

학생 출결 관리 및 조회 시스템

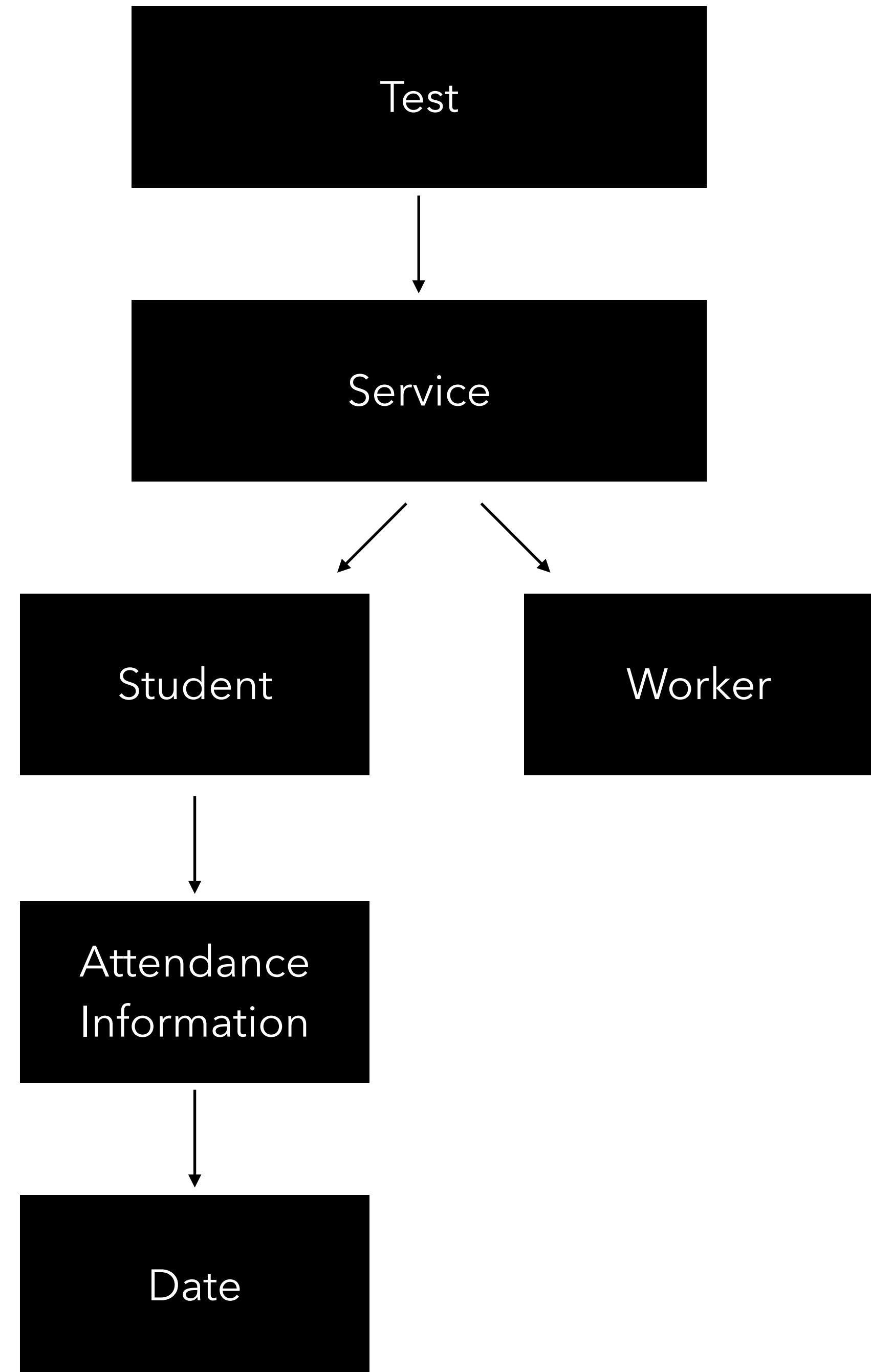
3조

해당 프로젝트는 데이터 사이언티스트 과정을 진행하는 학생에 대한 정보를 관리하는 프로그램을 제작하는데 목적을 두고 있다.

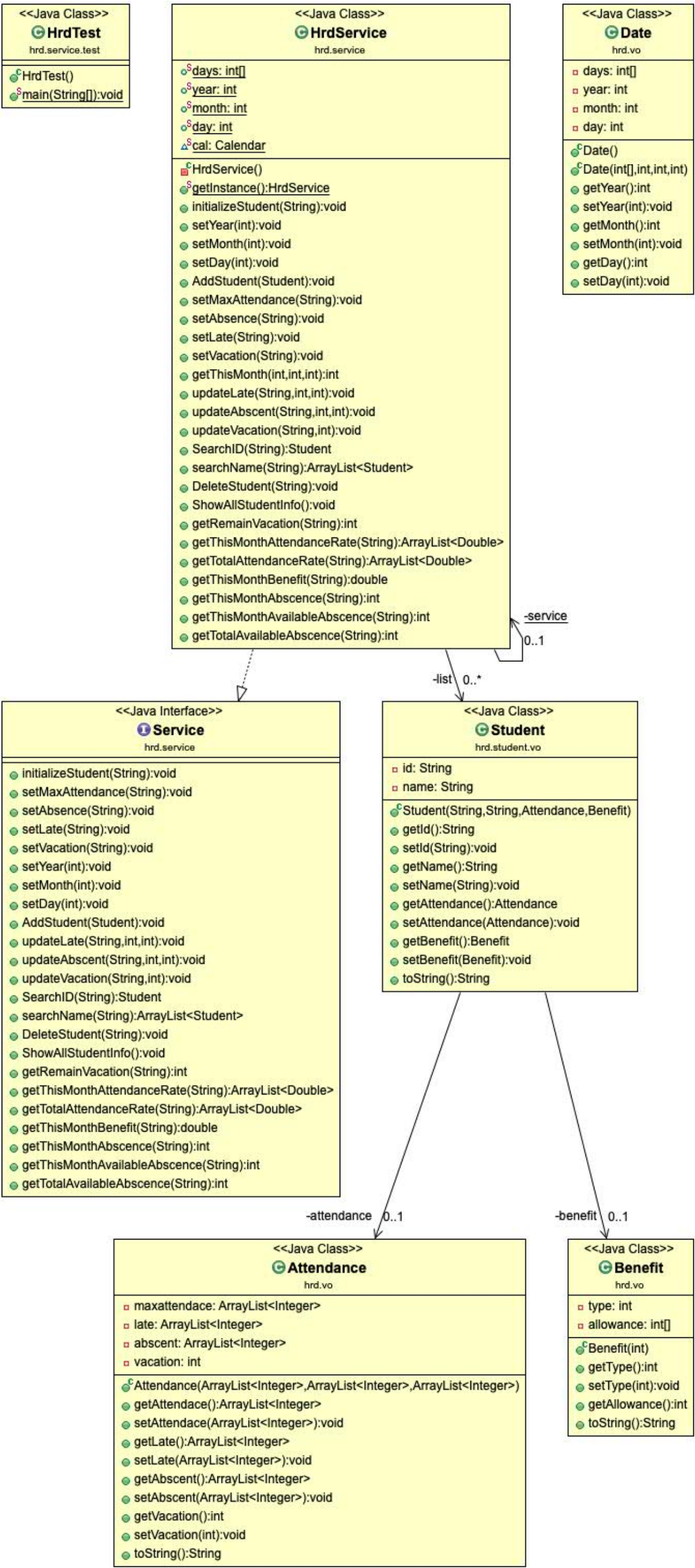
프로젝트 목적

1. 훈련수당을 받기위해 이번달에 얼마나 더 결석이 가능한지 계산
2. 수료를 위해 얼마나 더 결석이 가능한지 계산
3. 실제로 우리가 유용하게 사용할 수 있는 프로그램을 구현
4. 관리자 사용 위해 학생 관리 기능 추가

기존 Class Diagram



변경된 Class Diagram



초기 콘셉트와 다른 점

1. 관리 위주 기능 중점 에서 출결 계산 중점으로 변경
2. 접근 권한 제거
3. Date 를 vo가 아닌 service에 구현

기존 Interface

```
public interface Service {  
    public static Date today;  
    public void AddStudent(Student s);  
    public void UpdateStudent(String id, Student s);  
    public Student SearchStudent(String id);  
    public ArrayList<Student> SearchStudent(String name);  
    public void DeleteStudent(String id);  
    public double getThisMonthAttendance(String id);  
    public double getTotalAttendance(String id);  
    public double getThisMonthBenefit(String id);  
    public int getThisMonthAbscent(String id);  
    public int getThisMonthAvailableAbscent(String id);  
    public int getTotalAvailableAbscent(String id);  
    public ArrayList<Student> ShowAllStudentInfo();  
    public int getRemainVacation(String id);  
}
```

변경된 **Interface** 및 중요 기능

```
//=====기본=====
    public void setMaxAttendance String id ;//수업일수초기화
    public void setAbsence String id ;//결석초기화
    public void setLate String id ;//지각초기화
    public void setVacation String id ;//휴가 초기화
    public void setYear int year ;//년도 초기화
    public void setMonth int month ; // 월 초기화
    public void setDay int day ; // 일 초기화

// =====CURD=====
    public void AddStudent Student s ; // 학생 추가
    public void updateLate String id, int month, int late ; // 지각 횟수 업데이트
    public void updateAbscent String id, int month, int abscent ; // 결석한 일수 업데이트
    public void updateVacation String id, int vacation ; // 사용한 연가 일수 업데이
    public Student SearchID String id ;//고유ID값 반환
    public ArrayList<Student> searchName String name ;//이름으로 검색 -> 같은 이름 다른 아이디를 쓰는 학생들의 리스트를 반환
    public void DeleteStudent String id ;// 특정 학생 삭제
    public void ShowAllStudentInfo ; // 모든 학생 데이터 출
///=====기능=====
    public int getRemainVacation String id ; // 남은 휴가 출력
    public ArrayList<Double> getThisMonthAttendanceRate String id ;//이달 출석일수와 출석률 출력
    public ArrayList<Double> getTotalAttendanceRate String id ; //오늘일자까지 출석일수와 출석률 출력
    public double getThisMonthBenefit String id ; //이번달 수당 출력
    public int getThisMonthAbscence String id ; //이달 결석일수 출력
    public int getThisMonthAvailableAbscence String id ; //이달 결석가능일수 출력
    public int getTotalAvailableAbscence String id ; //교육기간 전체 결석가능일수 출력(지금까지 결석, 지각 고려)
```

변경된 **Interface** 및 중요 기능

```
//=====기본=====
    public void setMaxAttendance String id ;//수업일수초기화
    public void setAbsence String id ;//결석초기화
    public void setLate String id ;//지각초기화
    public void setVacation String id ;//휴가 초기화
    public void setYear int year ;//년도 초기화
    public void setMonth int month ; // 월 초기화
    public void setDay int day ; // 일 초기화

// =====CURD=====
    public void AddStudent Student s ; // 학생 추가
    public void updateLate String id, int month, int late ; // 지각 횟수 업데이트
    public void updateAbscent String id, int month, int abscent ; // 결석한 일수 업데이트
    public void updateVacation String id, int vacation ; // 사용한 연가 일수 업데이
    public Student SearchID String id ;//고유ID값 반환
    public ArrayList<Student> searchName String name ;//이름으로 검색 -> 같은 이름 다른 아이디를 쓰는 학생들의 리스트를 반환
    public void DeleteStudent String id ;// 특정 학생 삭제
    public void ShowAllStudentInfo ; // 모든 학생 데이터 출
///=====기능=====
    public int getRemainVacation String id ; // 남은 휴가 출력
    public ArrayList<Double> getThisMonthAttendanceRate String id ;//이달 출석일수와 출석률 출력
    public ArrayList<Double> getTotalAttendanceRate String id ; //오늘일자까지 출석일수와 출석률 출력
    public double getThisMonthBenefit String id ; //이번달 수당 출력
    public int getThisMonthAbscence String id ; //이달 결석일수 출력
    public int getThisMonthAvailableAbscence String id ; //이달 결석가능일수 출력
    public int getTotalAvailableAbscence String id ; //교육기간 전체 결석가능일수 출력(지금까지 결석, 지각 고려)
```


총평

1. 강제영

- 좋았던 점

A. 좋은 팀원과 배웠던 걸 응용 해볼 수 있어서 자바 oop에 대한 이해도가 높아져서 좋았다.

- 아쉬운 점

A. 수업 때 하지 않은 arraylist를 타입으로 하는 필드를 다루는 게 힘들었다.

B. 팀플레이할 때 구체적으로 컨셉과 구성을 함께 모여서 논의해야함을 느꼈다.

2. 박찬영

- 좋았던 점

A. OOP와 코드 구성에 관한 이해도가 이전보다 높아졌다 생각된다. 하지만 좀 더 노력은 필요해 보임

- 아쉬운 점

A. 맡은 부분을 제대로 구현해 내지 못해 아쉬운 부분이 크다.

3. 조용훈

- 좋았던 점

A. 여러명이 하나의 프로그램을 짜면서 기능별로 구현부분을 나누고 변수와 함수를 표준화 하는 작업을 배웠다.

B. GUI는 아니지만 콘솔과 프로젝트를 나누어서 조금이지만 프론트엔드와 백엔드 개발을 체험해본것같다.

- 아쉬운 점

A. 같은 프로그램에 대해서 생각하는 방향이 달라 조율하는 과정이 길었고

의사소통 실수로 다른 사람이 만든 함수를 제대로 활용하지 못한 경우가 있었던것 같아 아쉽다.

4. 박의근

- 좋았던 점

A. 현실의 문제를 어떻게 "자바의 시각"으로 풀어내는지, 그 과정을 볼 수있었다.

B. 수업을 통해 학습하는 것과, 그것을 기반으로 특정 과제를 구현하는 것 간의 차이가 어마어마함을 배웠다.
자바를 학습함에 있어, 그 어떤 자극보다 강한 동기부여가 됐다.

- 아쉬운 점

A. 자바에 대한 기본적인 이해가 현저히 부족하여 팀프로젝트에 전혀 기여하지 못했다.