пути

Вершины

Команды

Свойства 1

Свойства 2

Свойства 3

**Свойства 4**

Свойства 5

Метки

Дуги

Вдоль

Пример 1

Матрицы

Из вершин 1

Из вершин 2

Расстояния 1

Расстояния 2

Разделитель

Имена

Пример 2

Области

**above** = *⟨смещение⟩*

То же, что и **south**, но позволяет задавать дополнительное смещение вершины. Аналогично, существуют ключи **below** и т. д. При использовании библиотеки `positioning` значение можно указывать как «*⟨смещение⟩ of ⟨координата⟩*».

Пример

```
\node (A) at (0, 0) {A};  
\node [right = of A] {B};  
\node [below = 5mm of A] {C};
```

A diagram illustrating the placement of three nodes. Node A is at the origin (0,0). Node B is positioned to the right of A. Node C is positioned below A, with a vertical dashed line extending from A down to C.

# Основные свойства вершин (окончание)

Введение

Пути

Вершины

Команды

Свойства 1

Свойства 2

Свойства 3

Свойства 4

**Свойства 5**

Метки

Дуги

Вдоль

Пример 1

Матрицы

Из вершин 1

Из вершин 2

Расстояния 1

Расстояния 2

Разделитель

Имена

Пример 2

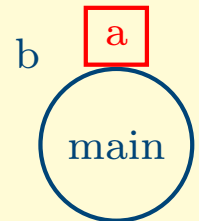
Области

**label** = [*настройки*][*угол*]:*текст*

Добавляет рядом с текущей вершиной дополнительную с заданной надписью (*текст*). *угол* определяет положение относительно исходной вершины, может быть числом (градусы) или именем якоря, по умолчанию имеет значение **above**. *настройки* задают свойства вершины. Для одной и той же вершины можно добавить несколько дополнительных.

## Пример

```
\node %  
  [circle, draw, %  
  label = {[red, draw]above:a}, label = 135:b] %  
{main};
```



# Глобальные настройки, влияющие на метки

Введение

Пути

Вершины

Команды

Свойства 1

Свойства 2

Свойства 3

Свойства 4

Свойства 5

**Метки**

Дуги

Вдоль

Пример 1

Матрицы

Из вершин 1

Из вершин 2

Расстояния 1

Расстояния 2

Разделитель

Имена

Пример 2

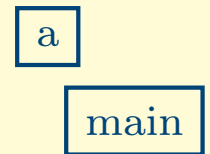
Области

**label distance** =  $\langle \text{расстояние} \rangle$  (по умолчанию: 0pt)

Добавляет дополнительное расстояние между основной вершиной и меткой.

## Пример

```
\begin {tikzpicture} [label distance = 1ex]
  \node %
    [draw, %
    label = {[draw]above left:a}] %
    {main};
\end {tikzpicture}
```



# Соединение вершин дугами

Введение

Пути

Вершины

Команды

Свойства 1

Свойства 2

Свойства 3

Свойства 4

Свойства 5

Метки

Дуги

Вдоль

Пример 1

Матрицы

Из вершин 1

Из вершин 2

Расстояния 1

Расстояния 2

Разделитель

Имена

Пример 2

Области

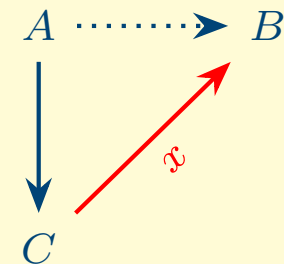
**edge** [*⟨парамтры⟩*] *⟨вершины⟩* (*⟨координата⟩*)

Операция пути, позволяет соединять вершины.

[*⟨парамтры⟩*] действуют локально для дуги. Можно располагать дополнительные *⟨вершины⟩* вдоль дуги.

## Пример

```
\node (A) at (0, 0) {$A$};
\node (B) [right = of A] {$B$};
\node (C) [below = of A] {$C$};
\path [-Stealth] %
  (A) edge [dotted] (B) edge (C) %
  (C) edge [red] %
  node [below, sloped] {$x$} %
  (B);
```





# Размещение вершин вдоль кривых

Введение

Пути

Вершины

Команды

Свойства 1

Свойства 2

Свойства 3

Свойства 4

Свойства 5

Метки

Дуги

**Вдоль**

Пример 1

Матрицы

Из вершин 1

Из вершин 2

Расстояния 1

Расстояния 2

Разделитель

Имена

Пример 2

Области

**pos** =  $\langle \text{доля} \rangle$

Вместо привязки к последней координате вершина располагается вдоль кривой от предыдущей координаты к текущей на расстоянии от начала кривой, составляющем заданную долю от её длины. Определены стандартные ключи, являющиеся синонимами **pos** с разными значениями долей: **midway** (0,5), **near start** (0,25) и т. д.

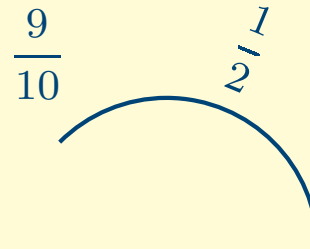
**sloped**

Приводит к повороту вершины вдоль кривой.

# Пример размещения вершин вдоль кривых

## Пример

```
\draw %  
  (0, 0) %  
  arc [radius = 1, %  
       start angle = 0, end angle = 135] %  
  node [midway, above, sloped] { $\frac{1}{2}$ } %  
  node [pos = .9, above left] { $\frac{9}{10}$ };
```



Введение

Пути

Вершины

Команды

Свойства 1

Свойства 2

Свойства 3

Свойства 4

Свойства 5

Метки

Дуги

Вдоль

**Пример 1**

Матрицы

Из вершин 1

Из вершин 2

Расстояния 1

Расстояния 2

Разделитель

Имена

Пример 2

Области

# Матрицы

Введение

Пути

Вершины

Команды

Свойства 1

Свойства 2

Свойства 3

Свойства 4

Свойства 5

Метки

Дуги

Вдоль

Пример 1

**Матрицы**

Из вершин 1

Из вершин 2

Расстояния 1

Расстояния 2

Разделитель

Имена

Пример 2

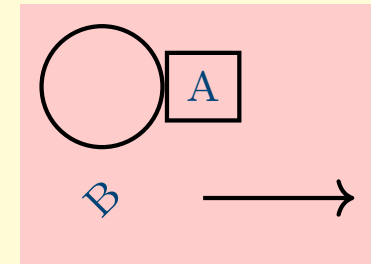
Области

**matrix** = true, false

Определяет, что текущая вершина является матрицей — таблицей из небольших рисунков. Команда `\matrix [...]` является сокращением для «`\node [matrix, ...]`».

## Пример

```
\matrix [fill = red!20!white] %  
{ %  
  \draw (0, 0) %  
    circle [radius = 4mm]; &  
  \node [draw] {A}; \\  
  \node [rotate = 45] {B}; &  
  \draw [->] %  
    (0, 0) -- (1, 0); \\  
};%
```



# Матрицы из вершин

Введение

Пути

Вершины

Команды

Свойства 1

Свойства 2

Свойства 3

Свойства 4

Свойства 5

Метки

Дуги

Вдоль

Пример 1

Матрицы

Из вершин 1

Из вершин 2

Расстояния 1

Расстояния 2

Разделитель

Имена

Пример 2

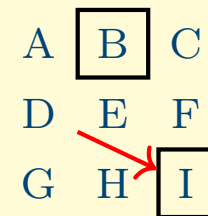
Области

## matrix of nodes

При подключении библиотеки `matrix` позволяет определять матрицы, состоящие из вершин, упрощённым синтаксисом.

### Пример

```
\matrix [matrix of nodes] %  
{ %  
  A & |[draw]| B & C \\  
  |(D)| D & E & F \\  
  G & H & |(I) [draw]|I \\  
};%  
\draw [red, ->] (D) -- (I);
```



# Матрицы из вершин (окончание)

Введение

Пути

Вершины

Команды

Свойства 1

Свойства 2

Свойства 3

Свойства 4

Свойства 5

Метки

Дуги

Вдоль

Пример 1

Матрицы

Из вершин 1

**Из вершин 2**

Расстояния 1

Расстояния 2

Разделитель

Имена

Пример 2

Области

## **matrix of math nodes**

Аналогично, только в начало и конец кода надписи вершины добавляется символ «\$», так что вывод производится в математическом режиме.

**nodes in empty cells** = `true, false` (по умолчанию: **false**)

Определяет, нужно ли помещать вершины в пустые ячейки матриц со стилем `matrix of nodes` или `matrix of math nodes`.

# Расстояния между строками/столбцами

Введение

Пути

Вершины

Команды

Свойства 1

Свойства 2

Свойства 3

Свойства 4

Свойства 5

Метки

Дуги

Вдоль

Пример 1

Матрицы

Из вершин 1

Из вершин 2

**Расстояния 1**

Расстояния 2

Разделитель

Имена

Пример 2

Области

**row sep** =  $\langle \text{размер} \rangle$

**column sep** =  $\langle \text{размер} \rangle$

Определяют расстояния между строками/столбцами.

Пример

```
\matrix %
```

```
[matrix of nodes, %
```

```
row sep = 2ex, column sep = .5ex] %
```

```
{ %
```

```
A & B & C \\
```

```
D & E & F \\
```

```
G & H & I \\
```

```
};%
```

A B C

D E F

G H I

# Расстояния между строками/столбцами (окончание)

*Замечание:* команды, отделяющие строки («`\\`») и столбцы (по умолчанию, «`&`») матрицы в качестве необязательных аргументов также принимают расстояния, аналогично ключам `row sep` и `column sep`. С их помощью можно устанавливать *дополнительные* расстояния между отдельными строками/столбцами.



## Пример

```
\matrix [matrix of nodes, %  
  row sep = 2ex, column sep = .5ex] %  
{ %  
  A & B & [2ex] C \\ [1ex]  
  D & E & F \\  
  G & H & I \\  
};%
```

A	B	C
D	E	F
G	H	I

Введение

Пути

Вершины

Команды

Свойства 1

Свойства 2

Свойства 3

Свойства 4

Свойства 5

Метки

Дуги

Вдоль

Пример 1

Матрицы

Из вершин 1

Из вершин 2

Расстояния 1

**Расстояния 2**

Разделитель

Имена

Пример 2

Области

# Определение разделителя между ячейками

Введение

Пути

Вершины

Команды

Свойства 1

Свойства 2

Свойства 3

Свойства 4

Свойства 5

Метки

Дуги

Вдоль

Пример 1

Матрицы

Из вершин 1

Из вершин 2

Расстояния 1

Расстояния 2

**Разделитель**

Имена

Пример 2

Области

**ampersand replacement** = *⟨команда⟩*

Определяет имя макроса, который заменяет символ «&» в определении матрицы. Бывает необходимо использовать в тех окружениях, где символ «&» имеет специальное значение, например, внутри слайдов класса beamer.

## Пример

```
\matrix %  
  [matrix of nodes, %  
  ampersand replacement = \endcell]%  
{ %  
  A \endcell B \endcell C \\  
  D \endcell E \endcell F \\  
  G \endcell H \endcell I \\  
};%
```

A	B	C
D	E	F
G	H	I



# Имена вершин матрицы

Введение

Пути

Вершины

Команды

Свойства 1

Свойства 2

Свойства 3

Свойства 4

Свойства 5

Метки

Дуги

Вдоль

Пример 1

Матрицы

Из вершин 1

Из вершин 2

Расстояния 1

Расстояния 2

Разделитель

**Имена**

Пример 2

Области

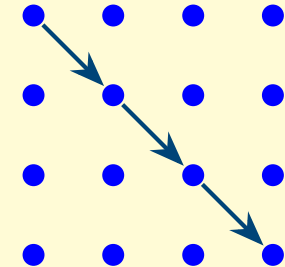
*Замечание:* вершины матрицы типа `matrix of nodes` или `matrix of math nodes` по умолчанию получают имена в виде:  $\langle \text{имя\_матрицы} \rangle - \langle \text{№\_строки} \rangle - \langle \text{№\_столбца} \rangle$ .



# Пример обращения к вершинам матрицы

## Пример

```
\matrix (mat) %  
[ %  
  matrix of nodes, nodes in empty cells, %  
  row sep = 1em, column sep = 1em, %  
  nodes = %  
    {circle, fill = blue, inner sep = .3ex} %  
] %  
{ & & & \\\ & & & \\\ & & & \\\ & & & \\\ };  
  
\foreach \x %  
  [remember = \x as \xp (initially 1)] %  
  in {2, ..., 4} %  
  \path [-Stealth] %  
    (mat-\xp-\xp) edge (mat-\x-\x);
```



Введение

Пути

Вершины

Команды

Свойства 1

Свойства 2

Свойства 3

Свойства 4

Свойства 5

Метки

Дуги

Вдоль

Пример 1

Матрицы

Из вершин 1

Из вершин 2

Расстояния 1

Расстояния 2

Разделитель

Имена

Пример 2

Области

Введение

Пути

Вершины

**Области**

Области

Пример

Цепочка

Имена

Соединение

# Области и цепочки

# Локальные области действия настроек

Введение

Пути

Вершины

Области

**Области**

Пример

Цепочка

Имена

Соединение

```
\begin{scope} [⟨параметры⟩]  
  ⟨содержимое_окружения⟩  
\end{scope}
```

Может находиться внутри окружения `tikzpicture` и ограничивает набор команд, для которых действуют заданные *⟨параметры⟩*. Фактически окружение `tikzpicture` является частным случаем области.

*Замечание:* после подключения пакета `scopes` становится доступным упрощённый синтаксис для областей:

```
«{ [⟨параметры⟩] ... }»
```



# Пример областей

Введение

Пути

Вершины

Области

Области

Пример

Цепочка

Имена

Соединение

## Пример

```
\begin {tikzpicture} [->]
  \draw (0, 3) -- (1, 3);
  { [<-] %
    \draw (0, 2) -- (1, 2);
    \draw (0, 1) -- (1, 1);
  } %
  \draw (0, 0) -- (1, 0);
\end {tikzpicture}
```



# Цепочки вершин

Введение

Пути

Вершины

Области

Области

Пример

**Цепочка**

Имена

Соединение

**Цепочка:** последовательность вершин, расположенных, как правило, вдоль одной прямой. Возможность определения цепочек становится доступной после подключения библиотеки `chains`.

**start chain** =  $\langle \text{имя\_цепочки} \rangle \langle \text{направление} \rangle$

Указывается для области. По умолчанию  $\langle \text{имя\_цепочки} \rangle$  равно **chain**,  $\langle \text{направление} \rangle$  — **going right**.

**on chain** =  $\langle \text{имя\_цепочки} \rangle \langle \text{направление} \rangle$

Указывается для вершины. Определяет её как часть цепочки.

**\chainin** ( $\langle \text{имя\_узла} \rangle$ ) [ $\langle \text{параметры} \rangle$ ]

Добавляет в текущую цепочку уже существующую вершину.

# Имена вершин цепочек

Введение

Пути

Вершины

Области

Области

Пример

Цепочка

**Имена**

Соединение

*Замечание:* первой вершине цепочки автоматически присваивается имя  $\langle \text{имя\_цепочки} \rangle\text{-begin}$ , (текущей) последней —  $\langle \text{имя\_цепочки} \rangle\text{-end}$ .



## Пример

```
{ [start chain = A] %  
  \node [on chain] {A};  
  \node [on chain] {B};  
  \node [on chain] {C};  
} %  
{ [start chain] %  
  \node [on chain, below = of A-begin] {D};  
  \node [on chain] {E};  
  \node [on chain] {F};  
} %
```

A	B	C
D	E	F

# Соединение вершин в цепочке

Введение

Пути

Вершины

Области

Области

Пример

Цепочка

Имена

Соединение

**join** = with *⟨вершина⟩* by *⟨параметры⟩*

Соединяет текущую вершину с другой (по умолчанию, предыдущей), используя *⟨параметры⟩* для рисования дуги.

## Пример

```
\node (A) {A};  
{ [start chain] %  
  \node [on chain, below = of A] {B};  
  \node [on chain, join] {C};  
  \node [on chain, join = by red] {D};  
  \chainin (A) [join = by {->}];  
}
```

