

## Quiz. SumAvg.java (class)

크기 5의 배열의 값을 입력(스캐너) 받아서 총점과 평균을 구하는 메서드를 만들어 호출해 보세요.

출력:

정수 5개를 입력하세요.

25

63

48

95

21

입력된 배열의 합은: 252

입력된 배열의 평균은 : 50

## Quiz. maxValuе.java (class)

주어진 배열을 입력 받아서 가장 큰 수를 반환하는 메서드를 만들어 보세요.

입력 : {78,54,89,57,84,95,74,91,84,67,52,94,82}

출력:

최대값 : 95

## Quiz. CharChange.java (class)

char[] 배열을 전달받아 배열 속의 공백(' ') 문자를 ','로  
대치하는 메소드(replace())를 작성하고 출력하는 메소드  
(ptr()) 작성해서 예시처럼 호출하세요.

입력 :

{'T','h','i','s',' ','i','s',' ','a',' ','p','e','n','c','i','l','.'}

출력:

This is a pencil.  
This,is,a,pencil.

## Quiz. AddTwoArr.java (class)

주어진 두 배열의 같은 위치의 숫자를 합쳐서 하나의 배열로 만들고 (`int[] arrAdd(int[] a, int[] b){}`) 출력하는(`void prt(int[] a){}`) 메서드를 만드세요.  
단, 같은 위치에 숫자가 없다면 해당 위치의 숫자를 그대로 출력하세요.

입력 : {78,54,89,57,84}  
      {95,74,91,84}

출력:

173, 128, 180, 141, 84,

# Quiz. Tv.java, TvEx.java (class)

다음 main() 메소드를 실행하였을 때 예시와 같이 출력되도록 TV 클래스를 작성하세요. brand, year, inch 필드, 생성자, show()메서드 구성.

**출력:**

LG에서 만든 2017년형 32인치 TV

# Quiz. Circle. Java, CircleArray.java (class)

Circle클래스로 객체 5개를 가지는 배열을 생성하고, Circle 객체의 반지름(매개변수)을 0에서 4까지 각각 지정한 후, 면적을 출력하세요.

$$\text{면적} = 3.14 * \text{반지름} * \text{반지름}$$

출력:

```
반지름0 원의 넓이 : 0  
반지름1 원의 넓이 : 3  
반지름2 원의 넓이 : 12  
반지름3 원의 넓이 : 28  
반지름4 원의 넓이 : 50
```

# Quiz. Card.java (class)

main메소드의 내용은 다음과 같다. 결과가 나오도록 info()를 작성하시오.

- ❖ Card 클래스에서 생성자 매개변수로 받아온 boolean 타입의 값을 if문으로 확인해서 true면 문자열 "k"와 결합, false면 숫자 그대로 출력
- ❖ Card 클래스의 필드 int num =1; boolean chk=true

```
public static void main(String[] args){  
    Card card1 = new Card(3,false);  
    Card card2 = new Card(2,true);  
  
    System.out.println(card1.info());  
    System.out.println(card2.info());  
}
```

출력:

```
3  
2K
```

# Quiz. Grade.java (class)

3 과목의 점수를 scanner로 입력 받아 Grade 객체를 생성하고 성적 평균을 출력하는 Grade 클래스를 만드시오.

Grade 클래스에 int 타입의 math, science, english 필드를 private로 선언하고, 생성자와 세 과목의 평균을 리턴하는 average() 메소드를 작성합니다.

**출력:**

```
수학, 과학, 영어 순으로 3개의 정수 입력 >> 89  
96  
80  
평균은 88
```



## Quiz. MyTv.java (class) - 캡슐화

다음 클래스의 멤버 필드 isPowerOn, channel, volume를 클래스 외부에서 접근할 수 없도록 제어자를 붙이고 대신 이 멤버 변수들의 값을 어디서나 읽고 변경할 수 있도록 getter메소드와 setter메소드를 추가하세요. 실행클래스 없음.

```
public class MyTv {  
    boolean isPowerOn;  
    int channel;  
    int volume;  
  
}
```

## Quiz. SingletonEx.java (class) - 싱글톤

다음 클래스에서 생성자가 private 접근 제어가 있을 때 이 클래스의 인스턴스를 사용할 수 있도록 getInstance메소드를 작성하세요.

```
public class Singleton {  
    private Singleton(){}  
    private int number;  
    public int getNumber() {  
        return number;  
    }  
    public void setNumber(int number) {  
        this.number = number;  
    }  
}
```

```
public static void main(String[] args) {  
    Singleton s = Singleton.getInstance();  
  
    s.setNumber(100);  
    System.out.println(s.getNumber());  
}
```

# Quiz. SongEx.java (class)

노래 한 곡을 나타내는 Song 클래스를 작성하세요. 또한 Song 클래스에 다음 생성자와 메소드를 작성하세요.

- 노래의 제목을 나타내는 title
  - 가수를 나타내는 artist
  - 노래가 발표된 연도를 나타내는 year
  - 국적을 나타내는 country
- 생성자 2개: 기본 생성자와 매개변수로 모든 필드를 초기화하는 생성자  
`this("title","artist",0000,"country");`
  - 노래 정보를 출력하는 show() 메소드

출력:

1978년 스웨덴국적의 ABBA가 부른 Dancing Queen

# Quiz. StudentEx.java (class)

오른쪽 테이블과 같은 멤버 필드를 가지는 Student(라이브러리)클래스를 작성하세요.

타입	필드명	설명
String	name	학생 이름
int	ban	반
int	no	번호
int	kor	국어점수
int	eng	영어점수
int	math	수학점수

Student클래스에 다음 내용이 포함된 3 개의 메소드를 작성하세요.

메소드 명	반환타입	매개변수	기능
getTotal	int	없음	국어,영어 수학 점수를 모두 더해서 반환한다.
getAverage	float	없음	총점을 과목수로 나눈 평균을 반환한다. <del>(추가 : 소수점 둘째 자리에서 반올림 해 보세요.)</del>
Info	void	없음	모든 필드와 메서드를 호출한다.

출력:

이름 : 고길동  
반 : 3  
번호 : 17  
국어점수 : 92  
영어점수 : 63  
수학점수 : 74  
총점 : 229  
평균 : 76.3

## StudentEx.java

```
public static void main(String[] args){  
    Student student3 = new Student("고길동", 3, 17, 92, 63, 74);  
    student3.info();  
}
```

# Quiz. BankEx.java (class)

1. Bank.java 라이브러리 클래스에 예금 기능/ 출금 기능/ 조회 기능을 각각 메서드를 만들어 봅니다.
2. 실행 클래스의 메인 메서드에서 종료(메서드)를 선택하지 않으면 메뉴가 반복적으로 출력되도록 하세요.
3. 각각의 메뉴 선택시 메서드를 통해서 해당 기능을 구현하세요
4. 출금기능은 잔액보다 큰 액수는 출금할 수 없습니다.
5. 잔액이 1,000,000원이 넘어서도록 입금할 수 없습니다.

예 ) 잔액 900,000원 있는데 110,000원을 입금하면 잔액이 1,010,000원으로 1,000,000원을 초과하므로 입금을 거부해야 합니다

안녕하세요 G BANK입니다.

\*\*\*\*\*

그린 은행에 오신것을 환영합니다.

메뉴를 선택해 주세요.

1. 예금
2. 출금
3. 잔액조회
4. 종료

>1

예금할 금액을 입력해주세요.

860000

860000원을 입금하였습니다.

총 금액은: 860000입니다.

\*\*\*\*\*

그린 은행에 오신것을 환영합니다.

메뉴를 선택해 주세요.

1. 예금
2. 출금
3. 잔액조회
4. 종료

>2

출금할 금액을 입력해주세요.

65000

65000원을 출금하였습니다.

총 금액은: 795000입니다.

\*\*\*\*\*

그린 은행에 오신것을 환영합니다.

메뉴를 선택해 주세요.

1. 예금
2. 출금
3. 잔액조회
4. 종료

>3

잔액은 795000원입니다.

\*\*\*\*\*

그린 은행에 오신것을 환영합니다.

메뉴를 선택해 주세요.

1. 예금
2. 출금
3. 잔액조회
4. 종료

>4

프로그램을 종료합니다.

\*\*\*\*\*

그린 은행에 오신것을 환영합니다.

메뉴를 선택해 주세요.

1. 예금
2. 출금
3. 잔액조회
4. 종료

>5

잘못 입력하였습니다.

1~4 값 중 선택해주세요.

# Quiz. MonthScheduleEx.java (class)

하루의 할 일을 표현하는 클래스 Day는 다음과 같다.

```
class Day {  
    private String work; //하루의 할 일을 나타내는 문자열  
    public void set(String work) {  
        this.work = work;  
    }  
    public String get() { return work; } //getter  
    public void show() {  
        if(work == null)  
            System.out.println("없습니다.");  
        else  
            System.out.println(work+"입니다.");  
    }  
}
```

한 달의 할 일을 표현하는 MonthSchedule 클래스를 작성하세요. Day클래스의 객체 배열과 적절한 필드, 메소드를 작성하고 실행 예시처럼 입력, 보기, 끝내기 등의 3개의 기능을 작성하세요.

[Hint]

MonthSchedule에는 생성자, input(), view(), finish(), run() 메소드를 만들고 run()에서 메뉴를 출력하고 처리합니다. **MonthScheduleEx.java** 에서 다음과 같이 호출 실행

```
MonthSchedule april = new MonthSchedule(30);  
// 4월 달 할 일  
april.run();
```

출력예시:

```
이번달 스케줄 관리 프로그램.  
할일(입력:1, 보기:2, 끝내기:3) >>1  
날짜(1~30)?5  
할일(빈칸없이입력)?공부하기  
할일(입력:1, 보기:2, 끝내기:3) >>2  
날짜(1~30)?5  
5일의 할 일은 공부하기입니다.  
할일(입력:1, 보기:2, 끝내기:3) >>3  
프로그램을 종료합니다.
```

```
MonthSchedule - Windows 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

package Exercise.Scheduler;
import java.util.Scanner;

public class MonthSchedule {
    private Scanner sc = new Scanner(System.in, "EUC-KR");
    private Day days[];

    MonthSchedule(int day) {
        // this.days = days[];
        for(int i=0; i<days.length; i++) {
            days[i] = new Day();
        }
        sc = new Scanner(System.in, "EUC-KR");
    }
    private void input() {
        System.out.print("날짜(1~30)?");
        int day = sc.nextInt();
        System.out.print("할일(빈칸없이입력)?");
        String work = sc.next();
        day--; //index num 맞추기 위해
        if(day < 0 || day > 30) {
            System.out.println("날짜를 잘못 입력하였습니다.");
            return;
        }
        days[day].set(work);
    }
    private void view() {
        System.out.print("날짜(1~30)?");
        int day = sc.nextInt();
    }
}
```

객체지향에서는 외부클래스를 타  
입으로 필드 선언이 가능

```
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

package Exercise.Scheduler;

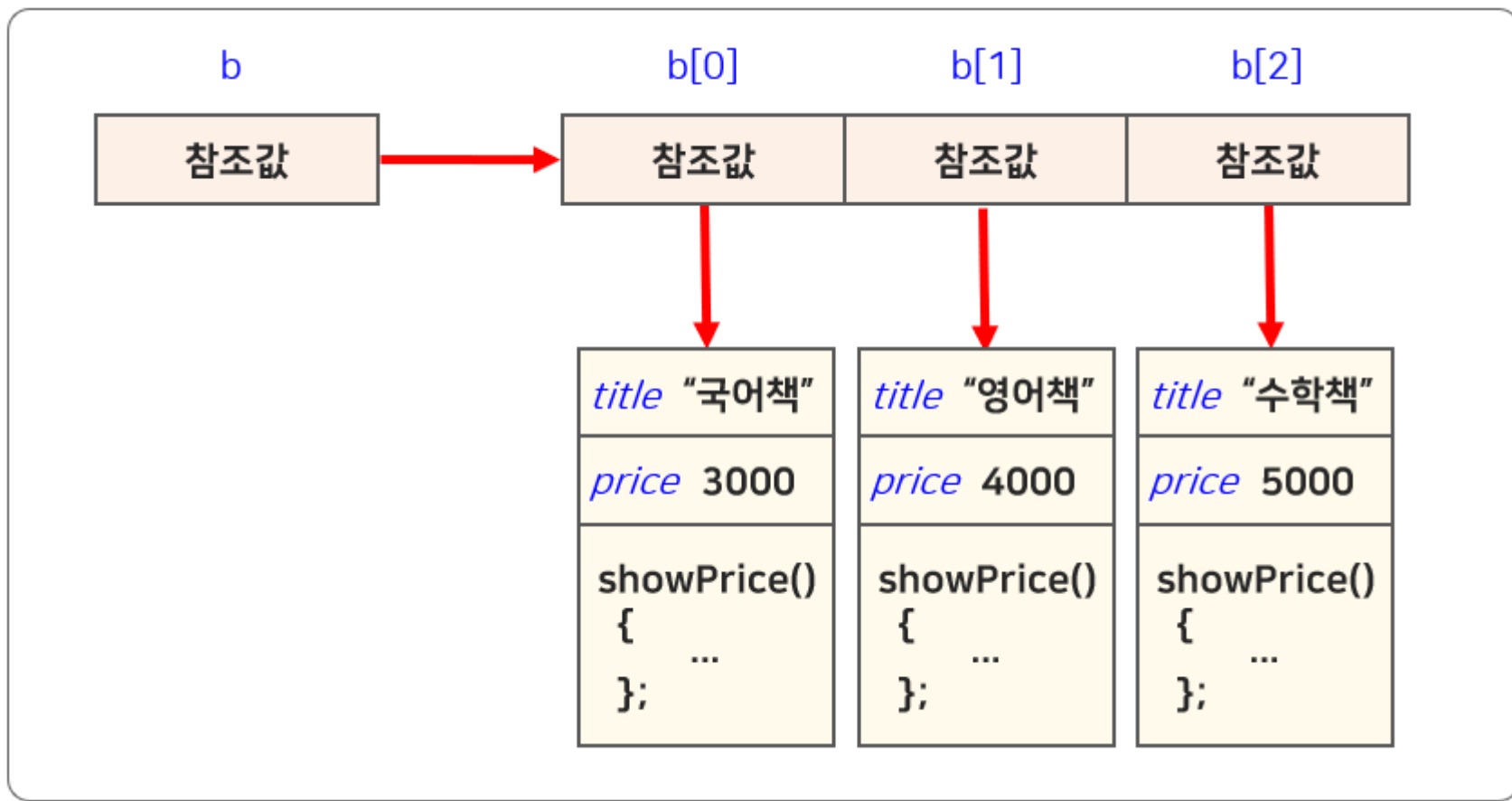
public class Day { // 하루의 할 일을 셋팅
    private String work; //하루의 할 일을 나타내는 문자열

    public void set(String work) {
        this.work = work;
    }

    public String get() { return work; } // getter

    public void show() {
        if(work == null) System.out.println("없습니다.");
        else System.out.println(work+"입니다.");
    }
}
```

## \* 배열로 객체 생성





# Quiz. AccountEx.java (class)

1. 예금주와 계좌번호, 초기 예금액을 입력 받아서 예금 계좌를 개설한다.
2. 계좌 정보는 총 100명분의 정보만 저장 가능하도록 합니다.
3. 은행업무를 하고 싶다면 계좌번호를 입력해서 있다면 업무를 볼 수 있고, 없다면 해당 계좌가 없다고 출력하세요.
4. 은행 업무(메소드 생성)에 조회 시 해당 계좌의 잔액을 조회 할 수 있고, 입금과 출금 기능을 이용할 수 있습니다.
5. 출금 시 잔액 보다 많은 금액을 출금할 수 없습니다.

## 1단계: [Account클래스]

필드 : 예금주, 계좌번호, 잔고

메서드 : 예금을 입금하는 메서드, 예금을 출금하는 메서드

생성자 : 예금주, 계좌번호, 초기 예금액을 입력 받아서 인스턴스 생성하는 생성자

-----

## 2단계 : [AccountEx클래스] main 메서드

메뉴로 계좌 생성기능, 은행업무 기능, 종료기능 작동시키는 while~ switch문을 완성

-----

## 3단계 [AccountEx클래스] Account를 담는 배열타입의 static변수를 생성

3-1단계 : 계좌 생성: 3개의 정보를 입력 받고 인스턴스를 생성한 후 위 변수 순서대로에 담는다.

3-2단계 : 은행 업무 : 계좌 번호를 입력 요구하고 성공하면 입금 출금, 잔액 조회 기능을 할 수 있다.

MIT 은행에 오신것을 환영합니다.  
메뉴를 선택해 주세요.

1. 계좌 생성
2. 계좌 업무
3. 종료

선택>1

계좌를 생성합니다.

예금주 이름을 입력하세요.

>이정은

계좌번호를 입력하세요.

>12345

초기 입금액을 입력하세요.

>50000

계좌 생성

MIT 은행에 오신것을 환영합니다.  
메뉴를 선택해 주세요.

1. 계좌 생성
2. 계좌 업무
3. 종료

선택>1

계좌를 생성합니다.

예금주 이름을 입력하세요.

>홍길동

계좌번호를 입력하세요.

>123456

초기 입금액을 입력하세요.

>70000

계좌 생성

MIT 은행에 오신것을 환영합니다.  
메뉴를 선택해 주세요.

1. 계좌 생성
2. 계좌 업무
3. 종료

선택>2

사용하고자 하는 계좌 번호를 입력하세요.

>123456

홍길동님 환영합니다.

업무를 선택하세요.

1. 입금
2. 출금
3. 잔액 조회
0. 메인메뉴 돌아가기

>1

입금 금액을 입력하세요.

60000

60000원 입금합니다.

입금완료

홍길동님 환영합니다.

업무를 선택하세요.

1. 입금
2. 출금
3. 잔액 조회
0. 메인메뉴 돌아가기

>2

출금 금액을 입력하세요.

30000

30000원 출금합니다.

출금 완료

홍길동님 환영합니다.

업무를 선택하세요.

1. 입금
2. 출금
3. 잔액 조회
0. 메인메뉴 돌아가기

>3

잔액을 조회합니다.

잔액 : 100000

홍길동님 환영합니다.

업무를 선택하세요.

1. 입금
2. 출금
3. 잔액 조회
0. 메인메뉴 돌아가기

>0

메인 메뉴로 돌아갑니다.

MIT 은행에 오신것을 환영합니다.  
메뉴를 선택해 주세요.

1. 계좌 생성
2. 계좌 업무
3. 종료

선택>3

프로그램 종료합니다.

# Quiz. LoginSys.java (class)

## 회원 가입과 로그인 시스템

로그인 성공 시 이름과 전화번호를 출력하는 코드를 작성해보자. 멤버 정보는 총 100명분의 정보만 저장 가능하도록 합니다.

LoginSys.java

```
public static Member[] members = new Member[100];
public static Scanner scan
= new Scanner(System.in);
// String id = LoginSys.scan.next(); public 식으로 다른 클래스에서
사용 가능.
static void main(String[] args) {
    // 로그인 프로그램
    new Controller();
}
```

## 1단계: [Member 클래스]

필드 : 아이디, 비밀번호, 이름, 전화번호

메서드(prt()) : 이름과 전화번호를 출력하는 메서드

생성자 : 아이디, 비밀번호, 이름, 전화번호를 입력 받아서

인스턴스 생성하는 생성자

---

## 2단계 : [view 클래스] 화면에 표시하는 메서드들 생성

menu() : 메뉴화면 표시 후 스캐너로 메뉴 넘버 리턴

userfault() : 메뉴 넘버 잘못 입력시 오류 메시지 호출

signUp() : 아이디, 비밀번호, 이름, 전화번호를 입력 받아서

Member 클래스의 객체로 등록.

LogIn() : 아이디, 비밀번호를 입력 받아서 account배열에 등록

success() : 로그인 성공 메시지와 prt()메서드 실행.

deadline() : 회원가입시 100명이 넘으면 메시지 호출.

"회원가입이 마감되었습니다." "나중에 다시 찾아와 주십시오"

programExit() : "y" or "Y" 입력 시 종료

"n" or "N" 입력 시 종료취소.

userfault() : 메뉴 선택 예외 처리

---

## 3단계 : [Controller 클래스]

메뉴로 회원가입기능, 로그인 기능, 종료 기능을 작동시키는 while~ switch문을 완성시켜보자.

3-1단계 : signUp() 메서드 이용하여 회원 가입 진행. 100명이 넘으면 deadline() 메서드 호출

3-2단계 : LogIn() 메서드 이용하여 변수에 담긴 인스턴스의 아이디와 비밀번호를 비교해서 로그인 처리를 진행.

3-3단계 : programExit() 메서드를 이용하여 프로그램 종료.

3-4단계 : userfault() 로 입력 예외 처리

MIT 사이트에 오신 것을 환영합니다.  
메뉴를 선택해 주세요.  
1. 회원가입  
2. 로그인  
3. 프로그램 종료  
선택>4  
잘못 입력했습니다.  
다시 확인해주세요.  
MIT 사이트에 오신 것을 환영합니다.  
메뉴를 선택해 주세요.  
1. 회원가입  
2. 로그인  
3. 프로그램 종료  
선택>1  
회원 가입을 합니다.  
아이디를 입력하세요  
lee  
비밀번호를 입력하세요  
1234  
이름을 입력하세요  
정은  
전화번호를 입력하세요  
000-0000

MIT 사이트에 오신 것을 환영합니다.  
메뉴를 선택해 주세요.  
1. 회원가입  
2. 로그인  
3. 프로그램 종료  
선택>2  
로그인합니다.  
아이디를 입력하세요  
park  
비밀번호를 입력하세요  
1234  
로그인에 실패했습니다.  
아이디 또는 비밀번호를  
확인해 주세요.

MIT 사이트에 오신 것을 환영합니다.  
메뉴를 선택해 주세요.  
1. 회원가입  
2. 로그인  
3. 프로그램 종료  
선택>2  
로그인합니다.  
아이디를 입력하세요  
lee  
비밀번호를 입력하세요  
1234  
1번째 회원  
로그인에 성공했습니다.  
사용자 이름: 정은  
사용자 폰번호: 000-0000

MIT 사이트에 오신 것을 환영합니다.  
메뉴를 선택해 주세요.  
1. 회원가입  
2. 로그인  
3. 프로그램 종료  
선택>3  
^^  
종료하시겠습니까? (y/n)  
y  
프로그램을 종료합니다.

# Quiz. BoardSys.java (class)

## 게시판 시스템

게시판 작성 프로그램을 만들어 보자  
글은 최대 50개 까지 작성

BoardSys.java

```
public static Scanner scan = new
Scanner(System.in, "EUC-kr");
public static Board[] boards = new
Board[50];
public static void main(String[] args) {
    // 게시판
    Controller a = new Controller();
    a.init();
    scan.close();
}
```

## 1단계: [Board 클래스]

필드 : 제목, 작성자, 내용, 비밀번호

메서드(prt()) : 제목, 작성자, 내용을 출력

생성자 : 제목, 작성자, 내용, 비밀번호를 입력 받아서 생성  
자 구성

---

## 2단계 : [Controller 클래스]

기본 메뉴를 구성. 게시 보기, 글 작성, 글 삭제로 구성

- 게시글 보기는 우선 제목을 번호 매겨서 보여주고 해당 번호 입력을 받아서 게시글의 제목, 작성자, 글 내용을 출력하도록 한다. 출력 후 목록으로 돌아간다.
  - 글 작성은 제목, 작성자, 비밀번호, 내용 작성해서 저장합니다.
- 

## 3단계 [view 클래스] 화면에 표시하는 메서드들 생성

menu() : 메뉴화면 표시 후 스캐너로 메뉴 넘버 리턴

list() : 게시 목록 호출

viewContent() : 게시글 번호 스캐너로 받아서 리턴

createBoard() : 제목, 작성자, 내용비밀, 번호 입력받아서

**Board** 객체로 생성하여 리턴

환영합니다 ^^

1.게시 보기 2.글 작성 3.종료  
번호를 입력해주세요

>

2

게시물을 작성합니다.  
작성자를 입력하세요>

정은

제목을 입력하세요>

게시판입니다.

글 내용을 입력하세요.

내용입니다.

삭제나 수정시 사용할 비밀번호를 입력하세요.

1234

글 등록 성공

환영합니다 ^^

1.게시 보기 2.글 작성 3.종료  
번호를 입력해주세요

>

1

\*\*\*리스트 목록을 확인하세요\*\*\*\*

1번째 게시글: 게시판입니다.

보고싶은 게시글의 번호를 선택하세요.

(0번 입력시 메인 메뉴로 돌아갑니다.)

1

제목: 게시판입니다.

작성자: 정은

내용: 내용입니다.

1.수정 | 2.삭제 | 3.돌아가기

선택>3

환영합니다 ^^

1.게시 보기 2.글 작성 3.종료  
번호를 입력해주세요

>

# Quiz. BoardSys.java (class)

## 게시판 시스템

게시판 작성 프로그램을 만들어 보자  
글은 최대 50개 까지 작성

BoardSys.java

```
public static Scanner scan = new
Scanner(System.in, "EUC-kr");
public static Board[] boards = new
Board[50];
public static void main(String[] args) {
    // 게시판
    Controller a = new Controller();
    a.init();
    scan.close();
}
```

## 1단계: [Board 클래스]

필드 : 제목, 작성자, 내용, 비밀번호

메서드(prt()) : 제목, 작성자, 내용을 출력

생성자 : 제목, 작성자, 내용, 비밀번호를 입력 받아서 생성  
자 구성

---

## 2단계 : [Controller 클래스]

기본 메뉴를 구성. 게시 보기, 글 작성, 글 삭제로 구성

- 게시글 보기는 우선 제목을 번호 매겨서 보여주고 해당  
번호 입력을 받아서 게시글의 제목, 작성자, 글 내  
용을 출력하도록 한다. 출력 후 목록으로 돌아간다.

- 글 작성은 제목, 작성자, 비밀번호, 내용 작성해서 저장합  
니다.

- 글 삭제는 글 목록을 보여주고 해당 번호와 작성자와 비  
밀번호를 받아서 일치하면 해당 글을 삭제 합니다.

---

## 3단계 [view 클래스] 화면에 표시하는 메서드들 생성

menu() : 메뉴화면 표시 후 스캐너로 메뉴 넘버 리턴

list() : 게시 목록 호출

viewContent() : 게시글 번호 스캐너로 받아서 리턴

createBoard() : 제목, 작성자, 내용비밀, 번호 입력받아서

**Board** 객체로 생성하여 리턴

[추가] 글 수정 기능도 추가해봅니다.

비밀번호를 입력 해야지 수정이 가능하도록 만들어 봅니다.