more involved than creating a rectangle or circle and the subtleties of creating paths of specific curvature are outside this book's scope. But we do examine the different path classes and properties Core Suite provides in creating a path, but our resulting shape is randomly created and does not reflect any true knowledge of paths.

To create a shape using a path, start with a `Path` instance. The `Path` class has six different constructor overloads, but all six take the path's starting `x,y` position as parameters. You then "draw" the path by adding one or more of the three available sub-paths. Programatically, of course, you add a sub-path to the path's `SubPaths` collection via its `Add` method.

There are four types of sub-paths: `CurveFromSubPath`, `CurveSubPath`, `CurveToSubPath`, and `LineSubPath`. When creating a sub-path you "draw" from either the path's starting point, or from the end-point of the last path. By adding multiple sub-paths you create a complete shape. You close the path by specifying the path's `ClosePath` property as `true` or `false`. As with the other page elements covered in this chapter, you can set a path's line and fill color, its line style and thickness, and other properties. For instance, you can set how the line segments join by setting the path's `LineJoin` property to `LineJoin.Bevel`, `LineJoin.Miter`, or `LineJoin.Round`.

Figure 5.4.4 illustrates the steps taken in Listing 5.4. Imagine you

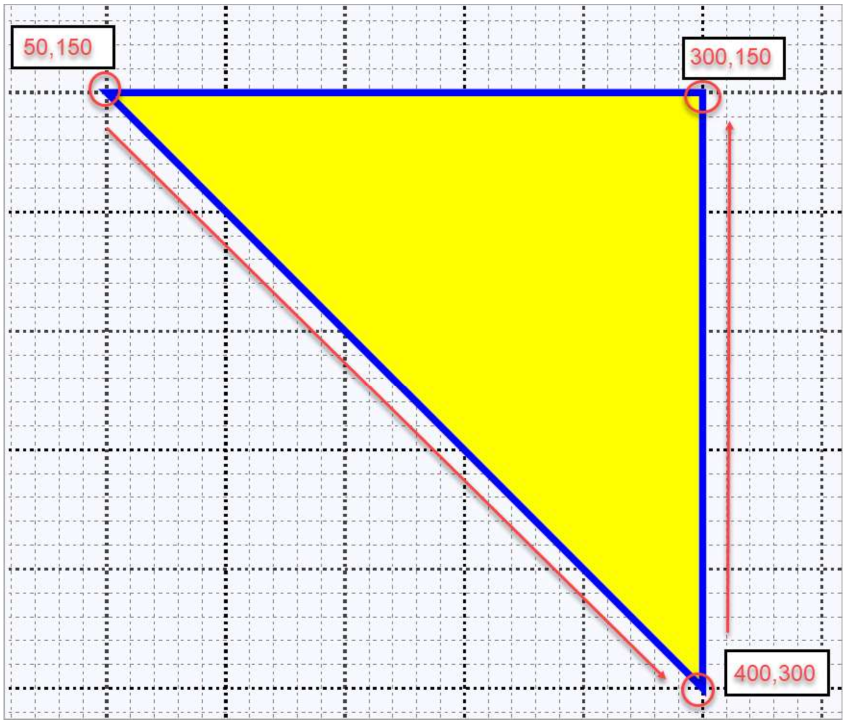

Figure 5.4.1: Adding a triangle to a PDF document.