

Corrigé type de l'examen P00

Exercice n°1: (17/17)

class MatchFoot {	3/3
private String equipeA, equipeB;	0.25
private int nbrButA, nbrButB;	0.25
public MatchFoot(String equipeA, String equipeB){	0.5
this.equipeA=equipeA;	
this.equipeB=equipeB;	
}	
public String getEquipeA() { return equipeA;}	0.25
public String getEquipeB() { return equipeB;}	0.25
public int getNbrButA() { return nbrButA;}	0.25
public int getNbrButB() { return nbrButB;}	0.25
public void ajouterBut(String eq){	
if (eq.equals("A")) nbrButA++;	0.5
else if (eq.equals("B")) nbrButB++;	0.5
}	
}	
class Championnat{	12/12
public static final int MAX_MATCH_IN_TAB=40;	
private int nbrMatch; // Nombre de match dans le tableau tabMatches	0.5
private MatchFoot[] tabMatches; // tableaux des matchs du championnat	0.5
private String tabEqPts[][]; // tableau des équipes participantes au championnat	0.5
public Championnat(){	1
tabMatches=new MatchFoot[MAX_MATCH_IN_TAB];	
tabEqPts=new String[10][2];	
}	
public void AjouterMatch(MatchFoot mf){	1
int i=nbrMatch;	
if(i<MAX_MATCH_IN_TAB){	
tabMatches[i]=mf;	
nbrMatch++;	
}	
}	
public void ListeEquipe(){	3
int k=0;	
for(int i=0;i<nbrMatch;i++){	
boolean trouve1=false, trouve2=false;	
int j=0;	
while(tabEqPts[j][0]!=null && !trouve1){	
if(tabMatches[i].getEquipeA().equals(tabEqPts[j][0])) trouve1=true;	
j++;	
}	
j=0;	
while(tabEqPts[j][0]!=null && !trouve2){	
if(tabMatches[i].getEquipeB().equals(tabEqPts[j][0])) trouve2=true;	
j++;	
}	
if (!trouve1) { tabEqPts[k][0]=tabMatches[i].getEquipeA(); k++; }	
if (!trouve2) { tabEqPts[k][0]=tabMatches[i].getEquipeB(); k++; }	
}	
}	
private int pointParEquipe(String eq){	2.5
int nbrPoint=0;	
for(int i=0;i<nbrMatch;i++){	
if(tabMatches[i].getEquipeA().equals(eq)) {	
if(tabMatches[i].getNbrButA()==tabMatches[i].getNbrButB()) nbrPoint++;	
else if(tabMatches[i].getNbrButA()>tabMatches[i].getNbrButB())nbrPoint+=3;	
}	
}	

