**탐색적 데이터 분석**

**사례연구**

**평가자료**

**-학생 성적 관리-**

수업명 :java,python 기반의 AI 활용 응용

소프트웨어 개발자 양성과정

제출일 : 2020. 12. 01

작성자 : 유승화

□프로젝트 개요

|  |  |  |
| --- | --- | --- |
| **구분** | **내용** | **비고** |
| **제목** | 학생 성적 관리 |  |
| **프로젝트**  **작성기간** | ~2020.12.01 |  |
| **조원 구성** | 유승화 |  |
| **프로젝트 내용** | 학생들의 기본 정보와 강의, 성적을 관리하는 학교 프로그램을 제작 |  |
| **주요기능** | 학교에 학생을 추가할 수 있게 한다 |  |
| 학교에 강의를 개설할 수 있게 한다 |  |
| 학생의 전공에 따라 다른 채점 방식을 적용한다 |  |
| **학습내용** | Interface를 이용하여 채점 방식을 만든다 |  |
| ArrayList에 수강생의 이름을 저장한다 |  |
| 학교를 singleton으로 만들어 하나만 이용한다 |  |
| Getter, setter를 통해 데이터 입출력을 관리한다 |  |

□최종 보고서

|  |  |  |
| --- | --- | --- |
| **구분** | **내용** | **비고** |
| **제목** | 학생 성적 관리 |  |
| **프로젝트 내용** | 학생들의 기본 정보와 강의, 성적을 관리하는 학교 프로그램을 제작 |  |
| **완료내용** | 학교에 학생을 추가할 수 있게 한다 | 프로젝트  개요와 동일 |
| 학교에 강의를 개설할 수 있게 한다 | 프로젝트  개요와 동일 |
| 학생의 전공에 따라 다른 채점 방식을 적용한다 | 프로젝트  개요와 동일 |
| **프로젝트 활동 후기** | | |
| **좋았던 점** | 객체지향적인 프로그램을 만들기 위해 자료를 찾고 구현해보며 지식이 늘었다 |  |
| **아쉬웠던 점 및**  **어려웠던 점** | 클래스와 변수들이 서로 상호 참조되는 일이 생겨 구조를 여러 번 수정하는 게 어려웠다 |  |
| 채점방식, 전공 등을 상수로 만들어 새로운 요소들이 추가되었을 때 상수 값 추가와 약간의 수정으로도 돌아갈 수 있도록 구현하고 싶었으나 시간이 부족했다 |  |
|  |
| 만들면서 새로운 기능을 이것저것 추가하다 보니 시간도 부족하고 내용들이 정리되지 못했다 |  |
| **학습 효과** | 프로그램 설계, 클래스 간의 관계 등 |  |

□프로젝트 세부 사항 분석 및 구조 설계

|  |  |  |  |
| --- | --- | --- | --- |
| **구분** | **주요 기능/구문** | **내용** | **입출력** |
| **1** | Class | 비슷한 기능을 실행하는 코드들끼리 묶어서 나타냈다(학교,과목,학생) |  |
| **2** | ArrayList | 학생 명단 등을 나타낼 때 사용했다 |  |
| **3** | StringBuilder | 과목 ID를 만들 때 개설연도, 과목 분류, 분반 String을 합쳐 만들었다 |  |
| **4** | Interface | 여러가지 채점 방식을 만들 때 사용했다 |  |

□작성 과정, 실행결과

|  |  |  |
| --- | --- | --- |
| **구현 내용** | **결과** | **비고** |
| 학교, 학생, 과목 클래스를 만들어 내용을 추가,제거하는 메서드를 만들었다 | 완료 |  |
| 전공이나 채점 방식 등을 추가할 수 있도록 하여 객체지향적인 프로그램을 구현한다 | 미완료 | 시간 부족 |

□실행결과-1

텍스트이(가) 표시된 사진

자동 생성된 설명

학생 추가

학번(입학년도+등록순서+전공코드)로 구성

□소스코드-1

학교 클래스--------------------------------------------------------------------------

package studentgrade;

import java.util.ArrayList;

import java.util.HashMap;

import java.util.Scanner;

public class School {

Scanner scanner = new Scanner(System.in);

private static String name="SCHOOL";

private ArrayList<Student> students = new ArrayList<Student>();

private ArrayList<Subject> subjects = new ArrayList<Subject>();

private static School instance=new School();

//---------------------------------------------

private School() {}

public static School getInstance() {

return instance;

}

//--------------------------------------------

public static String getName() {

return name;

}

public static void setName(String name) {

School.name = name;

}

public ArrayList<Student> getStudents() {

return students;

}

public ArrayList<Subject> getSubjects() {

return subjects;

}

//--------------------------------------------

public void addStudents() {

System.out.print("추가할 학생 이름 : ");

String inputName=scanner.nextLine();

System.out.print("입학년도 : ");

String inputYear=scanner.nextLine();

System.out.print("추가할 학생 전공(1:컴퓨터,2:국어) : ");

int inputMajor=Integer.parseInt(scanner.nextLine());

String tempStudentID=inputYear.concat(String.valueOf(inputMajor));//학번(입학년도+등록순서+전공코드)

tempStudentID=tempStudentID.concat(Integer.toString(students.size()));

//----------------------

students.add(new Student(tempStudentID,inputName,String.valueOf(inputMajor)));

}

public void removeStudents(Student s) {

students.remove(s);

}

public void addSubjects(Subject s) {

subjects.add(s);

}

public void removeSubjects(Subject s) {

subjects.remove(s);

}

}

학생 클래스-------------------------------------------------------------------------

package studentgrade;

import java.util.HashMap;

public class Student {

public String studentID;//학번(입학년도+등록순서+전공코드)

public String studentName;//이름

public String majorID;//전공코드

public HashMap<Subject,Score> scoreList = new HashMap<Subject,Score>();

public Student() {}

public Student(String studentID, String studentName, String majorID) {

super();

this.studentID = studentID;

this.studentName = studentName;

this.majorID = majorID;

}

@Override

public String toString() {

return "Student [studentID=" + studentID + ", studentName=" + studentName + ", majorID=" + majorID + "]";

}

public void addScore(Subject sub, Score sco) {

scoreList.put(sub, sco);

}

}

점수 클래스-----------------------------------------------------------------------------

package studentgrade;

import grade.ABgrade;

import grade.PFgrade;

import grade.SABgrade;

public class Score {

private int gradeInt;//채점방식

private int score;//점수

private String grade;//성적(A,B 등)

//---------------------------------

public Score() {}

public Score(int gradeInt, int score) {

super();

this.gradeInt = gradeInt;

this.score = score;

}

//---------------------------------

public int getGradeInt() {

return gradeInt;

}

public int getScore() {

return score;

}

public String getGrade() {

return grade;

}

//---------------------------------

public void setGrade() {

switch (gradeInt) {

case 1:

SABgrade sabGrade = SABgrade.getInstance();

grade=sabGrade.getGrade(score);

break;

case 2:

ABgrade abGrade = ABgrade.getInstance();

grade=abGrade.getGrade(score);

break;

case 3:

PFgrade pfGrade = PFgrade.getInstance();

grade=pfGrade.getGrade(score);

break;

default:

System.out.println("잘못된 채점 방식");

break;

}

}

}

과목 클래스---------------------------------------------------------------------------

package studentgrade;

import java.util.ArrayList;

public class Subject {

private String subjectName;//과목명

private String subjectID="";//수업코드(수업개설연도+과목고유코드+분반)

//수강생명단(학번목록)

private ArrayList<String> studentList = new ArrayList<String>();

private ArrayList<Integer> scoreList = new ArrayList<Integer>();

//-----------------------------------------------------------

public Subject() {}

public Subject(String subjectName, String subjectID) {

super();

this.subjectName = subjectName;

this.subjectID = subjectID;

}

//--------------------------------------------------------

public String getSubjectID() {

return subjectID;

}

public void setSubjectID(SubjectID sID) {

this.subjectID = sID.getSubjectID();

}

public String getSubjectName() {

return subjectName;

}

public void setSubjectName(String subjectName) {

this.subjectName = subjectName;

}

public ArrayList<String> getStudentList() {

return studentList;

}

public ArrayList<Integer> getScoreList() {

return scoreList;

}

//--------------------------------------------------------

public void addStudent(Student s) {

studentList.add(s.studentID);

}

public void deleteStudent(Student s) {

studentList.remove(s.studentID);

}

}

과목코드 클래스-----------------------------------------------------------------------

package studentgrade;

import java.util.Scanner;

public class SubjectID {//과목코드 만들기 위한 클ㄽ

Scanner scanner = new Scanner(System.in);

private String subjectCode;//과목분류(국어,수학 등)

private String year;//수업개설연도

private String division;//분반

private String subjectID="";//수업코드(수업개설연도+과목고유코드+분반)

public void inputSubjectID() {

System.out.print("개설할 수업 진행 연도 : ");

String inputYear = scanner.nextLine();

System.out.print("개설할 수업 과목 분류 코드 : ");

String inputSID = scanner.nextLine();

System.out.print("개설할 분반 : ");

String inputClassNum = scanner.nextLine();

setYear(inputYear);

setDivision(inputClassNum);

setSubjectCode(inputSID);

}

public void makeSubjectID() {

String tempID="";

StringBuilder subjectStringBuilder = new StringBuilder(tempID);

//수업코드(수업개설연도+과목분류+분반)

subjectStringBuilder.append(year);

subjectStringBuilder.append(subjectID);

subjectStringBuilder.append(division);

tempID=subjectStringBuilder.toString();

this.subjectID = tempID;

}

//----------------------------------------------------

public String getSubjectCode() {

return subjectCode;

}

public void setSubjectCode(String subjectCode) {

this.subjectCode = subjectCode;

}

public String getYear() {

return year;

}

public void setYear(String year) {

this.year = year;

}

public String getDivision() {

return division;

}

public void setDivision(String division) {

this.division = division;

}

public String getSubjectID() {

return subjectID;

}

}

채점 인터페이스------------------------------------------------------------------------

package grade;

public interface Grade {

public String getGrade(int score);

}

전공과목채점클래스-------------------------------------------------------------------

package grade;

public class SABgrade implements Grade{

private static SABgrade instance=new SABgrade();

private SABgrade() {}

public static SABgrade getInstance() {

return instance;

}

public String getGrade(int score) {

String gradeString="";

if(score<=100 && score>=95) {

gradeString="S";

}else if(score<95 && score>=90) {

gradeString="A";

}else if(score<90 && score>=80) {

gradeString="B";

}else if(score<80 && score>=70) {

gradeString="C";

}else if(score<70 && score>=60) {

gradeString="D";

}else if(score<60) {

gradeString="F";

}

return gradeString;

}

}

전공X 채점 클래스------------------------------------------------------------------

package grade;

public class ABgrade implements Grade{

private static ABgrade instance=new ABgrade();

private ABgrade() {}

public static ABgrade getInstance() {

return instance;

}

@Override

public String getGrade(int score) {

String gradeString="";

if(score<=100 && score>=90) {

gradeString="A";

}else if(score<90 && score>=80) {

gradeString="B";

}else if(score<80 && score>=70) {

gradeString="C";

}else if(score<70 && score>=55) {

gradeString="D";

}else if(score<55) {

gradeString="F";

}

return gradeString;

}

}

Pass/fail 채점 클래스---------------------------------------------------------------------

package grade;

public class ABgrade implements Grade{

private static ABgrade instance=new ABgrade();

private ABgrade() {}

public static ABgrade getInstance() {

return instance;

}

@Override

public String getGrade(int score) {

String gradeString="";

if(score<=100 && score>=90) {

gradeString="A";

}else if(score<90 && score>=80) {

gradeString="B";

}else if(score<80 && score>=70) {

gradeString="C";

}else if(score<70 && score>=55) {

gradeString="D";

}else if(score<55) {

gradeString="F";

}

return gradeString;

}

}

채점방식 정의--------------------------------------------------------------------------

package define;

//학점산출방식

public enum GradeMethod{

SAB(1),AB(2),PF(3);

private final int value;

private GradeMethod(int value) {

this.value = value;

}

public int getValue() {

return value;

}

}

과목 분류 정의-----------------------------------------------------------------------------

package define;

public enum SubjectCategory{//과목분류

KOREAN(1),MATH(2);

private final int value;

private SubjectCategory(int value) {

this.value = value;

}

public int getValue() {

return value;

}

}

전공 정의----------------------------------------------------------------------------

package define;

public enum Major{//전공

COMPUTER(1),KRLANGUAGE(2);

private final int value;

private Major(int value) {

this.value = value;

}

public int getValue() {

return value;

}

}

실행---------------------------------------------------------------------------------

package studentgrade;

public class RunProgram {

public static void main(String[] args) {

School school = School.getInstance();

school.addStudents();

System.out.println(school.getStudents());

}

}