

Programming Assignment 2

- Template: Assignment02
- 업로드: 각자의 프로젝트 폴더 (기존의 Self-Test 업로드 방식과 동일)
- Commit Message: Assignment02 (git commit -m "Assignment02")
- 제출 기한: 6/18(금) 23시 59분까지

회사는 각기 다른 역할 및 직책으로 구분된 인원들로 구성되어 있다. 이번 과제에서 우리는 이러한 회사의 구성 인원들을 클래스로 작성하고 전체적인 구조를 작성할 것이다. 클래스는 총 7개로, Company, Director, Employee, Member, PartTime, Helper, Worker 클래스가 존재한다.

- 회사에서 일하는 모든 멤버들의 정보를 저장하는 **Member** 클래스는 추상 클래스이며 **Comparable** 인터페이스를 구현한다. **Instance variable**로는 **name(이름)**, **position(직책)**이 있다. 이에 관련된 모든 메소드 및 생성자를 작성할 것

- 모든 변수를 인자로 받은 값으로 초기화하는 생성자 작성
- 필드 변수 값을 가져오고 설정하는 **accessor, mutator** 메소드 작성
- 인자로 받은 객체와 일치하는지 확인하는 **equals()** 메소드 작성
- **Name, position**의 값을 반환하는 **toString()** 메소드 오버라이딩
(출력 형태는 최종 출력 형태 참고)
- **Comparable** 인터페이스의 **compareTo()** 메소드 오버라이딩
(정렬 기준이 알파벳순이 되도록)

- 멤버 중 급여를 받고 일하는 **Employee** 클래스는 **Member** 클래스를 상속받으며 **instance variable**로 **residence(거주지)**와 **payRate(급여)**가 있다. 이에 관련된 모든 메소드 및 생성자를 작성할 것

- 인자로 받은 **Name, position** 변수는 **Member** 클래스의 생성자의 인자로 호출하고 필드의 **residence, payRate** 변수는 인자로 받은 값으로 설정하는 생성자 작성
- 필드 변수 값을 가져오고 설정하는 **accessor, mutator** 메소드 작성
- **Member** 클래스의 **toString()** 메소드를 호출해서 가져온 문자열에 **residence**의 값을 더해서 반환하는 **toString()** 메소드 오버라이딩
(출력 형태는 최종 출력 형태 참고)
- 필드의 **payRate** 변수 값을 반환하도록 **pay()** 메소드 오버라이딩

- 급여에 보너스를 받고 일하는 **Director** 클래스는 **Employee** 클래스를 상속받으며 **instance variable**로 **bonus**(보너스)가 있다. 이에 관련된 모든 메소드 및 생성자를 작성할 것

- 인자로 받은 모든 변수를 **Employee** 클래스의 생성자의 인자로 호출하고 필드의 **bonus** 변수 값을 0으로 설정하는 생성자 작성
- 필드의 **bonus** 변수 값을 인자로 받은 값으로 설정하는 **awardBonus()** 메소드 작성
- **Employee** 클래스의 **pay()** 메소드를 호출해서 얻은 급여에 보너스를 더해서 반환하도록 **pay()** 메소드 오버라이딩, **bonus** 변수 값을 0으로 설정

- 일한 시간에 따라 급여를 계산해서 받는 **PartTime** 클래스는 **Employee** 클래스를 상속받으며 **instance variable**로 **hoursWorked**(일한 시간)가 있다. 이에 관련된 모든 메소드 및 생성자를 작성할 것

- 인자로 받은 모든 변수를 **Employee** 클래스의 생성자의 인자로 호출하고 필드의 **hoursWorked** 변수 값을 0으로 설정하는 생성자 작성
- 필드 변수 값을 가져오고 설정하는 **accessor, mutator** 메소드 작성
- 인자로 받은 변수 값을 필드의 **hoursWorked** 변수 값에 더하는 **addHours()** 메소드 작성
- 상속받은 **payRate** 변수 값에 **hoursWorked** 변수 값을 곱해서 급여를 반환하도록 **pay()** 메소드 오버라이딩, **hoursWorked** 변수 값을 0으로 설정
- **Employee** 클래스의 **toString()** 메소드를 호출해서 가져온 문자열에 **hoursWorked**의 값을 더해서 반환하는 **toString()** 메소드 오버라이딩
(출력 형태는 최종 출력 형태 참고)

- 급여를 받지 않고 일하는 **Helper** 클래스는 **Member** 클래스를 상속받는다. 이에 관련된 모든 메소드 및 생성자를 작성할 것

- 인자로 받은 모든 변수를 **Member** 클래스의 생성자의 인자로 호출하는 생성자 작성
- 급여를 0으로 반환하도록 **pay()** 메소드 오버라이딩

- 회사의 모든 직원들을 관리하는 **Worker** 클래스는 **ArrayList, Iterator, Collections**와 **IO** 패키지를 **import**하고 **instance variable**로 **Member** 타입의 **ArrayList**인 **MemberList**(멤버 리스트)가 있다. 이에 관련된 모든 메소드 및 생성자를 작성할 것

- Director인 Sam, Cliff와 Employee인 Kim과 PartTime인 Takuya, Mike와 Helper인 Hong (주식으로 주어진)을 MemberList에 Member 타입으로 순서대로 추가한 다음, MemberList에 포함되어 있는 Director들에게는 보너스를 300씩, PartTime들에게는 시간을 35씩 메소드를 통해 추가하는 생성자 작성
- Iterator를 사용하여 MemberList를 순회하며 Sam을 찾아서 MemberList에서 제외하고 “Bye Sam”을 출력하는 fireSam() 메소드 작성
(출력 형태는 최종 출력 형태 참고)
- Iterator를 사용하여 MemberList를 순회하며 멤버들의 정보를 출력하고 각 멤버들의 급여를 저장한다. 급여가 0일 경우 “Thank you!”만 출력하고 그 외에는 계산된 급여를 출력하는데 콘솔창에도 출력하고 텍스트 파일로도 출력하는 payoff() 메소드 작성
(출력 형태는 최종 출력 형태 참고)
- Arrays.sort()와 같이 ArrayList를 정렬하기 위해서는 import된 Collections 클래스의 sort 메소드를 사용한다. sort 메소드를 호출하여 MemberList를 정렬하고 출력하는 memberSort() 메소드 작성
(출력 형태는 최종 출력 형태 참고)

- **main()** 메소드를 포함한 **Company** 클래스에서는 **payoff(), fireSam(), memberSort()** 메소드를 순서대로 호출하도록 한다.

- 최종 출력 형태는 다음과 같다

```
Name: Sam
Position: director
Residence: Busan
Paid: 3950.5
-----
Name: Cliff
Position: full-time
Residence: Seoul
Paid: 2753.15
-----
Name: Kim
Position: intern
Residence: America
Paid: 1548.15
-----
Name: Takuya
Position: temporary
Residence: China
Current hours: 35
Paid: 437.5
-----
Name: Mike
Position: temporary
Residence: Japan
Current hours: 35
Paid: 437.5
-----
Name: Hong
Position: volunteer
Thank you!
-----
Bye Sam

Sorted Name:
Cliff
Hong
Kim
Mike
Takuya
```

MemberList.txt - Windows 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

```
Name: Sam
Position: director
Residence: Busan
Paid: 3950.5
-----
```

```
Name: Cliff
Position: full-time
Residence: Seoul
Paid: 2753.15
-----
```

```
Name: Kim
Position: intern
Residence: America
Paid: 1548.15
-----
```

```
Name: Takuya
Position: temporary
Residence: China
Current hours: 35
Paid: 437.5
-----
```

```
Name: Mike
Position: temporary
Residence: Japan
Current hours: 35
Paid: 437.5
-----
```

```
Name: Hong
Position: volunteer
Thank you!
-----
```