

<직원 급여 관리 프로그램 >

직원 급여 관리 프로그램

연봉협삼팀 | 김근미, 김현지, 반수희

(Class)

01

Salary
Management

02

Salary
List

03

Employee

04

PayStep

05

OverPay

06

Kpi

07

Position



직원



사번	이름	비정규직(유/무)	직급	입사일자	근속연수	성과점수
1653521	김근미	무	L0	2017-12-26	6	5
1655431	김현지	유	L1	2019-12-26	4	4
2360757	반수희	무	L0	2020-01-02	4	5



직급

직급	직급별급여
주임	300,000
계장	500,000
대리	700,000
과장	1,000,000
차장	1,200,000
부장	1,500,000
임원	5,000,000




호봉

직급	근속연수	호봉	호봉급
주임	0	0	₩0
임원	0	0	₩0
계장	1년 ~ 3년	1	₩1,900,000
계장	4년 ~ 7년	2	₩2,100,000
대리	1년 ~ 3년	1	₩2,500,000
대리	4년 ~ 7년	2	₩2,700,000
과장	1년 ~ 3년	1	₩3,100,000
과장	4년 ~ 7년	2	₩3,300,000
차장	1년 ~ 3년	1	₩3,700,000
차장	4년 ~ 7년	2	₩3,900,000
부장	1년 ~ 3년	1	₩4,300,000
부장	4년 ~ 7년	2	₩4,500,000



성과

성과	지급률(월)
탁월	200.00%
우수	150.00%
양호	100.00%
보통	50.00%
미흡	25.00%

<div> 급여</div>							
사번	기본급	호봉급	성과급	시간외수당	세금	세금 전	세금 후
1653521	₩500,000	₩2,100,000	₩1,000,000	₩0	15%	₩3,600,000	₩3,060,000
1655431	₩700,000	₩2,700,000	₩1,400,000	₩0	15%	₩4,800,000	₩4,080,000
2360757	₩500,000	₩2,100,000	₩1,000,000	₩0	15%	₩3,600,000	₩3,060,000

월급여: 급여 계산식 : 직급별 기본급 + 호봉별 호봉급 + 성과점수별 지급률 *기본급 + 시간외수당 - 세금

세금: 월급여 * 15%

(Position, Kpi)

```
public class Position {  
    private String position; //직원의 급여  
    private int positionPay; //직급별 급여
```

```
void positionPayResult(String position) { //직급별 급여할당  
    switch(position) {  
        case "주임":  
            positionPay = 300000;  
            break;
```

position 객체 생성시
positionPay 자동 할당

```
public class Kpi {  
    private String kpi; //성과:탁월,우수,양호,보통,미흡  
    private double kpiRate; //성과점수에 따른 성과지급률
```

```
void kpiRateResult(String kpi)  
    switch(kpi) {  
        case "탁월" :  
            kpiRate = 2;  
            break;  
        case "우수" :  
            kpiRate = 1.5;  
            break;
```

Kpi 객체 생성시
kpiRate 자동 할당

(OverPay, PayStep)

```
public class OverPay {  
    private int overTime; // 시간외근무(분)  
    private int overPay; // 30분당 시간외수당  
  
    // constructor  
    public OverPay() {  
    }  
}
```

```
public void setOverTime(int overTime) {  
    this.overTime = overTime;  
    overPayResult(overTime); //overtime  
}  
  
public void overPayResult(int overTime) {  
    overPay = (overTime / 30) * 10000; //
```

overTime 입력 시
overPay 자동 할당 (30분당 1만원)

```
public class PayStep {  
    private int payStep; //호봉  
    private int stepSalary; //호봉별급여  
    private int tenure; //근속연수  
    private String position; //직급
```

```
public void StepSalaryResult() {  
    switch(position) {  
        case "주임":  
            stepSalary = 0;  
            payStep = 0;  
        case "임원":  
            stepSalary = 0;  
            payStep = 0;
```

직급과 근속연수 입력
↓
호봉과
호봉별급여 계산

(EMPLOYEE)

```
public class Employee {  
    private String employeeNum; //사번  
    private String name; //직원이름  
    private int temp = 0; // 0이면 정규직 1이면 비정규직  
    private Position position; //직급  
    private String hireDate; //입사일자  
    private int tenure; //근속연수  
    private Kpi kpi; //성과지표  
    private OverPay overPay; // 시간외근무(해당시) 분입력 (SET)  
    private PayStep payStep; // 자동계산  
    private double salary;  
    private double taxRate = 0.15;  
    private double tax;
```

Getter ,Setter

EmployeeTemp

주임,임원이면 비정규직여부인 1 반환

tenureResult

현재 날짜와 입사일자 차이를 구해
근속연수 반환

salaryResult

변수를 활용한 급여 계산

```
public Employee(String employeeNum, String name, String position,
                String hireDate, String kpi, int overtime) {
    this.employeeNum = employeeNum;
    this.name = name;
    this.position = new Position(position);
    this.hireDate = hireDate;
    this.kpi = new Kpi(kpi);
    EmployeeTemp(this.position);
    tenure = tenureResult(hireDate);
    payStep = new PayStep(position, tenure);
    overPay = new OverPay(overtime);
}
```

Employee
생성자

//constructor
//나머지는 자동계산, 필수입력 사번, 이름, 직급, 입사일자,
성과지표, 시간외근무(분)

자동
계산

positionPay , kpiRate

직급별 급여 성과지급률

overPay, stepSalary

시간외 수당 호봉별 급여

temp,

비정규직 유무

tenure

근속연수

(SALARYLIST)

```
ic class SalaryList {  
Employee []employeeList = new Employee[100];  
Scanner sc = new Scanner(System.in);  
int num = 0;
```

Method

employeeList[]

employee 객체를 담을 배열 생성

```
public void mainDisplay() { // main 화면 (=Home화면)..  
public void input() { //필수적인 정보를 입력받고 Employee 객체 생성 후 배열에 담음..  
public void viewInfo(int i) { //i 인덱스에 해당하는 직원 출력..  
public void viewPayInfo(int i) { //직원별 급여 정보 출력..  
public void search() { //1.직원 인사정보 조회 2. 직원 급여명세서 조회..  
public void allView() { //모든 직원 인사정보 출력..  
public void update() { //필수 요소중 변할수있는 직급,성과점수,시간외(분) 수정기능..
```

(MAIN)

```
public class SalaryManagement {
    public static void main(String args[]) {
        //나머지는 자동계산, 필수입력 사번, 이름, 직급, 입사일자, 성과지표
        SalaryList salaryList = new SalaryList();
        int selectNum = 0;
        Scanner sc = new Scanner(System.in);

        do {
            salaryList.mainDisplay(); // 초기화면
            selectNum = sc.nextInt();
            switch(selectNum) {
                case 1:
                    salaryList.input(); //입력메서드
                    break;
                case 2:
                    salaryList.update(); //수정메서드
                    break;
                case 3:
                    salaryList.search(); //검색메서드
                    break;
                case 4:
                    salaryList.allView(); //조회메서드
                    break;
                case 5:
                    selectNum = 5;
                    System.out.println("프로그램을 종료합니다.");
                    break;
                default:
                    System.out.println("1 ~ 5 사이의 숫자로 입력해주세요.");
                    break;
            }
        }while(selectNum != 5);
    }
}
```

(The End)



Thank you!