

2주차 고

go mod init gowork/syntax      go build      syn하고 탭

Println을 쓰려면 import fmt" 해야함

fmt.Println

math 패키지 Ceil 올림 Floor 버림 Round 반올림

fmt랑 같이쓰면 import (

    "fmt"

    "math"

)

strings에 Title함수는 단어 앞글자 다 대문자로

var char rune ="가" rune은 한글자 단어 int32 %T

println 하면 44032 유니코드 %c해줘야 가 가 나옴

var a int = 7

var a = 7

a:=7 추론기능

var b float64 = 5.3

a = int(b) 강제 형변환

3주차

reflect 패키지의 reflect.TypeOf() 는 %T와 같음

변수명은 영문자로 영문대문자면 다른패키지에서 접근가능 소문자면 동일패키지만

zero value      string 은 null bool은 false 나머지 0

```
texts := "G@ M@ney"
```

```
r := strings.NewReplace("@","o"); //r이 이제 @를 o로 바꿔주는애
```

```
new := r.Replace(texts)
```

os, bufio 패키지

```
re := bufio.NewReader(os.Stdin) 입력 받기 스탠다드 인풋
```

```
in, err := re.ReadString('\n') 줄바꿈 글자가 올때까지 읽음 성공하면 in에 실패시 err에
```

```
in은 String 타입 in=strings.TrimSpace(in)
```

```
dan, err := strconv.ParseInt(in,10(진수),32(비트))
```

```
dan, err := strconv.Atoi(in) //자동으로 읽어서 형변환
```

```
log.Fatal(err) err에 성공하면 nil 실패시 에러 하고 종료
```

```
if err!= nil{
```

```
    log.Fatal(err)//에러가 있는거
```

```
} else { fmt.Println(in)}//에러없음
```

```
for i:=1; i<10; i++){
```

```
dan*i 하면 에러 dan을 강제 형변환 int(dan) 해줘야함
```

```
}
```

```
i:=1
```

```
for i<10 { i++} while문 형태
```

math/rand 패키지

time 패키지

```
seed:=time.Now().Unix() 계속 바뀌는거
```

```
rand.Seed(seed) 랜드에 시간 마다 변하게
```

```
dice:=rand.Intn(6)+1 1~6 랜덤
```

4주차 소수 판정

git clone 깃주소 .은폴더 다 옮기나 폴더 만들어서 옮기나 클론해서 버전생성 가능

n, err := fmt.Scanln(&number) number에 입력받음

n은 입력받은 개수 err은 에러 number에 맞는 타입이 안오면 에러

os.Exit(1) 0은 정상적 종료 1은 뭔가 잘못됐다

새도우

var float64 float64 = 9.1

변수의 이름을 패키지이름, 자료타입, 함수 같은거로 하면 가려진다

함수 func sayHello(){

fmt.Println() //매개변수 반환값 없는 함수

}

func adsf(name string, kor int) <<매개변수 리턴없음

func adsf(name string, kor int) (int, int) <<매개변수 리턴인트형 두 개

return 계산한거, 계산한거

t, a := adsf('가', 16) 리턴 두 개라 두 개로 받음

return을 int와 error 로 했을 때 return 계산하고, rut.Errorf("에러라능")가능

5주차

fmt.Scanln(&a,&b) 두 개 받기 가능

5주차 function(1) 파일 보기 소수처리 최종본

포인터

```
var a int = 10
```

println(&a) 는 a 주소

```
var pa *int
```

```
pa = &a
```

println(pa,\*pa) > 주소 , pa가 가리키는 값

swap 할 때 포인터로 해줘야 바뀜

패키지

gowork 폴더(작업소)에서 packages폴더(큰 폴더) mymath에 함수를 쓸거면

```
import "gowork/packages/mymath"
```

실행파일에서 go mod init gowork/packages 후 실행파일 go build

함수는 무조건 대문자 시작

6주차 배열 array

타입이 같은 값 여러개 갖는 짜르르

```
var primes [3]int
```

```
primes[0] = 1
```

```
var primes [3]int = [3]int{2,3,5}
```

```
var primes [3]int = [3]int{} 제로값 0 들어감
```

```
primes :=[3]int{2,3,5} 단축변수 채고
```

포문이나 반복문 돌 때 배열의 길이 len(primes) for i<len(primes)

```
for index, prime := range primes{      앞에 index에 _ 사용가능
```

```
    fmt.Println(index, prime)
```

```
}    primes 배열을 순회하며 index와 prime 찍음
```

```
    var sum int = 0
```

```
    for _, prime := range primes {
```

```
        //fmt.Println(prime)
```

```
        sum = sum + prime
```

```
    }    //인덱스 안쓰고 합계 구하기
```

slice 씨샵으로 치면 list 크기 늘렸다 줄였다 가능 배열은 불가능

```
var primes [3]int = [3]int{2,3,5} //배열
```

```
var s []int    > 지금 아~무것도 없음[]
```

```
s = make([]int,5) 저장공간 5개 만듦 [0, 0, 0, 0, 0]
```

```
s[5]에 하나 더 넣고 싶으면 s = append(s,-9) 하면 5번에 -9가 들어감
```

```
var s []int = []int {0,0,0,0,0} 도 가능
```

```
s:= []int {0,0,0,0,0}도 가능
```

```
s:= make([]int,5) 도 가능
```

```
s = append(s,-9, 1, 5) 여러개도 가능 쭈루룩 늘어남
```

배열은 값에의한 전달이라 함수로 바꾸면 안바뀔 근데 슬라이스는 주소에 의한 전달이라 함수에서 값 바꾸면 바뀔

```
a:= [...]int {2,3,5} 는 배열
```

6주차 슬라이스 계속

파일 읽어오기 scores.txt에서

```
import fmt bufio log os
```

```
fs, err := os.Open("scores.txt")
```

에러나면 err처리 안나면 fs에 스코어 열어서 기능하게 (읽는도중에러처리)

```
scanner := bufio.NewScanner(fs)
```

```
for scanner.Scan(){ //스캔했는데 값이 있으면
    fmt.Println(scanner.Text())    한줄씩 읽음
}
```

err =fs.Close() 파일을 열었으면 닫아야함

에러 나면 err처리 (닫는도중에러)

```
if scanner.Err()!=nil{
```

```
    log.Fatal(scanner.Err()) 파일 읽는 도중 에러나면 에러처리
```

이걸 함수로 처리 불필요한 반환엔 nil 써도댐

```
func LoadFloats(fn string) ([]float64, error) {
```

```
    var numbers []float64
```

```
    fs, err := os.Open(fn)
```

```
    if err != nil {
```

```
        return numbers, err
```

```
    }
```

```
    scanner := bufio.NewScanner(fs)
```

```
    for scanner.Scan() {
```

```
        number, err := strconv.ParseFloat(scanner.Text(), 64)
```

```
        if err != nil {
```

```
            return numbers, err
```

```
        }
```

```
        numbers = append(numbers, number)
```

```
        //fmt.Println(scanner.Text())
```

```
    }
```

```
    err = fs.Close()
```

```
    if err != nil {
```

```
        return numbers, err
```

```
    }
```

```
    if scanner.Err() != nil {
```

```
        return numbers, scanner.Err()
```

```
    }
```

```
    return numbers, err
```

```
}
```

```

numbers, err := LoadFloats("scores.txt")
    if err != nil {
        log.Fatal(err)
    }
    for _, number := range numbers {
        sum = sum + number
    }
    fmt.Printf("평균 : %.2f\n", sum/float64(len(numbers)))

```

텍스트파일 불러와서 평균구하기 위해 슬라이스버전

```

func LoadFloats(fn string) ([4]float64, error) {
//    var numbers [4]float64
//    var i int = 0
//    fs, err := os.Open(fn)
//    if err != nil {
//        return numbers, err
//    }
//    scanner := bufio.NewScanner(fs)
//    for scanner.Scan() {
//        numbers[i], err = strconv.ParseFloat(scanner.Text(), 64)
//        if err != nil {
//            return numbers, err
//        }
//        i++
//        //fmt.Println(scanner.Text())
//    }
//    err = fs.Close()
//    if err != nil {
//        return numbers, err
//    }
//    if scanner.Err() != nil {
//        return numbers, scanner.Err()
//    }
//    return numbers, err
// }

```

이건 배열버전

7주차

슬라이스 slicing

슬라이스를 잘라내서 이용하기

```
a := []string {"a","b","c","d","e"}
```

as:=a[0:2]     a와 b출력 0부터 2 전까지                     배열도 가능

as[1] = "z"를 하면 a도 바뀌고 as는 당연히 바뀜 슬라이스라서 주소?

a := make([]string, 4, 5)     > len 4 cap 5     메모리 할당 미리 받기

c := append(a, "y")     c는 len 5 cap 5 a z c d y가 됨

c := append(a, "x","y") 하면 len은 6 cap 은10 (임의로 잡힘 여유분까지 보통2배)

값 4개 더 추가하면 len 10 cap 10 여기서 하나 더 추가하면 len 11 cap 20 이됨

a := []string{"a", "b", "c", "d"}     len(a) =>4 cap(a) =>4

as := a[0:2]     0부터 시작해서 2 앞에까지 > a부터 c 앞까지 > a, b

slice 한 as의 값을 바꿔도 원래 잘라온 a도 값이 바뀜> 번지의 주소를 받았기 때문(포인터)

=>as와 a의 주소는 같지만 c는 다름(별도의 메모리를 더 할당받아서) 따로 append 받은거

배열도 slicing 가능

```
b := [4]int{4, 3, 2, 1}
```

bs := b[1:3]     1부터 3전까지니까 1, 2번인 >3과 2 출력

(slice와 배열의 차이 []안에 숫자 있냐 없냐)

```
for i,v := range s1{
```

```
    s2[i] = v
```

```
}
```

```
s2 := append([]int{}, s1[0], s1[1], s1[2])
```

s2 := append([]int{}, s1...) 반복문은 귀찮아서 s1의 slice를 s2에 복제함

7-2주차

func numbers(n1 int, n2 int) => 인수를 2개 받는 함수

numbers(5,10,3)하면 인수가 2개가 넘어서 에러가남 => 가변처리

func numbers(n1 int, n2 ...int) 뒤에 인수는 슬라이스로 동작함

numbers (1, 3, 5, 6) 1은 n1 n2는 [3,5,6]

numbers의 인수에 슬라이스는 들어갈 수 없음. int 여러개만

p가 슬라이스면 numbers(p) 는 불가능 primes(p...)는 가능 ...이 슬라이스임을 알려줌

for range를 이용하면 편하게 다룰 수 있음

```
p := []int{111, 19, 1, 17, 15}
```

```
idx := 2
```

p = append(p[:idx], p[idx+1:]...) => 슬라이스p에서 1을 자름 append이용



```
for k := idx + 1; k < len(p); k++ {
    p[k-1] = p[k] 5개중에 4번째를 3번째로 5번째를 4번째로 덮어줌
}
p = p[:len(p)-1] 맨마지막ㄱ 자르기
=>위에 슬라이스에서 p를 자르는 것과 같음 for문 이용
sort.Ints(p) 는 인트형 슬라이스인 p를 솔트함
Float면 sort.Float64s()
```

7-3주차 map

```
var games map[int]string  int가 키 value가 string
games = make(map[int]string)
혹은
games := make(map[int]string)
games[456] = "성기훈"
games[218] = "박해수"
games[067] = "강새벽"
games[001] = "오일남"
games[199] = "알리"
games[101] = "아이오아이"
```

```
//fmt.Println(games[067])
//    for _, v := range games {
//        fmt.Println(v)
//    }

//    games[101] = "장덕수" // update
//    delete(games, 199) // delete

//    for k, v := range games {
//        fmt.Println(k, v)
//    }
```

games := map[int]string{456: "성기훈", 218: "박해수", 67: "강새벽", 1: "오일남", 199: "알리", 101: "아이오아이"} 도 가능

제로값은 다른거랑 똑같이 string 는 널 나머지는 0

```

func status(name string) {
    balls := map[string]int{"성기훈": 20, "오일남": 0}
    //ball := balls[name]
    ball, exists := balls[name] 하는 이유 없는사람은 0개로 돼서
    불타입으로 존재 하는지 없는지
    if !exists {
        //fmt.Println(name, "님은 ", balls[name], "개로 탈락")
        fmt.Println(name, "님은 게임 참여자가 아닙니다")
    } else if ball < 1 {
        fmt.Println(name, "님은 ", balls[name], "개로 탈락")
    } else {
        fmt.Println(name, "님은 게임에서 승리하셨습니다")
    }
}

func main() {
    status("오일남")
    status("강철")
    status("성기훈")
}

```

5주차 파일

```
package main
```

```
import (  
    "fmt"  
    "log"  
    "os"  
)
```

```
func isPrime(n int) (bool, error) {  
    if n < 2 {  
        return false, fmt.Errorf("%d 는(은) 소수가 아닙니다~", n)  
    }  
  
    for i := 2; i < n; i++ {  
        if n%i == 0 {  
            return false, nil  
        }  
    }  
  
    return true, nil // 소수 판정 값, 정상데이터  
}
```

```
func prime(number int) {  
    p, err := isPrime(number)  
    if err != nil {  
        fmt.Println(err)  
        os.Exit(0)  
    }  
  
    if p {  
        fmt.Println(number, "는(은) 소수입니다!")  
    } else {  
        fmt.Println(number, "는(은) 소수가 아닙니다~")  
    }  
}
```

```
func primeRange(a int, b int) {  
    if a > b {  
        temp := a  
        a = b
```

```
        b = temp
    }

    for i := a; i <= b; i++ {
        p, err := isPrime(i)
        if err != nil {
            fmt.Println(err)
            os.Exit(0)
        }
        if p {
            fmt.Print(i, " ")
        }
    }
}
```

## 9주차

구조체 정의

```
type Student struct {  
    Name  string  
    Id    int  
    Score float64  
    Class string  
}
```

```
var s1 Student  
s1.Name = "홍길동"  
s1.Id = 12345678  
s1.Class = "A"  
s1.Score = 3.91
```

// 변수 선언과 동시에 구조체 정의

```
var s1 struct {  
    Name  string  
    Id    int  
    Score float64  
    Class string  
}  
s1.Name = "홍길동"  
s1.Id = 12345678  
s1.Class = "A"  
s1.Score = 3.91
```

```
/var s1 Student = Student{"홍길동", 12345678, 3.91, "A"}
```

```
var s1 Student = Student{  
//      "홍길동",  
//      12345678,  
//      3.91,  
//      "A",  
//  }   여러줄 할땐 마지막꺼 뒤에 쉼표!
```

```
var s2 Student = Student{  
//      Id:    12341234,  
//      Class: "B",  
//  }
```

```
// func printStudent(s Student) {
//     fmt.Println("성명 : ", s.Name)
//     fmt.Println("분반 : ", s.Class)
//     fmt.Println("학번 : ", s.Id)
//     fmt.Printf("평점 : %.2f\n", s.Score)
// }
```

student구조체 받아서 출력해주는 함수

```
// // func defaultStudent(id int) Student {
// //     var s Student
// //     s.Id = id
// //     s.Name = "noname"
// //     s.Class = "001"
// //     s.Score = 0.0
// //     return s
// // }
```

```
// func defaultStudent(s *Student) {
//     s.Id = 11112222
//     s.Name = "noname"
//     s.Class = "001"
//     s.Score = 0.0
// }
```

디폴트 설정 방법 두가지

```
type Students []Student
```

```
func (s Students) Len() int { return len(s) }
func (s Students) Less(i, j int) bool { return s[i].Score < s[j].Score } //부등호 바뀌면
내림차순
func (s Students) Swap(i, j int) { s[i], s[j] = s[j], s[i] }
```

```
s := []Student{
    {"홍길동", 12345678, 3.91, "A"},
    {"최길동", 12341234, 3.76, "B"},
    {"박길동", 11112222, 4.01, "C"},
}
```

```
sort.Sort(Students(s))
fmt.Println(s) //정렬 작은>큰순서
```

```

type ScholarshipStudent struct {
//      // Name      string
//      // Id        int
//      // Score      float64
//      // Class      string
//      StudentInfo Student
//      Level        int
//      Scholarship int
// }

// func main() {
//      var s1 ScholarshipStudent = ScholarshipStudent{
//          Student{"홍길동", 12345678, 3.91, "A"},
//          1,
//          1000000,
//      }

```

장학금 구조체

```

type ScholarshipStudent struct {
    //StudentInfo Student
    Student
    Level        int
    Scholarship int
}

```

```

func main() {
    s1 := ScholarshipStudent{
        Student{"홍길동", 12345678, 3.91, "A", 4},
        1,
        1000000,
    }
    fmt.Printf("장학금 수여 학생은 %s이고 장학금액은 %d원 입니다\n", s1.Name,
s1.Scholarship)
    fmt.Println(s1.Student.Level) //구조체 양쪽에서 겹치면 승격x표기o
(s1.StudentInfo.Id)//승격

```

## 10주차

사용자 정의 타입

```
type Liters float64
type Gallons float64

func ToLiters(g Gallons) Liters {
    return Liters(g * 3.785)
}

func ToGallons(l Liters) Gallons {
    return Gallons(l * 0.264)
}

func main() {
    //    carfuel := Liters(10)
    //    busfuel := Gallons(60)
    var carFuel Liters = 10
    var busFuel Gallons = 60

    carFuel = carFuel + ToLiters(Gallons(10.0))
    busFuel = busFuel + ToGallons(Liters(10.0))

    fmt.Println(carFuel, busFuel)
}
```

Go : 리시버(receiver) parameter

```
// func LitersToGallons(l Liters) Gallons {
//     return Gallons(l * 0.264)
// }

// func MillilitersToGallons(ml Milliliters) Gallons {
//     return Gallons(ml * 0.000264)
// } //비효율적
```



```
func (m MyType) sayHi(n int, f bool) { //메서드
//      fmt.Println(n, f)
//      fmt.Println("Test : " + m)
// }
```

```
// func main() {
//      var t1 = MyType("메서드 테스트 중")
//      t2 := MyType("단축 연산자도 사용가능")
//      t2.sayHi(9, true)
//      t1.sayHi(-7, false)
// }
```

```
func (m MyType) myReturn() int {
//      return len(m)
// } //메서드
fmt.Println(t2.myReturn())
```

```
type Number int
```

```
//func (n Number) Square() Number {
func (n *Number) Square() { // method
    *n = *n * *n
    //return n
}
```

```
func pointerSquare(n *int) { // function
    *n = *n * *n
}
```

```
func main() {
    n1 := Number(5)
    n2 := 4
    //fmt.Println(n1.Square())

    pointerSquare(&n2)
    fmt.Println(n2)

    fmt.Println(n1)
    n1.Square()
    fmt.Println(n1)
}
```

메서드 활용

```
// type Liters float64
// type Gallons float64
// type Milliliters float64

// func (g Gallons) ToLiters() Liters {
//     return Liters(g * 3.785)
// }

// func (g Gallons) ToMilliliters() Liters {
//     return Liters(g * 3785.41)
// }

// func (l Liters) ToGallons() Gallons {
//     return Gallons(l * 0.264)
// }

// func (ml Milliliters) ToGallons() Gallons {
//     return Gallons(ml * 0.000264)
// }

// func main() {
//     //coke := Liters(2)
//     var coke Liters = 2
//     fanta := Milliliters(750)

//     fmt.Printf("%.2f리터는 %0.3f갤론과 같습니다\n", coke, coke.ToGallons())
//     fmt.Printf("%.2f밀리리터는 %0.3f갤론과 같습니다\n", fanta, fanta.ToGallons())
// }
```

메서드 활용

```
import (
    "fmt"
    "time"
)

type student struct {
    id      int
    name    string
    subject string
    seconds int
}

func (s student) study() {
    fmt.Printf("%s은 %s과목을 공부합니다", s.name, s.subject)
}

func (s student) countDown() {
    for s.seconds > 0 {
        fmt.Println(s.seconds)
        //time.Sleep(time.Second * 2)
        time.Sleep(time.Second)
        s.seconds--
    }
}

func funcStudy(fs student) {
    fmt.Printf("%s은 %s과목을 공부합니다", fs.name, fs.subject)
}

func main() {
    s1 := student{1000, "홍길동", "Go", 10}
    //fmt.Println(s1)
    s1.countDown()
    s1.study() // method
    //funcStudy(s1) // function
}
```

10주차 과제 사용자 정의타입 map

```
package main
```

```
import "fmt"
```

```
//type game[number]name map[int]string
```

```
//type game[int]string map[int]string
```

```
/*type game struct{
```

```
    game map[int]string
```

단일변수타입만 보다가 두개 받아야 하는 맵을 사용자 정의 타입으로 만들려다가 많이 헤멤..

```
}*/
```

```
type game map[int]string //사용자 정의 타입 정의
```

```
// func (g game) Find(number int) {
```

```
//     for k, v := range g {
```

```
//         if number == k {
```

```
//             fmt.Printf("찾으시는 번호는 참가자 %s입니다.", v)
```

```
//             break
```

```
//         }
```

```
//     }
```

```
//     fmt.Println("찾는 참가자가 없습니다")
```

```
// } //찾는 번호를 매개변수로 받아 맵을 반복해서 돌려 찾는 번호와 같으면
```

```
//찾는 번호의 v값을 print해주고 break로 반복문을 빠져나오는 메소드
```

```
func (g game) Find(number int) string {
```

```
    for k, v := range g {
```

```
        if number == k {
```

```
            return v
```

```
        }
```

```
    }
```

```
    return "찾는 참가자가 없습니다"
```

```
} //리턴타입이 있는 메서드로도 만들어봄
```

```
func main() {
```

```
    m1 := game{456: "성기훈", 218: "박해수", 67: "강새벽", 1: "오일남", 199: "알리",  
101: "아이오아이"}
```

```
    // m1.Find(1) 리턴 없는 메서드
```

```
    //fmt.Println(m1.Find(1)) 오일남 나옴
```

```
    // fmt.Printf("찾으시는 번호는 참가자 %s입니다.", m1.Find(1))
```

```
    //fmt.Println(m1.Find(6)) 참가자 없다고 나옴
```

```

    var n1 int
    fmt.Print("참가자 번호 입력 : ")
    fmt.Scanln(&n1)
    fmt.Printf("찾으시는 번호는 참가자 %s입니다.", m1.Find(n1))
}

```

## 11주차

배운거 총 정리

package main

import "fmt"

```

func main() { //2,3차원 배열
    //anArray := [4]int{1, 2, 4, -4}
    //twoD := [4][4]int{{1,2,3,4},{5,6,7,8},{9,10,11,12},{13,14,15,16}}
    threeD := [2][2][2]int{{{1, 0},{-2, 4}},{5, -1},{7, 0}}}

    for i := 0; i<len(threeD); i++){
        v := threeD[i]
        for j := 0; j<len(v); j++){
            m := v[j]
            for k := 0; k<len(m); k++){
                fmt.Print(m[k], " ")
            }
            fmt.Println()
        }
        fmt.Println("\n")
    }
    //fmt.Println("Hello, 世界")
}

```

package main

import "fmt"

```

func main() {
    anArray := [4]int{1, 2, 4, -4}

    for _, v := range anArray {

```

```

        fmt.Printf("%d ", v)
    }
    //for k := 0; k<len(anArray); k++){
    //    fmt.Printf("%d ",anArray[k])
    //}
}

```

```
package main
```

```
import "fmt"
```

```

func main() {
    //var s1 []int
    //s1 = make([]int, 5)

    s1 := make([]int, 5)
    reSlice := s1[1:3]
    fmt.Println(s1)
    fmt.Println(reSlice)

    reSlice[0] = -100
    reSlice[1] = 123456
    fmt.Println(s1)
    fmt.Println(reSlice)

}

```

```
package main
```

```
import "fmt"
```

```

func printSlice(numbers []int) {
    for _, number := range numbers {
        fmt.Print(number, " ")
    }
    fmt.Println()
}

```

```
func main() {
```

```

aSlice := []int{-1, 0, 4, 100}

printSlice(aSlice)
fmt.Printf("수용량 : %d, 길이: %d\n", cap(aSlice), len(aSlice))
aSlice = append(aSlice, -99)
printSlice(aSlice)
fmt.Printf("수용량 : %d, 길이: %d\n", cap(aSlice), len(aSlice))
aSlice = append(aSlice, 1)
aSlice = append(aSlice, 2)
aSlice = append(aSlice, 3)
aSlice = append(aSlice, 4)
printSlice(aSlice)
fmt.Printf("수용량 : %d, 길이: %d\n", cap(aSlice), len(aSlice))

}

```

```

package main

import "fmt"

func plus(p1, p2, p3 *int) {
    *p3 = *p1 + *p2
}

func main() {
    a := 11
    b := 2
    c := 0

    plus(&a, &b, &c)
    fmt.Print(c)
}

```

```

package main

import "fmt"

type Soldier struct {
    Name string
}

```

```

    Id    int
    Grade string
}

func NewSoldier(name string, id int, grade string) *Soldier {
    // 여러분이 작성할 코드가 여기에 ...

    // Solution03
    return &Soldier{name, id, grade}

    // Solution 02
    //var a = Soldier{name, id, grade}
    //return &a

    // Solution 01
    //ns := new(Soldier)
    //ns.Name = name
    //ns.Id = id
    //ns.Grade = grade
    //return ns
}

func main() {
    var s1 = NewSoldier("성윤모", 1234, "이병")
    var s2 = NewSoldier("박민석", 1221, "일병")

    fmt.Println(s2.Name)
    fmt.Println(s2.Id)

    fmt.Println(s1.Name)
    fmt.Println(s1.Grade)
}

```



## 12주차

패키지 하다가 끝남

## 13주차

go 패키지 импорт 할때

импорт 하는 파일 > package 해당 폴더 잘 확인하기 (맨위에) 위치는 메인잇는 곳에 폴더 만들기

메인 파일 > import gowork / 메인 잇는 폴더 / импорт파일 폴더까지만

go mod init은 메인 파일 잇는 폴더 하나 위에서 go mod init 메인 잇는 폴더 위에까지 범위 잡기

메인 잇는 폴더에서 go build

13주차 오픈소스 프로그래밍

10주차에 한 구조체 안에 구조체 (승격)

같은 패키지 안에 있으면 import안해도 됨 go mod init 은 메인파일 잇는 폴더 위에

임베딩,캡슐화(게터,세터) event

인터페이스

//세터에 포인터 항상 조심하기

package mymath

```
type Date struct {  
    year  int //캡슐화  
    month int  
    day   int  
}
```

```
type Event struct {  
    Title int  
    Date  //구조체 속 구조체(임베딩)  
}
```

```
func (d *Date) SetYear(year int) { //세터
```

```

        d.year = year
    }
    func (d *Date) Year() int {
        return d.year
    }

    func (d *Date) SetMonth(month int) {
        d.month = month
    }

    func (d *Date) SetDay(day int) {
        d.day = day
    }

    func (d *Event) SetTitle(year int) {
        d.Title = year
    }

package main

import (
    "fmt"
    "gowork/packages/mymath"
)

func main() {
    date := mymath.Event{}
    date.Title = 1 //바로 접근 (캡슐화 안함)
    date.SetYear(1) //승격
    fmt.Println(date.Year()) //승격
}

```

### 13주차 인터페이스 계속

```

package gaget

import "fmt"

type TapePlayer struct {
    Batteries string
}

```

```
func (t TapePlayer) Play(song string) {
    fmt.Println(song, " 재생 중...")
}
```

```
func (t TapePlayer) Stop() {
    fmt.Println("중지")
}
```

```
type TapeRecorder struct {
    Microphones int
}
```

```
func (t TapeRecorder) Play(song string) {
    fmt.Println(song, " 재생 중...")
}
```

```
func (t TapeRecorder) Record() {
    fmt.Println("녹화 중...")
}
```

```
func (t TapeRecorder) Stop() {
    fmt.Println("중지")
}
```

```
package main
```

```
import "gowork/chap11/gaget"
```

```
type Player interface { //인터페이스 아래 두 함수를 갖고있는 구조체면 해당
    Play(string)
    Stop()
}
```

```
// func playList(device gaget.TapePlayer , songs []string){
func playList(device Player, songs []string) {
    for _, song := range songs {
        device.Play(song)
    }
    device.Stop()
}
```

```

func TryOut(player Player) {
    player.Play("데모트랙")
    player.Stop()
    // player.Record() player 인터페이스에 Record() 가 없음

    // recorder := gaget.TapeRecorder(player) 인터페이스는 형 변환 문법이 기존
언어와 다르다.
    // recorder.Record()

    recorder := player.(gaget.TapeRecorder) // 인터페이스 형 변환 문법
    recorder.Record()
}

func main() {
    TryOut(gaget.TapeRecorder{})

    // // player := gaget.TapePlayer{}
    // player := gaget.TapeRecorder{}
    // mixtape := []string{"난 알아요", "거짓말", "으르렁"}
    // playList(player, mixtape)
}

```

인터페이스 형변환(타입 단언)

var 변수1 인터페이스1 = 구조체()

var 변수2 구조체2 = 인터페이스변수1.(구조체2)

변수2. 변수2만 갖고있는 함수 부르기 가능 책 p335

14주차

고루틴과 채널 p381

```
package main
```

```
import (  
    "fmt"  
    "time"  
)
```

```
func a() {  
    for i := 0; i < 20; i++ {  
        fmt.Print("a")  
    }  
}
```

```
func b() {  
    for i := 0; i < 20; i++ {  
        fmt.Print("b")  
    }  
}
```

```
func c() {  
    fmt.Println("c")  
}
```

```
func d() {  
    fmt.Println("d")  
}
```

```
func main() {  
    go a()  
    go b() //a b 같이 실행  
    defer c() //디퍼는 다 실행하고 마지막 근데 2개있으면 LIFO여서 d먼저  
    defer d()  
    time.Sleep(time.Second)  
    fmt.Println("end")  
}
```

```
package main //웹페이지 불러오기
```

```
import (  
    "fmt"  
    "io/ioutil"  
    "log"  
    "net/http"  
)
```

```
func responseSize(url string, channel chan WebPage) {  
    //fmt.Println("URL 주소 ", url)  
    response, err := http.Get(url)  
    if err != nil {  
        log.Fatal(err)  
    }  
    defer response.Body.Close()  
    body, err := ioutil.ReadAll(response.Body)  
    if err != nil {  
        log.Fatal(err)  
    }  
    //fmt.Printf("%s", body)  
    //fmt.Println(len(body))  
    channel <- WebPage{URL: url, Size: len(body)}  
}
```

```
type WebPage struct {  
    URL string  
    Size int  
}
```

```
func main() {  
    pages := make(chan WebPage)  
    urls := []string{"https://www.inhatc.ac.kr", "https://www.naver.com",  
"https://www.daum.net", "https://www.nate.com"}  
  
    for _, url := range urls {  
        go responseSize(url, pages)  
    }  
    for i := 0; i < len(urls); i++ {  
        page := <-pages  
        fmt.Printf("%s: %d\n", page.URL, page.Size)
```

```

    }
    // go responseSize("https://www.inhatc.ac.kr")
    // go responseSize("https://www.naver.com")
    // go responseSize("https://www.daum.net")
    // go responseSize("https://www.nate.com")
    //response, err := http.Get("https://www.inhatc.ac.kr")
    //time.Sleep(time.Second * 5) //이거 안주면 너무 빨리끝나서 아무것도 안뜸
}

// package main

// import (
//     "fmt"
//     "io/ioutil"
//     "log"
//     "net/http"
//     "time"
// )

// func responseSize(url string) {
//     fmt.Println("URL 주소 ", url)
//     response, err := http.Get(url)
//     if err != nil {
//         log.Fatal(err)
//     }
//     defer response.Body.Close()
//     body, err := ioutil.ReadAll(response.Body)
//     if err != nil {
//         log.Fatal(err)
//     }
//     //fmt.Printf("%s", body)
//     fmt.Println(len(body))
// }

// func main() {
//     go responseSize("https://www.inhatc.ac.kr")
//     go responseSize("https://www.naver.com")
//     go responseSize("https://www.daum.net")
//     go responseSize("https://www.nate.com")
//     //response, err := http.Get("https://www.inhatc.ac.kr")

```

```
//      time.Sleep(time.Second * 5) //이거 안주면 너무 빨리끝나서 아무것도 안뜸  
  
// }
```

스님

숫자를 3개를 입력받은 후, 이 숫자 중에서 가장 큰 수, 가장 작은수를 출력하라.

```
package number;  
import java.util.*;  
  
public class Number {  
  
    public static void main(String[] args) {  
        Scanner s = new Scanner(System.in);  
  
        int array [] = new int [3];  
  
        int max_num = 0;  
        int min_num = 9999999;  
  
        for(int i = 0; i<3; i++)  
        {  
            System.out.printf("%d 번째 숫자를 입력하세요 : ", i + 1);  
            array[i] = s.nextInt();  
  
            if(array[i] > max_num)  
                max_num = array[i];  
  
            if( array[i] < min_num)  
                min_num = array[i];  
  
        }  
        System.out.printf("가장 큰 수는 %d 이고, 가장 작은 수는 %d 입니다.",  
max_num, min_num);  
  
    }  
}
```



```
}
```

아파트의 분양 면적을 제곱미터 단위로 입력받아 이값을 평형 단위를 값으로 변환하라.

```
package apt;
```

```
import java.util.*;
```

```
public class Apt {
```

```
    public static void main(String[] args) {
```

```
        Scanner s = new Scanner(System.in);
```

```
        System.out.print("아파트의 분양면적(제곱미터)을 입력하십시오. : ");
```

```
        double m2_area = s.nextDouble();
```

```
        double pyung_area = m2_area / 3.305;
```

```
        System.out.printf("아파트의 평형은 %1f 입니다.\n", pyung_area);
```

```
        if(pyung_area < 15)
```

```
            System.out.print("소형 아파트 입니다.");
```

```
        else if ((pyung_area >= 15) && (pyung_area < 30 ))
```

```
            System.out.print("중소형 아파트 입니다.");
```

```
        else if ((pyung_area >= 30) && (pyung_area < 50 ))
```

```
            System.out.print("중형 아파트 입니다.");
```

```
        else if (pyung_area >= 50)
```

```
            System.out.print("대형 아파트 입니다.");
```

```
        else
```

```
            System.out.print("잘못입력 하셨습니다.");
```

```
    }
```

```
}
```

날짜를 월과 일로 입력받아 이날짜는 1년중 몇 번째 날에 해당되는지 계산하여 출력하라.

```
package days;
import java.util.*;

public class Days {

    public static void main(String[] args) {

        Scanner s = new Scanner(System.in);
        System.out.print("월을 입력하시오. : ");
        int month = s.nextInt();

        System.out.print("일을 입력하시오. : ");
        int day = s.nextInt();
        int sum = 0;

        for (int i = 1; i < month; i++)
        {
            switch (i) {
                case 1:
                    sum += 31;
                    break;
                case 2:
                    sum += 28;
                    break;
                case 3:
                    sum += 31;
                    break;
                case 4:
                    sum += 30;
                    break;
                case 5:
                    sum += 31;
                    break;
                case 6:
                    sum += 30;
```

```
        break;
    case 7:
        sum += 31;
        break;
    case 8:
        sum += 31;
        break;
    case 9:
        sum += 30;
        break;
    case 10:
        sum += 31;
        break;
    case 11:
        sum += 30;
        break;
    case 12:
        sum += 31;
        break;
    default :
        System.out.print("잘못입력하셨습니다.");

    }
}
int day_count = sum + day;
System.out.printf("이날짜는 1년중 %d 번째 날에 해당 됩니다. ", day_count);
}
}
```

스 46

```
package git_class_homework;

import java.util.*;
import java.io.*;

public class ExamForWhile {

    public static void main(String[] args) throws IOException {

        while (true) {
            System.out.println("*****");
            System.out.println(" 반복문 연습하기1");
            System.out.println("*****");
            Scanner s = new Scanner(System.in);

            System.out.println("1. 1부터 입력한 수까지 더하기");
            System.out.println("2. 최대값/최소값구하기");
            System.out.println("3. 입력받은숫자의 합계와 평균구하기");
            System.out.println("4. 원하는 구구단 출력하기");
            System.out.println("5. 짝수단/홀수단 출력하기");
            System.out.println("6. 종료하기");
            System.out.print("원하는 메뉴는>> ");
            int menu = s.nextInt();

            switch (menu)
            {
                case 1:
                    new ExamForWhile.SumOfNumbers();
                    break;
                case 2:
                    new ExamForWhile.MaxAndMin();
                    break;
                case 3:
                    new ExamForWhile.SumAndAvg();
                    break;
                case 4:
                    new ExamForWhile.GuGuDan();
                    break;
                case 5:
                    new ExamForWhile.GuGuDan2();
```

```

        break;
    case 6:
        System.exit(0);
        break;
    default :
        System.out.print("잘못 입력하셨습니다.\n");
        break;
    }
}
}

```

```

private static class SumOfNumbers {

    public SumOfNumbers() {
        Scanner s = new Scanner(System.in);
        System.out.print("\n ###1부터 입력한 수까지 더하기");
        System.out.print("마지막 수를 입력(Q:종료) >> ");
        String sinput = s.next();

        if ((sinput.equals("Q") || (sinput.equals("q")))
            return;
        else {
            int input = Integer.parseInt(sinput);
            int sum = 0;

            for (int i = 0; i<= input; i++) {
                sum += i;
            }
            System.out.printf("총 합은 %d 입니다.\n\n", sum);
        }
    }
}

```

```

private static class MaxAndMin {

    public MaxAndMin() {
        System.out.print("\n###최대값/최소값구하기\n");

        Scanner s = new Scanner(System.in);
    }
}

```

```

int max = 0;
int min = 999999999;
int a [] = new int [5];

for (int i =0; i<a.length; i++)
{
    System.out.print("0~100사이의 숫자를 입력(Q:종료) >> ");
    String sinput = s.next();

    if ((sinput).equals("Q") || (sinput).equals("q"))
        return;
    else {
        a[i] = Integer.parseInt(sinput);
        if (a[i] > max) max = a[i];
        if (a[i] < min) min = a[i];
    }
}
System.out.printf("가장큰 수는 %d이고, 작은수는 %d입니다\n\n", max,
min);
}
}

```

```

private static class SumAndAvg {

    public SumAndAvg() {
        System.out.print("\n###입력받은숫자의 합계와 평균구하기\n");
        Scanner s = new Scanner(System.in);

        int total = 0, avg = 0;
        int a [] = new int [5];

        for (int i =0; i<a.length; i++)
        {
            System.out.print("0~100사이의 숫자를 입력(Q:종료) >> ");
            String sinput = s.next();

            if ((sinput).equals("Q") || (sinput).equals("q"))
                return;
            else {
                a[i] = Integer.parseInt(sinput);
                total += a[i];
            }
        }
    }
}

```

```

        }
    }
    avg = total / a.length;
    System.out.printf(" 합계는 %d 이고, 평균은 %d 입니다.\n\n", total, avg);
}
}

private static class GuGuDan {

    public GuGuDan() {
        System.out.print("\n###원하는 구구단 출력하기\n");

        Scanner s = new Scanner(System.in);
        System.out.print("출력하고 싶은 단(Q:종료) >> ");
        String sinput = s.next();

        int input = Integer.parseInt(sinput);

        if ((sinput).equals("Q") || (sinput).equals("q"))
            return;
        else {

            while(true){
                if(input == 1 || input == 10){
                    System.out.println("==> 잘못입력");
                    System.out.print("출력하고 싶은 단(Q:종료) >> ");
                    sinput = s.next();
                    input = Integer.parseInt(sinput);
                }
                else break;
            }

            for (int i = 1; i <=9; i ++){
                System.out.printf(" %d * %d = %d      ", input , i, input * i) ;
                if(i%2 == 0) System.out.print("\n");
                if(i == 9) System.out.print("\n\n");
            }

        }
    }
}

```

```

    }
}

private static class GuGuDan2 {

    public GuGuDan2() throws IOException {
        System.out.print("\n###짝수단/홀수단 출력하기\n");
        System.out.print("E:짝수단, O:홀수단(Q:종료) >> ");

        char s = (char)System.in.read();
        int num = 0;

        if(s == 'Q' || s == 'q' )
            return;
        if(s == 'E' || s == 'e' )
            num = 2;
        if(s == 'O' || s == 'o')
            num = 3;

        for (int j = 1; j<= 9; j++){
            for (int i = num; i<=9; i+= 2) {
                System.out.printf("%d * %d = %d\t", i , j, i * j);
            }
            System.out.println();
        }
        System.out.print("\n");
    }
}

}

```