

Java Programming (실기 테스트)

Java를 이용하여 다음 요구사항을 만족하는 프로그램을 작성하여 문법과 객체지향 개념을 이해, Class Diagram 설계, 프로그램을 구현 할 수 있는지 확인합니다.

****요구사항****

회사는 사원이 일반 사원과 개발자 사원으로 구분된다.

일반 사원(Employee)은 사번, 이름, 부서명, 급여에 대한 정보를 확인할 수 있다.

개발자 사원(Programmer)은 사번, 이름, 부서, 급여, 담당SW이름(ex)Java, C, C++ ...)에 대한 정보를 확인할 수 있다.

- 일반 사원 와 개발자 사원의 정보를 각각 2명 이상 입력(총 4명 이상)

(단 사번은 중복 저장하지 않음)

- 사번 중복 여부 확인
- 입력된 모든 사원의 목록을 확인
- 입력된 사원중 개발자의 목록만 확인
- 특정 사원의 급여(salary)를 확인
- 특정 사원의 부서명(departmentName)를 수정
- 특정 사원의 담당SW이름(swName) 수정
- 같은 이름을 가진 사원의 사번 목록 확인
- 회사에서 지급하는 사원의 총 급여 확인
- 회사에서 지급하는 사원의 평균 급여 확인

다음은 요구사항을 기반 설계된 인터페이스(ManagerService.java) 제공

<<interface>> ManagerService
+ addEmployee(e : Employee) : void + isEmployeeId(employeeId : String) : boolean + getEmployees() : Collection<Employee> + getProgrammers() : Collection<Programmer> + getEmployeeSalary(employeeId : String) : int + setDepartmentName(employeeId : String, updateDeptName:String) : boolean + setSwName(employeeId : String, updateSwName:String) : boolean + getEmployeeIds(name : String) : ArrayList<String> + getEmployeeTotalSalary() : int + getEmployeeTotalAvgSalary() : float

- 재사용을 고려한 요구사항을 상속 설계(추가 점수)
- 정의된 메서드를 호출해서 구현(추가 점수)

1. ManagerService를 상속받아 ManagerServiceImpl.java로 구현
2. 인터페이스를 보고 추가 객체 정의
3. 해당 인터페이스를 확인하는 ManagerServiceTest.java 에서 main()로 구현
4. 원본 패키지 com.oopsw , 테스트 패키지 test.com.oopsw
5. 식별자 대소문자 규칙 준수

--제출 사항 --

원본소스(.java)

제출 문서 - 요구사항, ClassDiagram, 실행 결과(Test) 순으로 이름.pdf를 작성
