

파일입출력-java.io.*

클라이언트 / 서버

ip / port

java.net.InetAddress

tcp 네트워크

java.net. tcp 프로토콜

udp 프로토콜

프로토콜 – protocol

A-B 컴퓨터 약속규칙

서버-클라이언트 동일 프로토콜 사용

- tcp 프로토콜

1> 전화-연결

서버 출력 – 클라이언트 입력

클라이언트 출력 – 서버 입력.....

2>연결해제

java.net.ServerSocket

java.net.Socket

tcp server 클래스	tcp client 클래스
<p>1> ServerSocket 생성 (port=9999)</p> <p>3> tcp 연결 클라이언트 대기- 승인- Socket 리턴</p> <p>5> 클라이언트로부터 데이터 전 송받는다=서버 입력</p> <p>6> 처리 결과를 클라이언트로 전송한다=서버 출력</p> <p>9> Socket close = 연결해제</p> <p>10> 또다른 클라이언트 연결 요청 대기 승인</p>	<p>2> Socket 생성=서버 연결 ("tcp server 컴퓨터ip", 9999)</p> <p>4> 서버로 데이터 전송하다= 클라이언트 출력</p> <p>7> 서버로부터 데이터 전송받는다= 클라이언트 입력</p> <p>8> Socket close=서버 연결해제</p>

java.io.DataInputStream ---> DataOutputStream 출력된 데이터를 입력받는 객체	= java.util.Scanner 동일
java.io.DataOutputStream ---> 출력시 자바데이터타입 변환	

MemberServer 서버 1>클라이언트로부터 아이디와 암호 전송받는다 2>클라이언트로부터 아이디는 String ids []= {"java", "jsp", "spring"}; 중복체크한다. 3> 클라이언트에게 전송한다 중복된 아이디라면 "다른 아이디를 사용하세요" 전송한다 중복되지 않았다면 클라이언트 "xxxx 는 가능합니다" 전송한다.	MemberClient 클라이언트 1> 회원아이디입력: XXXX (키보드) 2> 암호입력 : XXX(키보드) 3> 서버로 전송한다 아이디 암호 4>서버로부터 결과를 입력받는다
---	---

- 시험

오늘 5시부터 - 8시까지 시험 진행

러닝클라우드 사이트 접속

5-6시사이 줌 켜놓고 시험 시작

5시55분 qr 게시 공시

6시이후 줌 끄고 나머지 시험 진행

8시까지 종료

시험 = 100/60 이상(12문제이상)

과목 점수 = 출석 + 과제 + 시험 =100

-과제

3월 23일 과제

러닝클라우드 사이트 접속

클래스명 파일명 압축파일명

과제1 압축

과제2 압축(과제1 수정 변형)

==> 전체압축

- 비정형 매개변수

```
class A{  
    int add(int i, int j){return i+j;}  
    String add(String i, String j){return i+j;}  
    void add(double i, double j){ }  
    void add(Object o1, Object o2){ o1.toString()+o2.toString() ; }  
    void add(Object o1){ o1.toString() ; }  
    void add(){ }
```

오류--> int add(){ return 0; }

오류--> void add(int a, int b){ a+b; }

}==> 메소드 overloading

(1 개 클래스/ 같은 이름 메소드 여러개

/ 매개변수(갯수나 타입이나 순서) 다르게 정의 / 리턴타입이나 modifier 상관없다)

main

A a1 = new A();

a1.add(1,2)->정수3

a1.add("jj","aa")->string"jjaa"

a1.add(3.14, 2.2)->실수5.34

==> add 이름으로 3가지 타입 모두 add

강사님 순서가 다르다는거는 매개변수가 String int 인거랑 int String 말하는건가

요 ?

```
void add(int i, String s){ }
```

```
void add(String i, int s){ }
```

===== 비정형 매개변수 =====

main

```
A a1 = new A();
```

```
a1.add(1,2)->정수 2
```

```
a1.add(1,2,3)->정수 5
```

```
a1.add(10, 1000, 10000, 347634, 12, 34, 243)->정수 7
```

```
a1.add(정수 타입 정하고 / 갯수 정하지 않고)
```

==> 비정형 매개변수 형태

```
class A{
```

```
    add( int ... numbers){
```

```
        numbers==> int[] 타입 간주
```

```
    }
```

```
}
```

시험 과제 대비 정리

MenuTest

1. 전체 메뉴

2. 세부 메뉴 실행 입력과 처리 데이터등 -main 메소드 구현

==>

사원등록

사번

이름

급여

직급

부서

==> Employee 변수 추가

==> 파일 저장

코드 복잡 메뉴 증가 - 수정 위치 어디 분석, 수정 코드 추가

- singleton dao dto

DAO = DATA ACCESS OBJECT 클래스

데이터 직접 접근 객체 = 파일 입출력/ DB 입출력

DTO = DATA TRANSFER OBJECT 클래스

VO = VALUE OBJECT 클래스

DO = DATA OBJECT 클래스

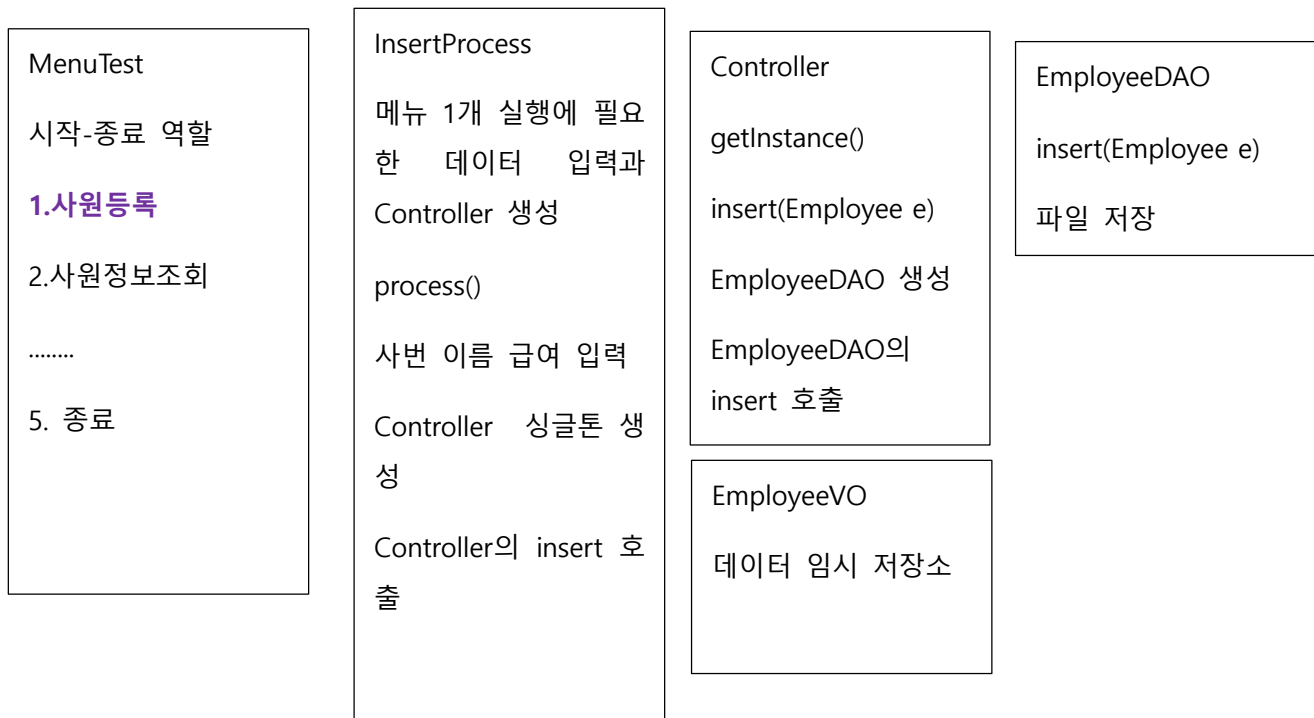
Controller = 제어 클래스

EmployeeDTO = EmployeeVO = EmployeeDO

사원등록 --> 등록 필요한 데이터 입력

---> Controller

--> ManagerDAO --> ManagerVO--> 저장



Controller – 1개 생성-singleton

```
new Controller();
```

```
new Controller();
```

- singleton dao dto

MenuTest

시작-종료 역할

1.사원등록

2.사원정보조회

.....

5. 종료

ViewProcess

process(){

모든 사원 조회 입력
데이터 없다

Controller 싱글톤 생
성

Controller의 **getAll()**
호출

}

Controller

getInstance()

insert(int, String,
double){

EmployeeVO 객체
생성

EmployeeDAO 생성

EmployeeDAO의
insert 호출

}

**ArrayList<Employee
VO>**

getAll(){

EmployeeDAO 생성

EmployeeDAO의

getAll() 호출

EmployeeDAO

void insert(Employee e)

**ArrayList<EmployeeVO>
getAll()**

**{ 파일 1 라인 ->
EmployeeVO**

--> ArrayList -> 리턴}

EmployeeVO