|  |  |  |  |
| --- | --- | --- | --- |
| 类型 | | 命令格式 | 解释 |
| 移动指令  Move Command（M） | | M x,y  或  m x,y | 比如：M 100,240 或 m 100,240   大写的 M 指示 x,y 是绝对值；  小写的 m 指示 x,y 是相对于上一个点的偏移量，如果是 (0,0)，则表示不存在偏移。  当您在 move 命令之后列出多个点时，即使您指定的是线条命令，也将绘制出连接这些点的线。 |
| 绘制指令  （Draw Command）   通过使用一个大写或小写字母输入各命令：   其中大写字母表示绝对值，  小写字母表示相对值。   线段的控制点是相对于上一线段的终点而言的。   依次输入多个同一类型的命令时，可以省略重复的命令项；   例如，L 100,200 300,400 等同于 L 100,200 L 300,400。 | 直线:Line(L) | 格式：  L 结束点坐标  或:  l 结束点坐标。 | 比如：L 100,100 或 l 100 100。  坐标值可以使用x,y（中间用英文逗号隔开）或x y（中间用半角空格隔开）的形式。 |
| 水平直线: Horizontal line(H) | 格式：  H x值  或  h x值  (x为System.Double类型的值) | 绘制从当前点到指定x坐标的直线。  比如：H 100或h 100，也可以是：H 100.00或h 100.00等形式。 |
| 垂直直线: Vertical line(V) | V y值  或  v y值  (y为System.Double类型的值) | 绘制从当前点到指定y坐标的直线。  比如：V 100或y 100，也可以是：V 100.00或v 100.00等形式。 |
| 三次方程式贝塞尔曲线:  Cubic Bezier curve(C) | C 第一控制点 第二控制点 结束点  或  c 第一控制点 第二控制点 结束点 | 通过指定两个控制点来绘制由当前点到指定结束点间的三次方程贝塞尔曲线。  比如：C 100,200 200,400 300,200 或 c 100,200 200,400 300,200  其中，点(100,200)为第一控制点，点(200,400)为第二控制点，点(300,200)为结束点。 |
| 二次方程式贝塞尔曲线:  Quadratic Bezier curve(Q) | Q 控制点 结束点  或  q 控制点 结束点 | 通过指定的一个控制点来绘制由当前点到指定结束点间的二次方程贝塞尔曲线。  比如：q 100,200 300,200。其中，点(100,200)为控制点，点(300,200)为结束点。 |
| 平滑三次方程式贝塞尔曲线:  Smooth cubic Bezier curve(S) | S 控制点 结束点  或  s 控制点 结束点 | 通过一个指定点来“平滑地”控制当前点到指定点的贝塞尔曲线。   比如：S 100,200 200,300 |
| 平滑二次方程式贝塞尔曲线:  smooth quadratic Bezier curve(T) | T 控制点 结束点  或  t 控制点 结束点 | 与平滑三次方程贝塞尔曲线类似。在当前点与指定的终点之间创建一条二次贝塞尔曲线。控制点假定为前一个命令的控制点相对于当前点的反射。如果前一个命令不存在，或者前一个命令不是二次贝塞尔曲线命令或平滑的二次贝塞尔曲线命令，则此控制点就是当前点。   比如：T 100,200 200,300 |
| 椭圆圆弧: elliptical Arc(A) | A 尺寸 圆弧旋转角度值 优势弧的标记 正负角度标记 结束点  或：  a 尺寸 圆弧旋转角度值 优势弧的标记 正负角度标记 结束点 | 在当前点与指定结束点间绘制圆弧。   * 尺寸(Size): System.Windows.Size类型，指定椭圆圆弧X,Y方向上的半径值。 * 旋转角度（rotationAngle）：System.Double类型。 * 圆弧旋转角度值（rotationAngle）：椭圆弧的旋转角度值。 * 优势弧的标记（isLargeArcFlag）：是否为优势弧，如果弧的角度大于等于180度，则设为1，否则为0。 * 正负角度标记（sweepDirectionFlag）：当正角方向绘制时设为1，否则为0。 * 结束点（endPoint）：System.Windows.Point类型。   比如：A 5,5 0 0 1 10,10 |
| 关闭指令  (close Command):  可选的关闭命令 | | 用Z或z表示 | 用以将图形的首、尾点用直线连接，以形成一个封闭的区域。 |
| 填充规则（fillRule）   如果省略此命令，则路径使用默认行为:即 EvenOdd。  如果指定此命令，则必须将其置于最前面。> | EvenOdd 填充规则 | F0 指定 EvenOdd 填充规则。 | EvenOdd 确定一个点是否位于填充区域内的规则方法：从该点沿任意方向画一条无限长的射线，然后计算该射线在给定形状中因交叉而形成的路径段数。如果该数为奇数，则点在内部；如果为偶数，则点在外部。 |
| Nonzero 填充规则 | F1 指定 Nonzero 填充规则。 | Nonzero 确定一个点是否位于路径填充区域内的规则方法：从该点沿任意方向画一条无限长的射线，然后检查形状段与该射线的交点。从 0 开始计数，每当线段从左向右穿过该射线时加 1，而每当路径段从右向左穿过该射线时减 1。计算交点的数目后，如果结果为 0，则说明该点位于路径外部。否则，它位于路径内部。 |