



급여계산

경북대학교
소프트웨어융합과
배희호 교수



급여 계산

- 다음에서 제시하는 처리 조건에 따라 사원이름(name), 사번(bunho), 생년월일, 호봉(hobong), 수당(extra_pay)을 객체 배열에 입력 받은 후, 아래의 처리 조건에 의하여 급여(salary)를 계산하는 프로그램을 객체지향으로 작성하시오.





급여 계산

■ 실행 결과

박대국님의 수당 입력 : 590000

하충민님의 수당 입력 : 560000

이름	사번	호봉	수당	본봉	세금	급여

길대연(2013년 12월 20일)	1345	5	790,000	750,000	1,540	1,538,460
김대한(1999년 5월 13일)	3456	5	780,000	750,000	1,530	1,528,470
박정환(2012년 7월 20일)	1422	4	780,000	720,000	1,500	1,498,500
이민국(1987년 6월 22일)	5634	1	750,000	650,000	1,400	1,398,600
나정산(2011년 4월 19일)	1362	3	678,900	700,000	1,378	1,377,522
박대국(2011년 8월 10일)	1789	2	590,000	680,000	1,270	1,268,730
하충민(2012년 3월 20일)	4521	1	560,000	650,000	1,210	1,208,790
황정민(2012년 4월 20일)	1456	4	450,000	720,000	1,170	1,168,830
홍길동(1993년 10월 5일)	1112	1	234,560	650,000	884	883,676
이나라(2010년 3월 7일)	1345	2	123,450	680,000	803	802,647

5,736,910 6,950,000 12,674,225





급여 계산



■ 입력 설계

- 최대 입력 자료의 건수는 10건 이상으로 함
- 사번(bunho)은 4자리 숫자형 문자로 입력
- 사원이름(name)은 3자리 이름
 - 테스트 데이터의 첫 번째는 반드시 **본인의 이름**을 등록할 것
- 호봉(hobong)은 문자로 (1 ~ 5) 입력
- 수당(extra_pay)은 0 ~ 1,000,000원까지로 입력



급여 계산



■ 입력 예시

사번	사원이름	호봉	생년월일	수당
9111	홍길동	1		400,000
2233	이한국	3		650,000

- 레코드 별 입력 순서는 “사번, 사원이름, 호봉, 생년월일, 수당”으로 한다
- 자료의 입력은 사번, 사원이름, 호봉, 생년월일은 초기화 하여 받을 수 있음



급여 계산



■ 처리 조건

■ 호봉에 따른 지급액

1호봉	2호봉	3호봉	4호봉	5호봉
1,650,000	1,680,000	1,700,000	1,720,000	1,750,000

■ 총 지급액에 따라 세율

총 지급액 기준	세율
1,700,000미만	5%
1,700,000이상 2,300,000미만	8%
2,300,000이상	10%



급여 계산



■ 처리 조건

- 지급액은 호봉에 따라 지급
- 총 지급액은 (지급액 + 수당)으로 함
- 세금은 (총 지급액 * 세율)로 하여 100원 이하는 버림
- 실 지급액은 (총 지급액 - 세금)으로 함
- 합계(Total)는 수당, 총계, 세금, 실 지급액을 각각 누적한 금액임
- 실 지급액(NET)이 많은 사람부터 작은 사람으로 정렬



급여 계산

클래스 Employee



→ 사원이름(name)

→ 사번(bunho)

→ 생년월일(birthday)

→ 호봉(hobong)

→ 수당(extra_pay)

데이터 입력하기

데이터 출력하기

지급액 계산하기

총 지급액 계산하기

세금 계산하기

실 지급액 계산하기



급여 계산

클래스 Company(회사)



사원

데이터

합계
정렬
출력

정보

정보 출력하기

합계 구하기

정렬하기



Futuristic Innovator

京福大學校
KYUNGBOK UNIVERSITY



급여 계산(I)

■ Date.JAVA

```
public class Date {  
    private int year;  
    private int month;  
    private int day;  
  
    public Date(int year, int month, int day) {  
        setYear(year);  
        setMonth(month);  
        setDay(day);  
    }  
  
    public void setYear(int year) {  
        if (year > 1940)  
            this.year = year;  
        else  
            System.err.println("년도 오류");  
    }  
}
```



급여 계산(I)



■ Date.JAVA

```
public void setMonth(int month) {  
    if (month >= 1 && month <= 12)  
        this.month = month;  
    else  
        System.err.println("월 오류");  
}
```

```
public void setDay(int day) {  
    boolean flag = false;  
    if (day >= 1) {  
        if (leapYear(year) && month == 2) {  
            if (day <= 29)  
                this.day = day;  
            else  
                flag = true;  
        } else {
```



급여 계산(I)



```
if (month == 1 || month == 3 || month == 5 || month == 7
    || month == 8) {
    if (day <= 31)
        this.day = day;
    else
        flag = true;
} else {
    if (day <= 30)
        this.day = day;
    else
        flag = true;
}
} else {
    flag = true;
}
if (flag)
    System.err.println("일자 오류");
}
```



급여 계산(I)



■ Date.JAVA

```
private boolean leapYear(int year) {  
    boolean flag = false;  
    if ((year % 4 == 0 && year % 100 != 0) || year % 400 == 0)  
        flag = true;  
    return flag;  
}
```

@Override

```
public String toString() {  
    return String.format("(%4d년 %2d월 %2d일)", year, month, day);  
}  
}
```



급여 계산(I)



■ Employee.JAVA

```
public class Employee {  
    private String name;  
    private String bunho;  
    private Date birthday;  
    private char hobong;  
    private int extra_pay;  
  
    public Employee(String name, String bunho, Date birthday, char hobong) {  
        this.name = name;  
        this.bunho = bunho;  
        this.birthday = birthday;  
        this.hobong = hobong;  
    }  
}
```



급여 계산(I)



■ Employee.JAVA

```
public Employee(String name, String bunho, Date birthday,  
                char hobong, int extra_pay) {  
    this.name = name;  
    this.bunho = bunho;  
    this.birthday = birthday;  
    this.hobong = hobong;  
    this.extra_pay = extra_pay;  
}  
  
public void setExtra_pay(int extra_pay) {  
    this.extra_pay = extra_pay;  
}  
  
public int getExtra_pay() {  
    return extra_pay;  
}
```



급여 계산(I)



■ Employee.JAVA

```
public int input() throws IOException {  
    Scanner keyboard = new Scanner(System.in);  
    int pay;  
    while (true) {  
        System.out.printf("%s님의 수당 입력 : ", name);  
        pay = keyboard.nextInt();  
        if (pay >= 0 && pay <= 1000000)  
            break;  
        else {  
            System.err.println("입력 오류");  
            System.in.read();  
        }  
    }  
    return pay;  
}
```




급여 계산(I)

```
public int gross() {  
    int gross = 0;  
    switch(hobong) {  
        case '1':  
            gross = 650000;  
            break;  
        case '2':  
            gross = 680000;  
            break;  
        case '3':  
            gross = 700000;  
            break;  
        case '4':  
            gross = 720000;  
            break;  
        case '5':  
            gross = 750000;  
    }  
    return gross;  
}
```



급여 계산(I)



■ Employee.JAVA

```
public int net() {  
    return extra_pay + gross() - tax();  
}  
  
public int tax() {  
    int total = extra_pay + gross();  
    int tax;  
    if (total < 700000) {  
        tax = (int)((double)total * 0.05D * 1.0D / 100.0D);  
    } else if (total < 750000) {  
        tax = (int)((double)total * 0.08D * 1.0D / 100.0D);  
    } else {  
        tax = (int)((double)total * 0.1D * 1.0D / 100.0D);  
    }  
    return tax;  
}
```



급여 계산(I)

■ Employee.JAVA

@Override

```
public String toString() {  
    return String.format(" %3s%s %7s %3c %,10d %,10d %,6d %,10d",  
        name, birthday, bunho, hobong, extra_pay, gross(), tax(), net());  
}  
}
```



급여 계산(I)



■ Company.JAVA

```
public class Company {  
    private Employee[] employees;  
  
    public Company(Employee[] employees) {  
        this.employees = employees;  
    }  
  
    private int sudangtotal() {  
        int temp = 0;  
        for (int i = 0; i < employees.length; i++)  
            temp += employees[i].getExtra_pay();  
  
        return temp;  
    }  
}
```



급여 계산(I)

■ Company.JAVA

```
private int bonbongtotal() {  
    int temp = 0;  
    for (int i = 0; i < employees.length; i++)  
        temp += employees[i].gross();  
  
    return temp;  
}
```

```
private int netttotal() {  
    int temp = 0;  
    for (int i = 0; i < employees.length; i++)  
        temp += employees[i].net();  
  
    return temp;  
}
```



급여 계산(I)

■ Company.JAVA

```
private void sort() {  
    Employee temp;  
    for (int i = 0; i < employees.length - 1; ++i) {  
        for (int j = i + 1; j < employees.length; ++j) {  
            if (employees[i].net() < employees[j].net()) {  
                temp = employees[i];  
                employees[i] = employees[j];  
                employees[j] = temp;  
            }  
        }  
    }  
}
```

```
private String total() {  
    return String.format("WtWtWtWtWtWtWtWt %10d %10d Wt %10d",  
        sudangtotal(), bonbongtotal(), nettotal());  
}
```



급여 계산(I)



■ Company.JAVA

```
public void display() {  
    sort();  
    line();  
    System.out.println("      이름      사번  호봉      수당      본봉  
                        세금      급여");  
    line();  
  
    for(int i = 0; i < employees.length; ++i) {  
        if (i != 0 && i % 5 == 0) {  
            System.out.println();  
        }  
        System.out.println(employees[i]);  
    }  
  
    line();  
    System.out.println(total());  
    line();  
}
```



급여 계산(I)

■ Company.JAVA

```
private void line() {  
    System.out.println("*****  
                        *****");  
}  
}
```




급여 계산(I)

■ Main.JAVA

```
public class Main {  
    public static void main(String[] args) throws IOException {  
        Employee[] person = new Employee[]{  
            new Employee("홍길동", "1112", new Date(1993, 10, 5), '1'),  
            new Employee("이나라", "1345", new Date(2010, 3, 7), '2'),  
            new Employee("나정산", "1362", new Date(2011, 4, 19), '3'),  
            new Employee("박정환", "1422", new Date(2012, 7, 20), '4'),  
            new Employee("김대한", "3456", new Date(1999, 5, 13), '5'),  
            new Employee("이민국", "5634", new Date(1987, 6, 22), '1'),  
            new Employee("황정민", "1456", new Date(2012, 4, 20), '4'),  
            new Employee("길대연", "1345", new Date(2013, 12, 20), '5'),  
            new Employee("박대국", "1789", new Date(2011, 8, 10), '2'),  
            new Employee("하충민", "4521", new Date(2012, 3, 20), '1') };  
  
        for (int i = 0; i < person.length; i++)  
            person[i].setExtra_pay(person[i].input());  
    }  
}
```



급여 계산(I)



■ Main.JAVA

```
Company company = new Company(person);  
company.display();
```

```
}
```

```
}
```



급여 계산(II)



■ Employee.JAVA

```
public class Employee {  
    private String name;  
    private String bunho;  
    private char hobong;  
    private int extra_pay;  
  
    public Employee(String name, String bunho, char hobong, int extra_pay) {  
        this.name = name;  
        this.bunho = bunho;  
        this.hobong = hobong;  
        this.extra_pay = extra_pay;  
    }  
  
    public int getExtra_pay() {  
        return extra_pay;  
    }  
}
```



급여 계산



■ Employee.JAVA

```
public void setExtra_pay(int extra_pay) {  
    this.extra_pay = extra_pay;  
}  
public int input() throws IOException {  
    Scanner keyboard = new Scanner(System.in);  
    int pay;  
    while (true) {  
        System.out.printf("%s님의 수당 입력 : ", name);  
        pay = keyboard.nextInt();  
        if (pay >= 0 && pay <= 1000000)  
            break;  
        else {  
            System.err.println("입력 오류");  
            System.in.read();  
        }  
    }  
    return pay;  
}
```



급여 계산

```
public int gross() {  
    int gross = 0;  
    switch(hobong) {  
        case '1':  
            gross = 650000;  
            break;  
        case '2':  
            gross = 680000;  
            break;  
        case '3':  
            gross = 700000;  
            break;  
        case '4':  
            gross = 720000;  
            break;  
        case '5':  
            gross = 750000;  
    }  
    return gross;  
}
```



급여 계산



```
public int net() {  
    return extra_pay + gross() - tax();  
}  
public int tax() {  
    int total = extra_pay + gross();  
    int tax;  
    if (total < 700000) {  
        tax = (int)((double)total * 0.05D * 1.0D / 100.0D);  
    } else if (total < 750000) {  
        tax = (int)((double)total * 0.08D * 1.0D / 100.0D);  
    } else {  
        tax = (int)((double)total * 0.1D * 1.0D / 100.0D);  
    }  
    return tax;  
}  
public void print() {  
    System.out.printf(" %s %5s %3c %,10d %,10d %,6d %,10d\n",  
        name, bunho, hobong, extra_pay, gross(), tax(), net());  
}  
}
```



급여 계산



■ Company.JAVA

```
public class Company {  
    private Employee[] employees;  
  
    public Company(Employee[] employees) {  
        this.employees = employees;  
    }  
  
    private int sudangtotal() {  
        int temp = 0;  
        for (int i = 0; i < employees.length; i++)  
            temp += employees[i].getExtra_pay();  
  
        return temp;  
    }  
}
```



급여 계산



■ Company.JAVA

```
private int bonbongtotal() {  
    int temp = 0;  
    for (int i = 0; i < employees.length; i++)  
        temp += employees[i].gross();  
  
    return temp;  
}
```

```
private int netttotal() {  
    int temp = 0;  
    for (int i = 0; i < employees.length; i++)  
        temp += employees[i].net();  
  
    return temp;  
}
```




급여 계산

■ Company.JAVA

```
private void sort() {  
    Employee temp;  
    for (int i = 0; i < employees.length - 1; ++i) {  
        for (int j = i + 1; j < employees.length; ++j) {  
            if (employees[i].net() < employees[j].net()) {  
                temp = employees[i];  
                employees[i] = employees[j];  
                employees[j] = temp;  
            }  
        }  
    }  
}
```

```
private String total() {  
    return String.format("WtWtWtWtWtWtWtWt %10d %10d Wt %10d",  
        sudangtotal(), bonbongtotal(), nettotal());  
}
```



급여 계산



■ Company.JAVA

```
public void display() {  
    sort();  
    line();  
    System.out.println("      이름      사번  호봉      수당      급여");  
                                본봉      세금  
    line();  
  
    for(int i = 0; i < employees.length; ++i) {  
        if (i != 0 && i % 5 == 0) {  
            System.out.println();  
        }  
        System.out.println(employees[i]);  
    }  
  
    line();  
    System.out.println(total());  
    line();  
}
```



급여 계산

■ Company.JAVA

```
private void line() {  
    System.out.println("*****  
                        *****");  
}  
}
```



급여 계산(II)



■ Main.JAVA

```
public class Main {  
    public static void main(String[] args) throws IOException {  
        Employee[] employees = new Employee[]{  
            new Employee("홍길동", "1112", '1'),  
            new Employee("이나라", "1345", '2'),  
            new Employee("나정산", "1362", '3'),  
            new Employee("박정환", "1422", '4'),  
            new Employee("김대한", "3456", '5'),  
            new Employee("이민국", "5634", '1'),  
            new Employee("황정민", "1456", '4'),  
            new Employee("길대연", "1345", '5'),  
            new Employee("박대국", "1789", '2'),  
            new Employee("하충민", "4521", '1') };  
    }
```



급여 계산(II)



■ Main.JAVA

```
for (int i = 0; i < employees.length; i++)  
    employees[i].setExtra_pay(employees[i].input());
```

```
Company company = new Company(employees);  
company.display();
```

```
}
```

```
}
```



급여 계산(III)



■ Employee.JAVA

```
public class Employee {  
    private String name;  
    private String bunho;  
    private char hobong;  
    private int extra_pay;  
  
    public Employee(String name, String bunho, char hobong, int extra_pay) {  
        this.name = name;  
        this.bunho = bunho;  
        this.hobong = hobong;  
        this.extra_pay = extra_pay;  
    }  
  
    public int getExtra_pay() {  
        return extra_pay;  
    }  
}
```



급여 계산



■ Employee.JAVA

```
public void setExtra_pay(int extra_pay) {  
    this.extra_pay = extra_pay;  
}  
public int input() throws IOException {  
    Scanner keyboard = new Scanner(System.in);  
    int pay;  
    while (true) {  
        System.out.printf("%s님의 수당 입력 : ", name);  
        pay = keyboard.nextInt();  
        if (pay >= 0 && pay <= 1000000)  
            break;  
        else {  
            System.err.println("입력 오류");  
            System.in.read();  
        }  
    }  
    return pay;  
}
```



급여 계산



```
public int gross() {  
    int gross = 0;  
    switch(hobong) {  
        case '1':  
            gross = 650000;  
            break;  
        case '2':  
            gross = 680000;  
            break;  
        case '3':  
            gross = 700000;  
            break;  
        case '4':  
            gross = 720000;  
            break;  
        case '5':  
            gross = 750000;  
    }  
    return gross;  
}
```




급여 계산



```
public int net() {  
    return extra_pay + gross() - tax();  
}  
public int tax() {  
    int total = extra_pay + gross();  
    int tax;  
    if (total < 700000) {  
        tax = (int)((double)total * 0.05D * 1.0D / 100.0D);  
    } else if (total < 750000) {  
        tax = (int)((double)total * 0.08D * 1.0D / 100.0D);  
    } else {  
        tax = (int)((double)total * 0.1D * 1.0D / 100.0D);  
    }  
    return tax;  
}  
public void print() {  
    System.out.printf(" %s %5s %3c %,10d %,10d %,6d %,10d\n",  
        name, bunho, hobong, extra_pay, gross(), tax(), net());  
}  
}
```



급여 계산(III)



■ Company.JAVA

```
public class Company {  
    private ArrayList<Employee> employees;  
  
    public Company(ArrayList<Employee> employees) {  
        this.employees = employees;  
    }  
  
    public int getLength() {  
        return employees.size();  
    }  
  
    public Employee getEmployees(int index) {  
        return employees.get(index);  
    }  
}
```



급여 계산



■ Company.JAVA

```
public int sudangtotal() {  
    int temp = 0;  
    for (int i = 0; i < employees.size(); i++)  
        temp += employees.get(i).getExtra_pay();  
  
    return temp;  
}
```

```
public int bonbongtotal() {  
    int temp = 0;  
    for (int i = 0; i < employees.size(); i++)  
        temp += employees.get(i).gross();  
  
    return temp;  
}
```



급여 계산



■ Company.JAVA

```
public int nettotal() {  
    int temp = 0;  
    for (int i = 0; i < employees.size(); i++)  
        temp += employees.get(i).net();  
  
    return temp;  
}  
  
public void sort() {  
    Collections.sort(employees, new netComparator().reversed());  
}  
  
private class netComparator implements Comparator<Employee> {  
    @Override  
    public int compare(Employee o1, Employee o2) {  
        return o1.net() - o2.net();  
    }  
}
```



급여 계산



■ Company.JAVA

```
public Employee getEmployees(int index) {  
    return employees.get(index);  
}  
  
public String getSudangtotal() {  
    return String.format("%,d 천원", sudangtotal / 1000);  
}  
  
public int getBonbongtotal() {  
    return bonbongtotal;  
}  
  
public int getNettotal() {  
    return nettotal;  
}
```



급여 계산



■ Company.JAVA

```
public int nettotal() {  
    int temp = 0;  
    for (int i = 0; i < employees.size(); i++)  
        temp += employees.get(i).net();  
    return temp;  
}  
  
public void sort() {  
    Collections.sort(employees, new netComparator().reversed());  
}  
  
private class netComparator implements Comparator<Employee> {  
    @Override  
    public int compare(Employee o1, Employee o2) {  
        return o1.net() - o2.net();  
    }  
}
```



급여 계산



■ Company.JAVA

@Override

```
public String toString() {  
    return String.format("WtWtWtWt %,10d %,10d Wt %,10d",  
        sudangtotal(), bonbongtotal(), nettotal());  
}  
}
```



급여 계산



■ Output.JAVA

```
public class Output {  
    private Company company;  
  
    public Output(Company company) {  
        this.company = company;  
        company.sort();  
    }  
  
    public void display() {  
        line();  
        System.out.println(" 이름   사번   호봉   수당   본봉   세금   급여");  
        line();  
        for(int i = 0; i < company.getLength(); ++i) {  
            if (i != 0 && i % 5 == 0) {  
                System.out.println();  
            }  
            System.out.print(company.getEmployees(i));  
        }  
    }  
}
```




급여 계산



■ Output.JAVA

```
    line();  
    System.out.println(company);  
    line();  
}  
  
static void line() {  
    System.out.println("*****");  
}  
}
```



급여 계산

■ Main.JAVA

```
public class Main {  
    public static void main(String[] args) throws IOException {  
        ArrayList<Employee> employees = new ArrayList<>();  
        employees.add(new Employee("홍길동", "1112", '1'));  
        employees.add(new Employee("이나라", "1345", '2'));  
        employees.add(new Employee("나정산", "1362", '3'));  
        employees.add(new Employee("박정환", "1422", '4'));  
        employees.add(new Employee("김대한", "3456", '5'));  
        employees.add(new Employee("이민국", "5634", '1'));  
        employees.add(new Employee("황정민", "1456", '4'));  
        employees.add(new Employee("길대연", "1345", '5'));  
        employees.add(new Employee("박대국", "1789", '2'));  
        employees.add(new Employee("하충민", "4521", '1'));  
    }  
}
```



급여 계산



■ Main.JAVA

```
for (int i = 0; i < employees.size(); i++)  
    employees.get(i).setExtra_pay(employees.get(i).input());
```

```
Company company = new Company(employees);  
Output output = new Output(company);  
output.display();
```

```
}
```

```
}
```