# Point간 각도 및 거리 구현

김도윤

Go Code(main)

```
package main

import (
    "DOTPRODUCT/point"
    "math"
)

func main() {
    p1 := Point{1, 2, 3, 4}
    p2 := Point{4, 5, 6, 2}
    angle := angleBetween(p1, p2)
    fmt.Printf("Angle between points: %.4f radians\n", angle)
    dist := distance(p1, p2)
    fmt.Printf("Distance between points: %.4f\n", dist)
}
--------------------------------------------------------------------------------
### Go Code(point package)
package point

import (
    "math"
)

// Point type representing a point in n-dimensional space
type Point []float64

// dotProduct calculates the dot product of two vectors
func dotProduct(a, b Point) float64 {
    if len(a) != len(b) {
        panic("Vectors must be of same dimension")
    }
    var result float64
    for i := range a {
        result += a[i] * b[i]
    }
    return result
}

// magnitude calculates the magnitude of a vector
func magnitude(a Point) float64 {
    return math.Sqrt(dotProduct(a, a))
}

// normalize normalizes a vector to a unit vector
```

```
func normalize(a Point) Point {
    mag := magnitude(a)
    unit := make(Point, len(a))
    for i := range a {
        unit[i] = a[i] / mag
    }
    return unit
}

// angleBetween calculates the angle between two vectors using their dot product and magnitudes
func AngleBetween(a, b Point) float64 {
    if len(a) != len(b) {
        panic("Vectors must be of same dimension")
    }
    unitA := normalize(a)
    unitB := normalize(b)
    dot := dotProduct(unitA, unitB)
    return math.Acos(dot)
}

// distance calculates the distance between two points using the Law of Cosines
func Distance(a, b Point) float64 {
    if len(a) != len(b) {
        panic("Points must be of same dimension")
    }
    angle := AngleBetween(a, b)
    return math.Sqrt(magnitude(a)*magnitude(a) + magnitude(b)*magnitude(b) - 2*magnitude(a)*magnitude(b)*math.Cos(angle))
}



-----------------------------------------------------------------
-실행결과
Angle between points: 0.6242 radians
Distance between points: 5.5678
-----------------------------------------------------------------
```