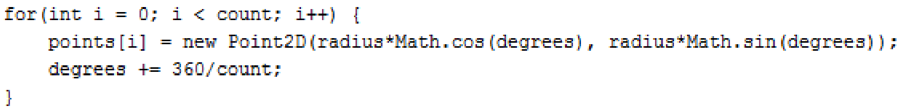
In order to accurately assess the efficiency of our algorithms we came up with two different ways to generate test data: circular points and random points. When given circular points the efficiency of quick hull should decrease significantly, because each time it draws a triangle and searches for points inside the triangle it will find nothing, so the convex hull would be the entire set of points. The random data generated would give closer to average case efficiency, because usually provided points would be closer to random distribution.

In order to generate a circular set of points the function below received a count of how many points need to be generated. At the start of the loop degrees is set to 0, and is incremented by 360 divided by the total number of points so that the data will be distributed well around the circle. Each point is given an x value of the radius multiplied by the cos of the current value of degrees, and the y value is generated using the sin function. The random data is generated point by point and uses the Math.random function.



```
for(int i = 0; i < count; i++) {
    points[i] = new Point2D(radius*Math.cos(degrees), radius*Math.sin(degrees));
    degrees += 360/count;
}
```