

```
type Shape interface {  
    Area() float64  
}
```

```
type Circle struct {  
    Radius float64  
}
```

```
func (c Circle) Area() float64 {  
    return math.Pi * c.Radius * c.Radius  
}
```

```
type Rectangle struct {  
    Width float64  
    Height float64  
}
```

```
func (r Rectangle) Area() float64 {  
    return r.Width * r.Height  
}
```

```
func GetShape() Shape {  
    if someCondition {  
        return Circle{Radius: 5}  
    } else {  
        return Rectangle{Width: 10, Height: 5}  
    }  
}
```

```
func PrintArea(s Shape) {  
    fmt.Println("Area:", s.Area())  
}
```

```
func main() {  
    circle := Circle{Radius: 5}  
    rectangle := Rectangle{Width: 10, Height: 5}  
    PrintArea(circle) // Полиморфно обращаемся с кругом «Circle»  
                      // как с фигурой «Shape»  
    PrintArea(rectangle) // Полиморфно обращаемся с прямоугольником  
                        // «Rectangle» как с фигурой «Shape»  
}
```