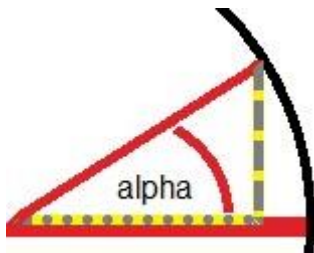


## Greenfoot – Projektphase

### Position auf Kreisbahn

Um die Position eines Objekts auf einer Kreisbahn zu errechnen, müsst ihr ein wenig Mathematik einsetzen. erinnert euch an die Definition der trigonometrischen Funktionen Sinus und Cosinus am Einheitskreis. In der Grafik sind die beiden roten Geraden der Radius, die gestrichelte und gepunktete Linie sind die beiden Katheten. Vervollständigt die folgenden Formeln:



$$\sin(\alpha) =$$

$$\cos(\alpha) =$$

Stellt die Formeln nun jeweils nach der Kathete um:

$$=$$

$$=$$

Kennt ihr also die Koordinaten des Mittelpunkts und den Radius eines Kreises, könnt ihr zu jedem Winkel die dazugehörige Position auf dem Kreis errechnen. JAVA bietet bereits Methoden zur Berechnung des Sinus und Cosinus in der Klasse Math an:

```
public static double sin (double a)
public static double cos (double a)
```

Beispiele für die Benutzung im Quelltext:

```
double x = Math.cos(360);
double y = Math.cos(2*x/360);
```

Die Methoden bekommen jeweils den Winkel als Wert vom Typ double als Parameter übergeben. Allerdings arbeiten diese Methoden nicht mit dem Gradmaß, sondern mit dem Bogenmaß. Daher müsst ihr den Winkel noch umrechnen:



Für die Winkel  $\alpha$  im Gradmaß und  $\varphi$  im Bogenmaß gilt:  $\varphi = \frac{2\pi}{360} * \alpha$

Stellt nun die Formeln für die x- und y-Koordinate für den Winkel  $\alpha$  zusammen. Nehmt dabei an, dass der Mittelpunkt M die Koordinaten  $(x_m, y_m)$  und der Radius den Betrag  $r$  haben:

## Greenfoot – Projektphase

x =

y =

Berechnet unter Verwendung des Taschenrechners (Start -> Alle Programme -> Zubehör -> Rechner) die zu den gegebenen Winkeln gehörenden x- und y-Koordinaten für den Fall, dass der Mittelpunkt des Kreises im Koordinatenursprung liegt und der Radius den Betrag 123 hat (rundet erst beim Endergebnis und dann auf 0 Dezimalstellen):

	0°	15°	30°	45°	60°	90°	135°	245°	360°
x									
y									

Nun sollt ihr die Methode `berechnePosition` implementieren, welche euch später die Arbeit erleichtern soll. Diese Methode bekommt als Parameter die x- und y-Koordinate des Mittelpunktes sowie den Radius als Wert vom Typ `int` und den Winkel  $\alpha$  als Wert vom Typ `double` übergeben. Sie gibt dann ein eindimensionales array mit 2 Einträgen zurück, wobei der erste die x- und der zweite die y-Position auf der Kreisbahn ist. Die Methode hat also folgende Signatur:

```
public int[] berechnePosition(int xm, int ym, int radius, double winkel)
```

Verwendet während der Implementation für Pi den Wert der Konstanten aus der Klasse `Math`:

```
public static final double PI
```

Wollt ihr also zum Beispiel der Variablen `pi` den Wert von Pi zuweisen, so sähe das wie folgt aus:

```
double pi = Math.PI;
```

## Greenfoot – Projektphase

Schreibt eure Methode in den folgenden Kasten:

Verwendet diese Methode immer in euren Quelltexten, wenn ihr die Position auf einer Kreisbahn berechnen müsst

## Greenfoot – Projektphase



### Quellenverzeichnis:

**Trigonometrische Funktionen** – Quelle: InfoSphere

 ,  ,  ,  ,  angefertigt vom InfoSphere-Team