2016-2학기 프로그램설계방법론 중간고사 준비용 코딩 문제

주제 1 : 경기결과 관리 프로그램 ScoreManager

개요 : 어떤 스포츠의 리그가 4개 팀으로 구성되어 있을 때, 임의의 2개 팀 경기의 결과를 입력받아 이 스코어들을 관리하는 프로그램이다. (첨부된 부분 소스 참조할 것!)

1. Score 클래스 (경기결과)
   데이터 : 경기일련번호(int), 경기일자(String), 홈팀번호(int), 원정팀번호(int), 홈팀점수(int), 원정팀점수(int)
   메소드 : 생성자, 필요한 setter, getter 메소드 등

2. ScoreManager 클래스 (경기관리)
   데이터 : 경기결과객체들(ArrayList<Score>), 팀이름(String[]), 팀별경기수(int[]), 팀별승리경기수(int[]), 팀별패배경기수(int[]), 팀별무승부경기수(int[]), 팀별승점수(int[])
   메소드 : 생성자, 관리작업진행(start), 경기결과등록(addScore), 경기결과수정(modifyScore), 등록된모든경기결과출력(listScores), 경기결과검색(findScores), 경기통계(viewResult), 상대전적(viewMatchScore), 그 외 setter, getter 메소드 등 **필요한 경우 추가하라.**
   단, **경기결과검색은 오버로딩**하여 구현하라.
   findScores(int no) : n이 0~3의 팀번호이며, 해당 팀의 모든 Score를 출력한다.
   findScores(String name) : 특정 팀이름을 파라미터로 해당 팀의 모든 Score를 출력한다.
   static 메소드 : main()에서는 ScoreManager 객체를 하나 만들어 start()를 실행한다.

3. 운영규칙
   - ScoreManager의 start() 메소드는 무한루프를 돌면서 관리작업(스코어등록, 스코어수정, 전체스코어출력, 스코어검색, 경기통계, 상대전적, 종료 등)을 수행하도록 한다.
   - 팀의 개수는 **4팀으로 하되**, start() 수행 시작 후, 팀 이름을 임의로 입력받도록 하라.
   - 스코어등록 : **경기일자를 입력받고, 경기팀 2개를 선택하여 각각의 점수를 입력**받는다. 단, 경기일련번호는 자동으로 부여한다. (시작번호 1), 이때, 팀별 승리/패배/무승부/승점 수에 적절하게 반영하도록 해야 한다. 승점은 승리팀 3점, 무승부팀 1점, 패배팀 0점을 부여함.
   - 스코어수정 : 경기일련번호를 입력받아 해당 경기결과의 **각 팀별 점수만 새로 입력받아** 수정한다. 이때, 팀별 승리/패배/무승부/승점 수에 수정내용이 적절하게 반영하도록 해야 한다.
   - 전체스코어출력 : 현재 등록된 모든 스코어를 화면에 출력하되, 스코어의 각 줄마다 팀번호 대신 팀 이름이 출력되도록 하라. (예 : [8] 2015/03/01 연세곰돌이 4 : 5 한양호랑이)
   - 스코어검색 : 팀번호 또는 팀이름으로 검색할지 물어본 후, 입력받은 내용으로 해당 팀

이 경기한 모든 스코어를 화면에 출력하되, 출력방식은 위의 전체스코어출력과 같은 형식으로 하라. 그리고 마지막 줄에 해당 팀의 승리경기수, 무승부경기수, 패배경기수, 승점수를 출력하라. (예 : 7승 2무 3패, 승점 23점)

- 경기통계 : 4팀의 각각 승리경기수, 무승부경기수, 패배경기수, 승점수를 출력한 후, 승점수가 가장 높은 1위 팀 이름을 출력하라.
- 상대전적 : 임의의 팀 2개를 선택하게 한 후, 이 2개 팀이 진행한 모든 경기에 대해 상대전적을 계산하여 출력하라. (예 : 한양호랑이의 연세곰돌이 상대전적 - 3승 1무 1패)

주제 2 : 통장관리 프로그램 AccountManager

1. BankAccount 클래스
데이터 : 통장번호(int), 예금주이름(String), 통장잔액(int), 신용등급(int)
메소드 : 생성자, 입금(save), 출금(draw), 통장출력(print), 잔액알아내기(getAmount)
그 외 필요한 getter 메소드 등은 임의로 정할 것.

2. AccountManager 클래스
데이터 : 통장객체배열(BankAccount[]), 통장개설갯수
메소드 : 관리작업진행(start), 통장개설(addAccount), 특정통장에입금하기(saveMoney), 특정통장에서출금하기(drawMoney), 통장내역출력(printAccount), 통장정보확인(viewAccount), 송금하기(transferMoney), 모든통장이자지급하기(putInterest) 등
static 메소드 : main()에서는 AccountManager 객체를 하나 만들어 start()를 실행한다.

3. 운영규칙
- AccountManager 의 start() 메소드는 무한루프를 돌면서 관리작업(개설, 입금, 출금, 송금, 이자지급, 통장내역출력, 종료 등)을 수행하도록 한다.
- 신용등급은 1~3등급으로 구분하며, 등급에 따라 이자율이 다르다. (1등급 3%, 2등급 2%, 3등급 1%)
- 통장개설시 통장번호는 자동으로 부여된다. (시작번호 1)
- 통장개설시 예금주이름과 잔액, 신용등급을 입력받아 생성한다.
- 1등급의 경우 마이너스출금이 가능하며, 마이너스 상태인 경우 이자지급이 안된다.
- 2,3등급의 경우 통장잔액이 부족하면 출금이 되지 않는다.
- 송금의 경우 출금통장과 입금통장의 통장번호와 함께 송금액을 입력받아 처리한다.

4. 테스트 순서
- 5개 이상의 통장을 임의로 개설하라. 1개 이상은 1등급으로 지정
- 각 통장별로 적절히 입금, 출금, 송금을 수행하여 AccountManager의 메소드 실행을 확인하라.
- 출금 및 송금 시 잔액이 부족한 상황에서 처리되는 모습을 보여라.
- 마이너스출금이 가능한 통장에서 출금 및 송금이 처리되는 모습을 보여라.

참고 부분소스 (ScoreManager.java)

```
/*
주제 1 : 경기결과 관리 프로그램 ScoreManager
*/

import java.util.Scanner;
import java.util.ArrayList;

class Score
{
	int no, home, away, h_score, a_score; // 경기일련번호(int), 홈팀번호(int), 원정팀번호(int), 홈팀점수(int), 원정팀점수(int)
	String date;	//경기일자(String)
	public Score(int n, int h, int a, int s1, int s2, String d)
	{
		no = n;
		home = h;
		away = a;
		h_score = s1;
		a_score = s2;
		date = d;
	}

}
class ScoreManager
{
	ArrayList<Score> scores;	// 경기결과객체들()
	String teamname[];			// 팀이름()
	int count_game[], count_win[], count_draw[], count_lose[], point[];
// 팀별경기수(int[]), 팀별승리경기수(int[]), 팀별패배경기수(int[]), 팀별무승부경기수(int[]), 팀별승점수(int[])
	int gameno;					//현재 게임번호

	public ScoreManager()
	{
		scores = new ArrayList<Score>();
		count_game = new int[4];
		count_win = new int[4];
		count_draw = new int[4];
		count_lose = new int[4];
		point = new int[4];
		teamname = new String[4];
		gameno = 0;
		for (int i=0;i<4;i++)
		{
			count_game[i]=0;
			count_win[i]=0;
			count_draw[i]=0;
			count_lose[i]=0;
			point[i]=0;
		}
	}

	void printTeamname()
	{
		System.out.print("팀 목록 : ");
		for (int i=0;i<4;i++)
		{
			System.out.print("["+i+"]"+teamname[i] + " ");
		}
		System.out.println();
	}
	//경기결과등록
	void addScore()
	{
		Scanner s = new Scanner(System.in);
		printTeamname();
		System.out.print("경기일자를 입력하세요 : ");
		String d = s.next();
		System.out.print("홈팀 번호와 점수를 입력하세요 : ");
		int h = s.nextInt();
		int s1 = s.nextInt();
```

```
                System.out.print("원정팀 번호와 점수를 입력하세요 : ");
                int a = s.nextInt();
                int s2 = s.nextInt();
                gameno++;
                Score s = new Score(gameno, h, a, s1, s2, d);
        }


        //경기결과수정(modifyScore)
        //등록된모든경기결과출력(listScores)
        //경기결과검색(findScores)
        //경기통계(viewResult)
        //상대전적(viewMatchScore)

    //관리작업진행(start)
        void start()
        {
                int menu, go=true;
                for (int i=0;i<4;i++)
                {
                        System.out.print((i+1) + "번 팀의 이름은?");
                        teamname[i] = s.next();
                }
                while(go)
                {
                        System.out.println("1.스코어등록, 2.스코어수정, 3.전체스코어출력, 4.스코어검색,
5.경기통계, 6.상대전적, 7.종료");
                        System.out.prin("메뉴선택 : ");
                        menu = s.nextInt();
                        switch (menu)
                        {
                        case 1 :
                                addScore();
                                break;
                        case 2 :
                                modifyScore();
                                break;
                        case 3 :
                                modifyScore();
                                break;
                        case 4 :
                                modifyScore();
                                break;
                        case 5 :
                                modifyScore();
                                break;
                        case 6 :
                                modifyScore();
                                break;
                        case 7 :
                                go=false;
                                break;

                        }
                }

        }

        public static void main(String[] args)
        {
                ScoreManager m = new ScoreManager();
                m.start();
        }
}
```