

제출하기.

1. Class Main 2. Class 시작여부 ($\begin{matrix} + \\ - \\ + \end{matrix}$) 3. Class Save_Delete

o queue ~ 321
size) :
75 10 11 0 2 1 12
3 1 6 1 1 1 2 1

· 리아블한거

강다. enum. 제네릭

기능. 연산 기능. ~~재~~ 저장 기능. 출력 기능.

강다. 2의 변환화 + 매치변수로 함수를 쓸수있다. ~ 라만. 인터페이스가 있어야 한다

제네릭 ~ 매치변수를 쓰는 것에 간병화

enum ~ 상수 ~ 연산 기호.

기호코드

할라라라라라 ~ 이제

(PMI
PMI2)
공통된 것

해라해라해라 ~

new PMI

이게

(BA, 란다식)

→ 연산도. $A \odot B \Rightarrow result$
다 이리

배치정사시 매치변수에서

모수와 매치를 넣는다

해라 매치를 구현할때,

필요한 모든 것

배치를 매치변수에 넣으로 넣지 못하게

인터페이스를 활용한다

* 인터페이스의 매치는 매치변수로써
활용 가능하다

ex) 1~10까지

좌4 를 나누는 매치 구현시,
출4

확인할라하는

확인할라하는

1~10' 가 BA
매치

두가지가 매치



를 가지고, 실행 코드를 작성할라

new PMI 를 작성

강다. : "

클래스명 : PMI

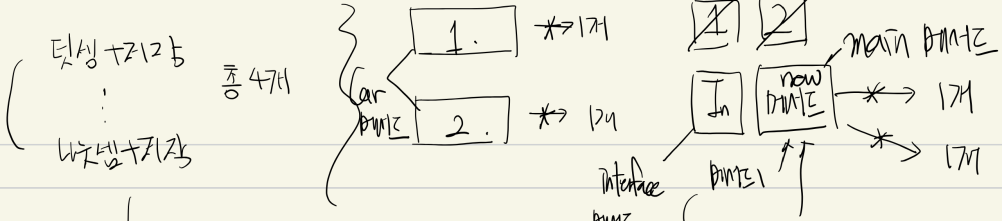
클래스 안에 매치

~ 다른 표현
(의명함)

(클래스명 매치) → return 실행문

실행문이 간단하다면,

'return' 과 'return' 을 생략할수있다



↓
불필요보임!

✓ 메소드명 (BA, interface명)

boolean 새로이생성함 < BA

✓ 인터페이스 만들고,
↳ 제네릭.

◦ 제네릭 쓰는 방법이보자.
↳ obable

Same
log 들어.

BA 메소드. 강제성
(정해진 결과 값)
log: : 조리개

◦ 저장 값에 결과 값이 들어가는 메소드 만들어야함

• try - catch - finally ~ 입력 값에
바뀌는 '0'

◦ map ~ 키 값과 값이 안맞아서 X ~ 데이터는 두자리 값이 필요하겠지만

• nextLine(); & ~~가~~ ~~있~~ ~~음~~ 4는 → 문자로 해석해서 오류가 없다.

```
public static boolean hasTicket (Car car) {  
    return car.hasParkingTicket;  
}
```

→ 자식 클래스 (BA, ...)
↑
가져오는 name - parkcars

```
public List<Car> parkcars (List<Car> carsWant,  
                           predraet<Car>fn  
                           알려짐) {  
    List<Car> cars = new ArrayList<>();  
    for (Car cars : carsWantToPark) {  
        if (function.test(cars)) {  
            cars.add(cars);  
        }  
    }  
    return cars;  
}
```

인스턴스

```
interface Predicate<T> { boolean test(T t); }
```