

## 12. Геометрические построения

Функции библиотеки `geometry` позволяют работать в евклидовой геометрии с двухмерными объектами. Заметим, что пакет не поддерживает расширенную плоскость, то есть он не может работать с точками и прямыми в бесконечности. Мы не ставим целью полностью описать эту библиотеку, но постараемся кратко рассказать о принципе работы в пакете и описать в таблице основные команды.

Библиотека `geometry` поддерживает основные геометрические примитивы. Среди них: точка, отрезок, луч, прямая, треугольник, квадрат, окружность, эллипс, парабола, гипербола и другие. Каждый объект обязательно имеет свои имя и размер. Для некоторых из них необходимо определить специфические параметры, например, для окружности - имя центра. Примитив задается с помощью одноименных команд. Например, для определения точки используется функция `point` ( $P, P_x, P_y$ ), где  $P$  - имя точки,  $P_x$  - горизонтальная координата,  $P_y$  - вертикальная координата. Треугольник можно задать несколькими способами:

```
triangle(T, [A, B, C], n ),
triangle(T, [l1, l2, l3], n ),
triangle(T, [side1, side2, side3] ),
triangle(T, [side1, 'angle'=theta, side3], n ),
```

где  $T$  - имя треугольника;  $A, B, C$  - три точки;  $L1, l2, l3$  - три прямые;  $Side1, side2, side3$  - три стороны треугольника;

$Side1, "angle" = theta, side3$  - две стороны треугольника и  $theta$  - угол между ними;

$n$  - список из двух имен, представляющих названия горизонтальной и вертикальной оси соответственно (необязательный параметр).

Таким же образом определяются и остальные примитивы. Кроме того, есть команды, определяющие основные параметры фигуры, такие как способ задания примитива, координаты, радиус, площадь, алгебраическое уравнение и многие другие.

```
Для графической визуализации объектов используется функция
draw ([obj_1 (localopts_1) , ..., obj_n (localopts_n )], globalopts);
где obj_1, ..., obj_n - геометрические объекты,
localopts_1, ..., localopts_n - параметр для каждого объекта,
globalopts - параметры, которые относятся ко всем объектам.
```

*Localopts\_i* - последовательность параметров: цвет, стиль линии, количество точек, стиль, символ, толщина, текст, заполнение.

---

© Прохоров Г.В., Колбеев В.В., Желнов К.И., Леденев М.А., 1998

«Математический пакет Maple V Release 4».

При перепечатке ссылка на первоисточник обязательна.

Ниже приведены основные команды из библиотеки `geometry`.

Объекты можно подвергнуть различным преобразованиям: перенос вдоль горизонтальной или вертикальной осей, различные пово-

| <b>КОМАНДА</b>             | <b>ОПИСАНИЕ</b>  |
|----------------------------|--|
| <b>area(A1)</b>            | Определение площади объекта <b>A1</b> (треугольника, квадрата, круга, и т. п.)   |
| <b>asymptotes (P)</b>      | Определение асимптот гиперболы <b>P</b>  |
| <b>bisector(T)</b>         | Определение биссектрисы треугольника <b>T</b>  |
| <b>center(A1)</b>          | Определение центра круга, эллипса, гиперболы   |
| <b>circle(c,m)</b>         | Определение круга с именем "с" методом <b>m</b>  |
| <b>conic(k, m)</b>         | Определение конуса с именем "k" методом <b>m</b>   |
| <b>coordinates(p)</b>      | Определение координат объекта <b>p</b> [P <sub>x</sub> , P <sub>y</sub> ], где P <sub>x</sub> - горизонтальная координата и P <sub>y</sub> - вертикальная координата |
| <b>definedAs (Sq)</b>      | Определение четырех вершин квадрата <b>Sq</b>  |
| <b>detail(A1)</b>          | Детальное описание объекта <b>A1</b>   |
| <b>diagonal(Sq)</b>        | Определение длины диагонали квадрата   |
| <b>hyperbola(H, m)</b>     | Определение гиперболы с именем "H", методом <b>m</b>   |
| <b>diameter(V)</b>         | Вычисление диаметра окружности <b>V</b>  |
| <b>distance(a, b)</b>      | Вычисление расстояния между двумя объектами  |
| <b>draw(A1)</b>            | Построение объекта <b>A1</b>   |
| <b>ellipse(E, m)</b>       | Определение эллипса с именем "E" методом <b>m</b>  |
| <b>Equation(A1)</b>        | Вывод уравнения геометрического объекта <b>A1</b>  |
| <b>Foci(E)</b>             | Нахождение фокуса эллипса или гиперболы  |
| <b>form(P)</b>             | Определение формы геометрического объекта (то есть, <code>point2d</code> , если <b>P</b> - точка)  |
| <b>HorizontalCoord (P)</b> | Определение значения P <sub>x</sub> , которое является горизонтальной координатой точки <b>P</b>   |
| <b>MajorAxis (p)</b>       | Вычисление длины главной оси эллипса <b>p</b>  |
| <b>Median(t)</b>           | Определение медианы треугольника <b>t</b>  |
| <b>method(T)</b>           | Вывод метода определения объекта <b>T</b>  |
| <b>MinorAxis (p)</b>       | Определение длины малой оси эллипса  |
| <b>parabola (P,m)</b>      | Определение параболы с именем "P", методом <b>m</b>  |
| <b>radius (c)</b>          | Определение радиуса окружности <b>c</b>  |
| <b>square(K,m)</b>         | Определение квадрата с именем "K", методом <b>m</b>  |
| <b>Vertex (p)</b>          | Вывод названия вершины объекта <b>p</b>  |
| <b>VerticalCoord (P)</b>   | Определение значения P <sub>y</sub> , которое является вертикальной координатой точки <b>P</b>   |

© Прохоров Г.В., Колбеев В.В., Желнов К.И., Леденев М.А., 1998

«Математический пакет Maple V Release 4».

При перепечатке ссылка на первоисточник обязательна.

роты и сдвиги и т. п.

В качестве иллюстрации приведем доказательство теоремы о точке пересечения биссектрис в треугольнике, которая гласит:

*Центр окружности, вписанной в треугольник, является точкой пересечения его биссектрис.*

```
> restart: with(geometry):
```

Построим треугольник T:

```
> triangle(T, [point(A,0,0),point(B,3,-1),
point(C,4,2)]):
```

Построим окружность inc, вписанную в треугольник с центром O1.

```
> incircle(inc,T, 'centername'=O1):
```

Построим биссектрисы bA, bB, bC углов треугольника T:

```
> bisector(bA,A,T): bisector(bB,B,T):
bisector(bC,C,T):
```

Проверим, действительно ли центр вписанной окружности O1 лежит на пресечении трех биссектрис:

```
IsOnLine(O1,bA) and IsOnLine(O1,bB) and
IsOnLine(O1,bC);
```

true

Мы убедились, что точка O1 действительно принадлежит биссектрисам bA, bB, bC. Подтвердим вышесказанное рисунком:

```
> draw([inc(color=green), bA(color=black),
bB(color=black), bC(color=black), T(color=blue,
filled=false)], filled=true,
scaling=constrained);
```

