

Corrigé type de l'examen POO

Exercice n°1: (17/17)

<pre> class MatchFoot { private String equipeA, equipeB; private int nbrButA, nbrButB; public MatchFoot(String equipeA, String equipeB){ this.equipeA=equipeA; this.equipeB=equipeB; } public String getEquipeA() { return equipeA; } public String getEquipeB() { return equipeB; } public int getNbrButA() { return nbrButA; } public int getNbrButB() { return nbrButB; } public void ajouterBut(String eq){ if (eq.equals("A")) nbrButA++; else if (eq.equals("B")) nbrButB++; } } class Championnat{ public static final int MAX_MATCH_IN_TAB=40; private int nbrMatch; // Nombre de match dans le tableau tabMatchs private MatchFoot[] tabMatchs; // tableaux des matchs du championnat private String[] tabEqPts[][]; // tableau des équipes participant au championnat public Championnat(){ tabMatchs=new MatchFoot[MAX_MATCH_IN_TAB]; tabEqPts=new String[10][2]; } public void AjouterMatch(MatchFoot mf){ int i=nbrMatch; if(i<MAX_MATCH_IN_TAB){ tabMatchs[i]=mf; nbrMatch++; } } public void ListeEquipe(){ 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(tabMatchs[i].getEquipeA().equals(tabEqPts[j][0])) trouve1=true; j++; } j=0; while(tabEqPts[j][0]!=null && !trouve2){ if(tabMatchs[i].getEquipeB().equals(tabEqPts[j][0])) trouve2=true; j++; } if (!trouve1) { tabEqPts[k][0]=tabMatchs[i].getEquipeA(); k++; } if (!trouve2) { tabEqPts[k][0]=tabMatchs[i].getEquipeB(); k++; } } } private int pointParEquipe(String eq){ int nbrPoint=0; for(int i=0;i<nbrMatch;i++){ if(tabMatchs[i].getEquipeA().equals(eq)) { if(tabMatchs[i].getNbrButA()==tabMatchs[i].getNbrButB()) nbrPoint++; else if(tabMatchs[i].getNbrButA()>tabMatchs[i].getNbrButB())nbrPoint+=3; } } } } </pre>	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 2px;">3/3</td><td style="padding: 2px;"></td></tr> <tr> <td style="padding: 2px;">0.25</td><td style="padding: 2px;"></td></tr> <tr> <td style="padding: 2px;">0.25</td><td style="padding: 2px;"></td></tr> <tr> <td style="padding: 2px;">0.5</td><td style="padding: 2px;"></td></tr> <tr> <td style="padding: 2px;">0.25</td><td style="padding: 2px;"></td></tr> <tr> <td style="padding: 2px;">0.5</td><td style="padding: 2px;"></td></tr> <tr> <td style="padding: 2px;">0.5</td><td style="padding: 2px;"></td></tr> <tr> <td style="padding: 2px;">12/12</td><td style="padding: 2px;"></td></tr> <tr> <td style="padding: 2px;">0.5</td><td style="padding: 2px;"></td></tr> <tr> <td style="padding: 2px;">0.5</td><td style="padding: 2px;"></td></tr> <tr> <td style="padding: 2px;">0.5</td><td style="padding: 2px;"></td></tr> <tr> <td style="padding: 2px;">1</td><td style="padding: 2px;"></td></tr> <tr> <td style="padding: 2px;">1</td><td style="padding: 2px;"></td></tr> <tr> <td style="padding: 2px;">3</td><td style="padding: 2px;"></td></tr> <tr> <td style="padding: 2px;">2.5</td><td style="padding: 2px;"></td></tr> </table>	3/3		0.25		0.25		0.5		0.25		0.25		0.25		0.25		0.5		0.5		12/12		0.5		0.5		0.5		1		1		3		2.5	
3/3																																					
0.25																																					
0.25																																					
0.5																																					
0.25																																					
0.25																																					
0.25																																					
0.25																																					
0.5																																					
0.5																																					
12/12																																					
0.5																																					
0.5																																					
0.5																																					
1																																					
1																																					
3																																					
2.5																																					

```

        if(tabMatchs[i].getEquipeB().equals(eq)) {
            if(tabMatchs[i].getNbrButB()==tabMatchs[i].getNbrButA()) nbrPoint++;
            else if(tabMatchs[i].getNbrButB()>tabMatchs[i].getNbrButA())nbrPoint+=3;
        }

    }
    return nbrPoint;
}

public void calculeNbrePoint(){ 1.5
    int i=0;
    while(tabEqPts[i][0]!=null){
        tabEqPts[i][1]=pointParEquipe(tabEqPts[i][0])+""; i++;
    }
}

@Override
public String toString(){ 1.5
    String s="";
    s+="-----\r\n";
    s+="| Equipe | Points |\r\n";
    s+="-----\r\n";
    int i=0;
    while(tabEqPts[i][0]!=null){
        s+="| "+tabEqPts[i][0]+" | // case 0: Equipe
        s+="| "+tabEqPts[i][1]+" | // case 1: Nombre de Points
        s+="|\r\n";
        i++;
    }
    s+="-----";
    return s;
}
}

public class Examen_16_17 { 2/2
    public static void main(String[] args) { 0.25
        Championnat chp=new Championnat();
        MatchFoot mf1=new MatchFoot("Algérie", "France");
        MatchFoot mf2=new MatchFoot("Algérie", "Brésil");
        MatchFoot mf3=new MatchFoot("Brésil", "France"); } 0.25
        mf1.ajouterBut("A");
        mf2.ajouterBut("A");
        mf3.ajouterBut("A"); } 0.25
        chp.AjouterMatch(mf1);
        chp.AjouterMatch(mf2);
        chp.AjouterMatch(mf3); } 0.25
        chp.ListeEquipe(); 0.25
        chp.calculeNbrePoint(); 0.5
        System.out.print(chp); 0.25
    }
}

```

Exercice n°2: (3/3)

Ligne 16:

appel au constructeur A() pour créer un objet référencé par a (ligne 3) :

->aucune instruction: valeur de x par défaut x=1 -----0.5

Ligne 17:

Appel de la méthode f()(ligne 5) :

Affichage : je suis f, je travaille pour A, x=1 -----0.5

Ligne 18:

appel au constructeur B(a) pour créer un objet référencé par b (ligne 10) :

1ere instruction->appel de super(a.x) donc le constructeur A(1) (ligne 4) :

-> x=1-1=0 -----0.75

2eme instruction->y=1-2=-1 -----0.75

Ligne 19:

b.f() -> appel de f() de la classe B (ligne 11)

Affichage : je suis f, je travaille pour B, x=0, y=-1 -----0.5