

급여 관리 프로그램

7 조
어 동 규 손 효 정 정 예 은

CONTENTS

- 001 프로그램 설명
- 002 클래스 다이어그램
- 003 소스 및 실행 화면

프로그램 설명

급여 관리 프로그램

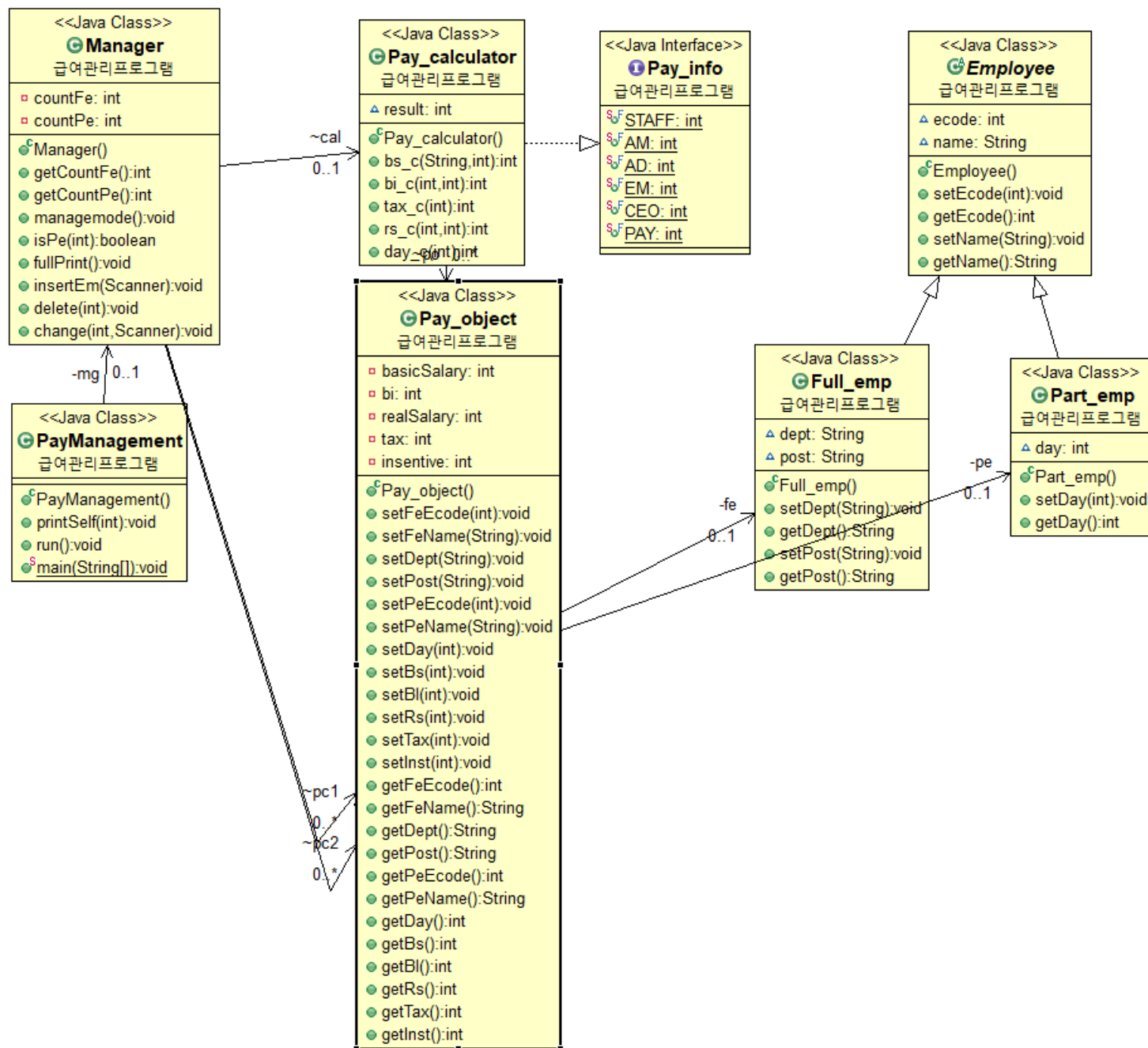
1. 프로그램 명 : 급여 관리 프로그램
2. 이용 고객 : 관리자 , 사용자 (2가지 타입)
3. 사용자 기능 : 사원번호를 입력해 자신의 월급 명세서 출력
4. 관리자 기능 : 사원정보 , 사원 삭제 , 사원 성과급 수정
전체 사원 출력

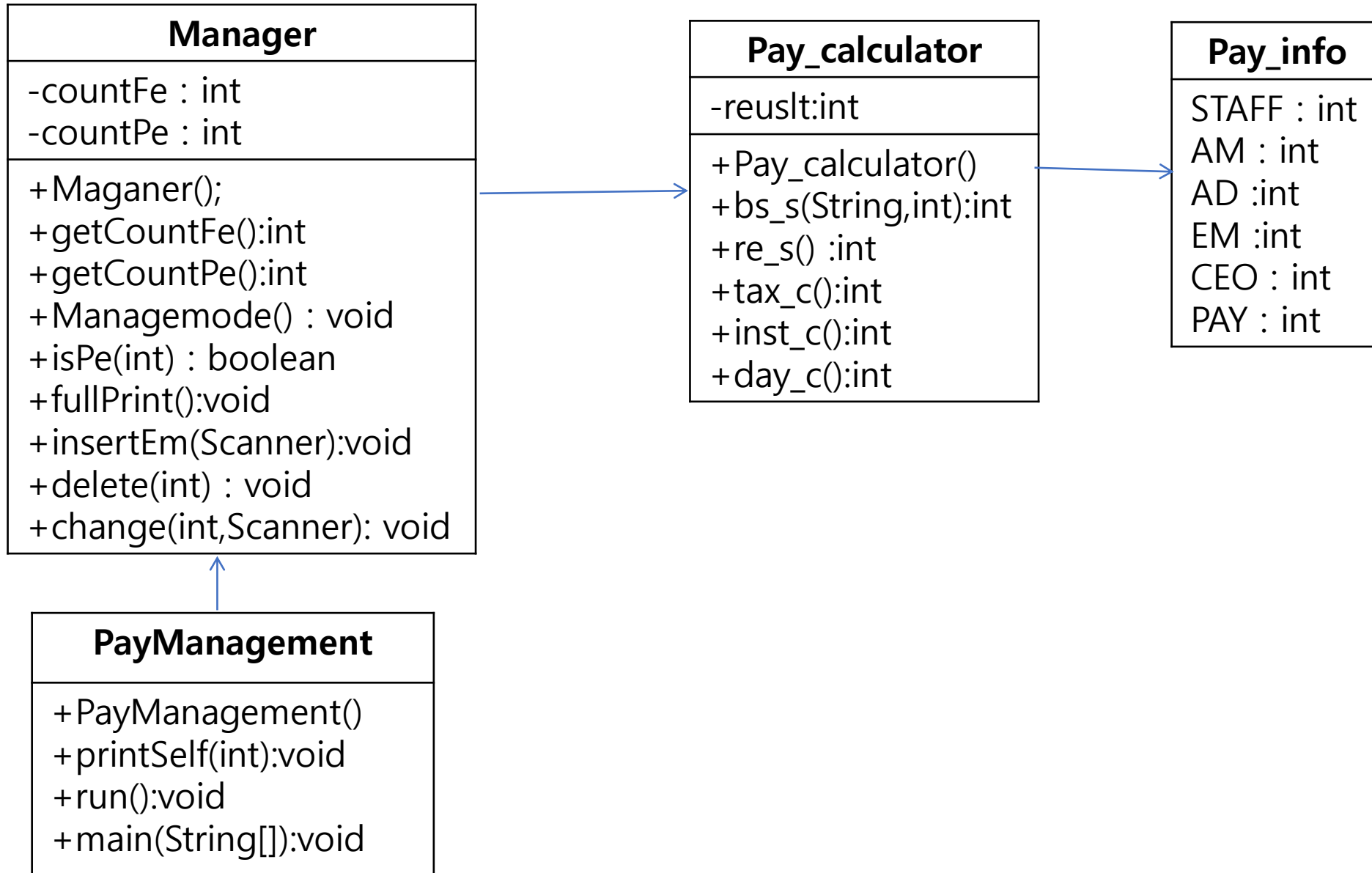
사원번호	이름	월급
------	----	----

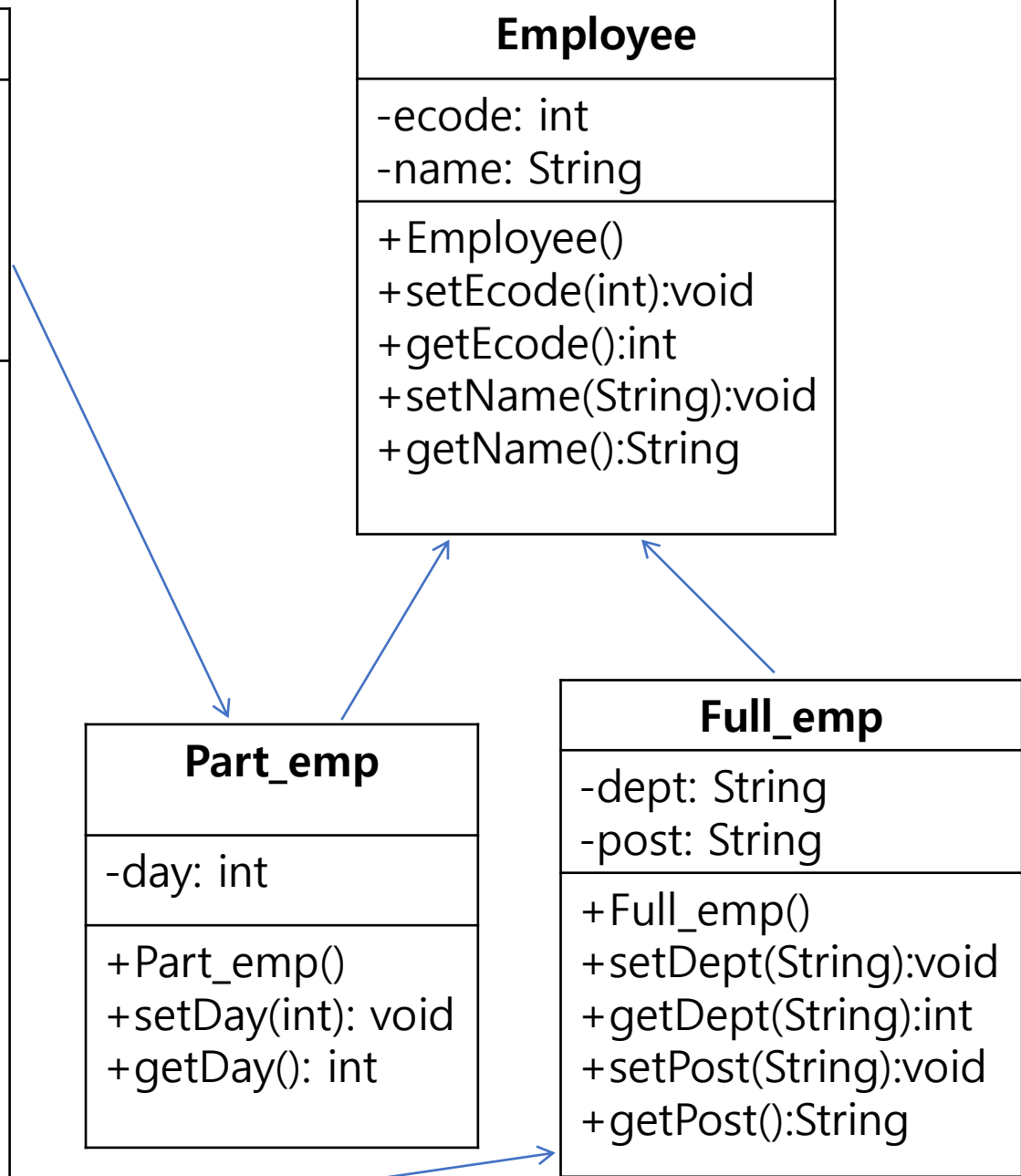
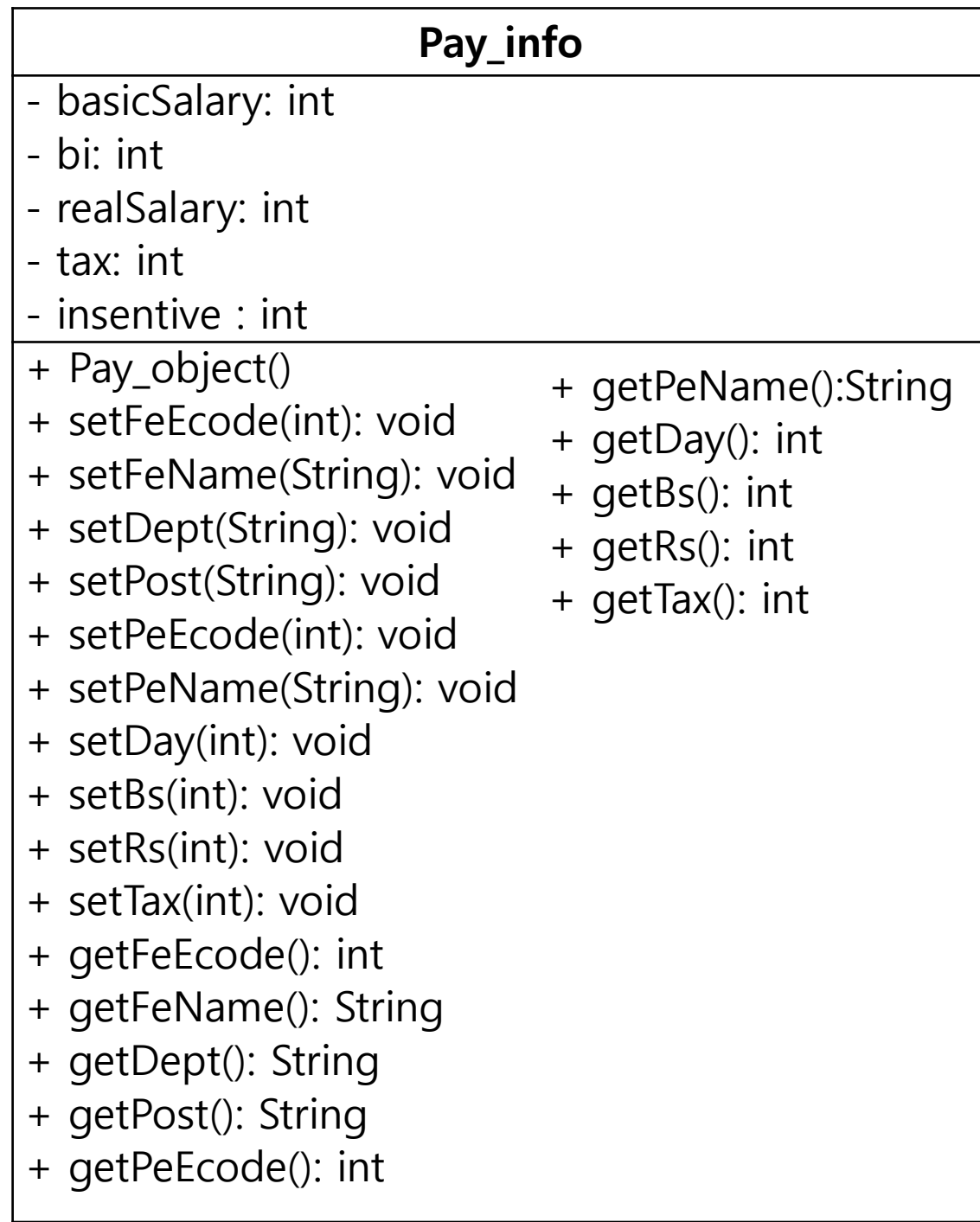
사원번호	이름	부서	직위	기본급	지급합계	공제합계	실지금액
------	----	----	----	-----	------	------	------

클래스 다이어그램

급여 관리 프로그램







소스 및 실행화면

소스

사원 정보 입력

```
public void insertEm(Scanner sc) { // 사원정보 입력

    System.out.print("입력할 사원정보를 입력하세요 사원번호:");
    String no = sc.next();
    if (no.charAt(0) == '2') { // 사원번호가 2로 시작하면 비정규직이므로 이름, 근무일수만 받음
        pc2[countPe] = new Pay_object();
        pc2[countPe].setPeEcode(Integer.parseInt(no));
        System.out.print("추가할 사원정보를 입력하세요 이름:");
        pc2[countPe].setPeName(sc.next());
        System.out.println("근무한 일수를 입력하세요");
        pc2[countPe].setBs(cal.day_c(sc.nextInt()));

        countPe++;
    } else { // 정규직은 이름, 부서, 직위, 입사년차를 추가로 받음
        pc1[countFe] = new Pay_object();
        pc1[countFe].setFeEcode(Integer.parseInt(no));
        System.out.print("추가할 사원정보를 입력하세요 이름:");
        pc1[countFe].setFeName(sc.next());
        System.out.print("추가할 사원정보를 입력하세요 부서:");
        pc1[countFe].setDept(sc.next());
        System.out.print("추가할 사원정보를 입력하세요 직위:");
        pc1[countFe].setPost(sc.next());
        System.out.print("추가할 사원정보를 입력하세요 입사년차:");
        pc1[countFe].setBs(cal.bs_c(pc1[countFe].getPost(), sc.nextInt())); //사원별 기본급 초기화
        pc1[countFe].setBI(cal.bi_c(pc1[countFe].getBs(), pc1[countFe].getInst())); //사원별 지급합계(기본급+성과급) 초기화
        pc1[countFe].setTax(cal.tax_c(pc1[countFe].getBI())); //사원별 세금 초기화
        pc1[countFe].setRs(cal.rs_c(pc1[countFe].getBI(), pc1[countFe].getTax())); //사원별 실급여액(지급합계-세금) 초기화

        countFe++;
    }
}
```

소스 및 실행화면

소스

전체 출력

```
public void fullPrint() { // 전체 사원 출력

    System.out.print("사원번호\t이름\t일당 \n");
    for (int i = 0; i < countPe; i++) {
        System.out.println(pc2[i].getPeEcode()+"\t"+pc2[i].getPeName()+"\t"+pc2[i].getBs()+"\n");
    }

    System.out.println("사원번호\t이름\t부서\t직위\t기본급\t성과금\t지급합계\t공제합계 \t실지급액 \n");
    for (int i = 0; i < countFe; i++) {
        System.out.println(pc1[i].getFeEcode()+"\t"+pc1[i].getFeName()+"\t"+pc1[i].getDept()+"\t"
            +pc1[i].getPost()+"\t"+pc1[i].getBs()+"\t"+pc1[i].getInst()+"\t"+pc1[i].getBI()+"\t"
            +pc1[i].getTax()+"\t"+pc1[i].getRs()+"\t");
    }
}
```


소스 및 실행화면

소스

급여 계산

```
//기본급 : 직책 + 호봉
public int bs_c(String post, int year) {
    if(post.equals("사원"))
        result = STAFF * year;
    else if (post.equals("대리"))
        result = AM * year;
    else if (post.equals("차장"))
        result = AD * year;
    else if (post.equals("부장"))
        result = EM * year;
    else if (post.equals("사장"))
        result = CEO * year;
    else
        System.out.println("오류");
    return result;
}
```

```
//지급합계: (기본급+성과급)
public int bi_c(int bs, int inst) {
    result = bs + inst;
    return result;
}
```

```
//공제합계: 지급합계 * 세금(10%)
public int tax_c(int rs) {
    result = (int)(rs * 0.10);
    return result;
}
```

```
//실지급액: 지급합계-공제합계
public int rs_c(int bi, int tax) {
    result = bi - tax;
    return result;
}
```

```
//비정규직 월급: 일수 * 일당
public int day_c(int day) {
    result = DPAY * day;
    return result;
}
```

상속 인터페이스 사용

소스 및 실행화면

소스

```
public class Full_emp extends Employee{

    String dept;    //부서
    String post;    //직급

    public Full_emp() {
        dept = null;
        post = null;
    }
    public void setDept(String dept) {
        this.dept = dept;
    }
    public String getDept() {
        return dept;
    }
    public void setPost(String post) {
        this.post = post;
    }
    public String getPost() {
        return post;
    }
}
```

```
2 public class Part_emp extends Employee{
3
4     int day;    //일 수
5
6     public Part_emp() {
7         day = 0;
8     }
9     public void setDay(int day) {
10        this.day = day;
11    }
12    public int getDay() {
13        return day;
14    }
15
16 }
17
18 interface Pay_info {
19
20     int STAFF = 1800000;    //사원
21     int AM = 2100000;    //대리
22     int AD = 2400000;    //차장
23     int EM = 2700000;    //부장
24     int CEO = 3000000;    //사장
25     int DPAY = 100000;    //일당
26
27 }
```

소스 및 실행화면

실행화면

관리자 실행화면

원하는 번호를 선택하세요 1. 관리자 2. 사용자(월급 명세서 출력) 3. 종료

1

원하는 번호를 선택하세요 1. 전체출력 2. 사원추가 3. 사원삭제 4. 수정 5. 종료

2

입력할 사원정보를 입력하세요 사원번호: 101

추가할 사원정보를 입력하세요 이름: 정

추가할 사원정보를 입력하세요 부서: 개발

추가할 사원정보를 입력하세요 직위: 차장

추가할 사원정보를 입력하세요 입사년차: 2

원하는 번호를 선택하세요 1. 전체출력 2. 사원추가 3. 사원삭제 4. 수정 5. 종료

2

입력할 사원정보를 입력하세요 사원번호: 102

추가할 사원정보를 입력하세요 이름: 어

추가할 사원정보를 입력하세요 부서: 개발

추가할 사원정보를 입력하세요 직위: 대리

추가할 사원정보를 입력하세요 입사년차: 1

원하는 번호를 선택하세요 1. 전체출력 2. 사원추가 3. 사원삭제 4. 수정 5. 종료

2

입력할 사원정보를 입력하세요 사원번호: 201

추가할 사원정보를 입력하세요 이름: 손

근무한 일수를 입력하세요

25

원하는 번호를 선택하세요 1. 전체출력 2. 사원추가 3. 사원삭제 4. 수정 5. 종료

1

사원번호	이름	일당
201	손	2500000

사원번호	이름	부서	직위	기본급	성과급	지급합계	공제합계	실지금액
101	정	개발	차장	4800000	0	4800000	480000	4320000
102	어	개발	대리	2100000	0	2100000	210000	1890000

원하는 번호를 선택하세요 1. 전체출력 2. 사원추가 3. 사원삭제 4. 수정 5. 종료

감사합니다