**Pointer คืออะไร**

คือประเภทข้อมูลแบบหนึ่งของ Go ครับ มันเก็บตำแหน่ง (Address) อ้างอิงของข้อมูลประเภทอื่น แต่เรามักจะเรียกว่ามันชี้ไปที่ตำแหน่งไหน มันเลยได้ชื่อว่า Pointer ที่แปลแบบตรงๆหน่อยคือ ตัวชี้เอาแบบจำง่ายๆคือ Pointer ชี้ไปที่ตำแหน่งใดตำแหน่งหนึ่งหรือไม่ชี้ไปที่ไหนเลย แล้วยังไงต่อ เวลามันชี้ไปที่ตำแหน่งไหน เราสามารถจัดการกับข้อมูลในตำแหน่งนั้นได้โดยอ้อมๆ

**ประกาศตัวแปร Pointer ให้แต่ละ Type ด้วย Operation \*<Type>**

ตัวอย่าง ประกาศตัวแปร p ชนิด \*int เพื่อใช้เก็บ pointer ชี้หา memory address ประเภท type int

**ส่ง Memory Address ด้วย Operator &**

วิธีการอ่าน memory address เรามักใช้ & ใส่ไว้หน้าตัวแปร เช่น ตัวอย่างนี้ต้องการส่ง pointer int ของ ตัวแปร i ให้กับตัวแปร p

**อ่านค่าจาก Pointer ด้วยการใส่ \* หน้าตัวแปร**

เราสามารถอ่านค่าใน pointer p ที่กำลังชี้หา i ด้วยการใส่ operation \*p

**ข้อแตกต่างระหว่าง struct กับ interface**

**Struct** คือ ชนิดของข้อมูล หรือโครงสร้างของข้อมูลที่เราสามารถกำหนดเองได้ โดยภายใน Struct เราจะมี Field ของข้อมูลต่างๆ เก็บไว้

type Person struct {

Id string

Name string

}

**Interface** คือ Interface ในภาษา Go นั้นเป็นความสามารถที่แจ่มมากแต่มันกลับสร้างความสร้างความสับสนให้กับผู้เริ่มต้นศึกษาเช่นตัวผมเองอย่างมากดังนั้น จึงลองมาทำความเข้าใจกันหน่อยว่าinterface ในภาษา Go มันเป็นอย่างไร และใช้งานอย่างไร

เริ่มต้นก่อนว่า interface ในภาษา Go นั้นไม่มีการเขียน implement แบบในภาษาอื่นๆ เช่น

class MyImplement implements MyInterface {}

แต่สามารถใช้งานเพียงกำหนด function ที่ต้องการใน interfaceแล้ว compiler จะจัดการให้เองและจงจำไว้ว่าinterface คือ type ชนิดหนึ่งนะครับ

ตัวอย่างเช่น

สร้าง interface ชื่อ Shape โดยมี function Area() ไว้สำหรับคำนวรพื้นที่ของ Shape ต่างๆ

type Shaper interface {

Area()

}

ใน Go นั้น interface จะต้องลงท้ายด้วย -er เสมอ เป็นเพียงข้อตกลงเท่านั้นใช้หรือไม่ใช้ก็ได้ ใช้เป็นการแยกแยะให้ชัดเจนเท่านั้นเอง

เมื่อต้องการสร้าง implement class แล้วสามารถสร้างได้ดังนี้

type Rectangle struct {

width float64

height float64

}

func (rectangle Rectangle) Area() {

return rectangle.width \* rectangle.height

}

สังเกตได้ว่า

ชื่อ function Area() ตรงกับที่ประกาศไว้ใน interface Shaperดังนั้น Rectangle จึงทำการ implement มาจาก interface Shaper โดยอัตโนมัติทำให้ภาษา Go แตกต่างจากภาษอื่นๆส่วนแนวคิดเรื่อง interface ยังคงเหมือนในโลก OOP

**ทำไมต้องใช้ package สำหรับภาษา Go**

Go packages คือ การแบ่ง Sorce code ของเราออกแป็นส่วนๆได้ ยกตัวอย่างเช่น โดยปกติแล้วโปรแกรมที่เราเขียนขึ้นมาสักตัวหนึ่ง แน่นอนว่าภายในจะประกอบไปด้วยส่วนต่างๆมากมาย ดังนั้นเพื่อความเป็นระเบียบ ดูแลได้ง่าย และที่สําคัญสามารถ Reuse sorce code ได้ เราจําเป็นจะต้องแยก Sorce code ออกเป็นส่วนๆ โดยใช้ความสามารถของ Go packages นั้นละครับ และไม่เพียงแค่แบ่ง Sorce code ภายใน Project เท่านั้น Go packages ยังสามารถนําไปใช้ประโยชน์เมื่อเราต้องการจะ Public sorce code ของเราให้คนอื่นๆ ได้ใช้อีกด้วย