

## Projet LU2IN002 - 2020-2021

Numéro du groupe de TD/TME : **Groupe 7**

Nom : LIN	Nom : HWANG	Nom :
Prénom : Jérémy	Prénom : Chan Yeong	Prénom :
N° étudiant : 28602292	N° étudiant : 28610448	N° étudiant :

*Thème choisi (en 2 lignes max.)*

Le thème choisi est la simulation d'un match de tennis de table ainsi que d'un tournoi entre professionnels. Le déroulement du match est défini par un système de probabilité.

*Description des classes et de leur rôle dans le programme (2 lignes max par classe)*

- **Joueur.java** : classe mère qui est abstraite, elle sera étendue par les deux classes Amateur.java et Pro.java
- **Amateur.java** : classe qui hérite de joueurs, elles regroupent les joueurs qui n'ont pas de classement et qui n'implémentent pas l'interface Effet.java et qui ne peuvent que faire des matchs classiques.
- **Pro.java** : classe qui hérite de joueurs et qui implémente l'interface Effet.java et qui étend l'exception ClassementException.java, possède un classement minimal qui est d'être au moins en position 100, peut participer aussi bien à des matchs classiques qu'à des tournois.
- **Effet.java** : Interface qui regroupe les effets possibles que les professionnels maîtrisent plus ou moins, qui augmente leurs chances de réussir à marquer le point pendant un match.
- **ProHomme.java** : classe qui hérite de Pro.java , possèdent des méthodes spécifiques aux professionnels qui peuvent augmenter leur probabilités de réussir les smash ainsi que celle de renvoyer correctement la balle durant une période d'entraînement ou de la diminuer si ils ne s'entraînent pas avant le tournoi ou pendant la pause lorsqu'ils ne jouent pas.
- **ProFemme.java** : classe qui hérite de Pro.java , possèdent également des méthodes spécifique aux attributs de la classe Pro.java, augmentation de la précision, de la probabilité de réussir les lifts et les coupés, entraînement avant la compétition...
- **Balle.java** : classe qui est utilisée pour modéliser la balle sur le terrain.
- **ClassementException.java** : classe étendant Exception qui est utilisé dans la classe Pro.java pour vérifier s'il rentre bien dans la catégorie pro.

- **Gestion.java** : classe statique qui modélise un système de paris en début de match.
- **Terrain.java** : classe qui modélise la table de ping pong, fixée et unique.
- **Simulation.java** : classe qui possède les méthodes pour simuler les matchs, les tournois et les entraînements.
- **TestSimulation.java** : classe qui possède le main et qui gère les exceptions.
- **TestTerrain.java** : classe de test d'affichage de la table.

# Schéma UML des classes vision fournisseur (dessin "à la main" scanné ou photo acceptés)



<i>Checklist des contraintes prises en compte:</i>	<i>Nom(s) des classe(s) correspondante(s)</i>
Classe contenant un tableau ou une ArrayList	<ul style="list-style-type: none"> <li>• Terrain.java</li> </ul>
Classe avec membres et méthodes statiques	<ul style="list-style-type: none"> <li>• Gestion.java</li> </ul>
Classe abstraite et méthode abstraite	<ul style="list-style-type: none"> <li>• Joueur.java</li> </ul>
Interface	<ul style="list-style-type: none"> <li>• Effet.java</li> </ul>
Classe avec un constructeur par copie ou clone()	<ul style="list-style-type: none"> <li>• Balle.java</li> </ul>
Définition de classe étendant Exception	<ul style="list-style-type: none"> <li>• Pro.java</li> <li>• ProHomme.java</li> <li>• ProFemme.java</li> </ul>
Gestion des exceptions	<ul style="list-style-type: none"> <li>• TestSimulation.java</li> </ul>
Utilisation du pattern singleton	<ul style="list-style-type: none"> <li>• Terrain.java</li> <li>• TestSimulation.java</li> </ul>

*Présentation de votre projet (max. 2 pages) : texte libre expliquant en quoi consiste votre projet.*

Nous avons choisi de simuler un match de tennis de table. Il y a plusieurs types de joueurs, des amateurs et des professionnels qui sont sous-divisés entre homme et femme.

Les professionnels possèdent un classement qui est augmenté ou diminué selon le résultat du match. Les amateurs ne peuvent pas faire d'effets au cours du match et ne peuvent pas participer à des tournois contrairement aux professionnels qui peuvent participer à des tournois pour augmenter leur classement.

Les professionnels peuvent jouer des matchs entre pro, ou jouer contre des amateurs, pas de restrictions particulières.

Pendant un match avec un professionnel, celui-ci pourra aléatoirement décider d'enclencher un des trois effets disponibles. Cet effet sera choisi par l'utilisateur.

Un match se termine lorsque l'un des joueurs obtient un score de 21. S'il y a 20-20 alors on entre dans une deuce, balle de match.

Il y a également un système de paris au début de chaque match entre les 2 joueurs qui s'affronte, pas de restrictions spécifiques aux amateurs ou pros.

Les pros divisés en catégorie homme et femme peuvent s'entraîner avant la compétition ou avant un match permettant ainsi d'augmenter leurs attributs spécifiques.

Pendant un tournoi, les joueurs qui ne jouent pas sont en pause et peuvent également augmenter leurs attributs pour être plus performant.

*Copier / coller vos classes et interfaces à partir d'ici :*

- **Joueur.java :**

```
/**
 * @author J\u00e9r\u00e9my LIN | Chang Yeong HWANG
 */
public abstract class Joueur{
    protected String nom;
    protected double Pbtaper=0.9;
    protected double Pbprecision=0.5;
    protected int pointTournois;

    protected Joueur(String name){
        nom=name;
        pointTournois=0;
    }

    public String getNom(){
        return nom;
    }
    public String toString(){
        return "Je suis un joueur : "+nom;
    }
    public abstract int getClassement();
    public abstract void setClassement(int a);
    public void setPbtaper(double d){
        Pbtaper=d;
    }

    public void setPbprecision(double d){
        Pbprecision=d;
    }

    public double getPbtaper(){
        return Pbtaper;
    }

    public double getPbprecision(){
        return Pbprecision;
    }

    public void setPointTournois(int a){
        pointTournois=a;
    }

    public int getPointTournois(){
        return pointTournois;
    }
}
```

```

        public abstract boolean renvoyerBien();

        public abstract boolean renvoyerBienPro(Joueur j, boolean coupe, boolean lift,
boolean smash); //joueur adverse en parametre
    }

```

- **Amateur.java :**

```

/**
 * @author J\u00e9my LIN | Chang Yeong HWANG
 */
public class Amateur extends Joueur{

    /**
     * probabilit\u00e9 de taper correctement la balle
     */
    private double Pbtaper=0.5;
    /**
     * probabilit\u00e9 de renvoyer la balle correctement sur le terrain adverse
     */
    private double Pbprecision=0.4;

    /**
     * Constructeur standard a partir d'un nom
     * @param name : String
     */
    public Amateur(String name){
        super(name);
    }

    /**
     * Methode pour retourner un classement excessif, pour les amateurs.
     * @return Retourne arbitrairement 1000
     */
    public int getClassement(){
        return 1000;
    }

    public void setClassement(int a){}

    public String toString(){
        return "Je suis "+nom;
    }

    public String getNom(){
        return nom;
    }
}

```

```

/**
 * Methode pour voir si la balle a pu etre taper correctement ou non
 * Le retour de balle doit etre precis egalement
 */
public boolean renvoyerBien(){
    double p=(Math.random()*(1));
    double t=(Math.random()*(1));
    if(p<=Pbtaper && t<=Pbprecision){
        return true;
    }
    if(p>Pbtaper)
        System.out.println(nom+ " n'a pas pu taper la balle");
    else if(t>Pbprecision)
        System.out.println("Let | Sortie de table");
    return false;
}

/**
 * Methode exclusive aux pro
 * @return Renvoie toujours false
 */
public boolean renvoyerBienPro(Joueur j, boolean coupe, boolean lift, boolean
smash){//joueur adversaire en parametre
    return false;
}
}

```

- **Pro.java :**

```

/**
 * @author J\u00e9my LIN | Chang Yeong HWANG
 */
import java.util.ArrayList;
import java.util.Scanner;
public class Pro extends Joueur implements Effet{
    protected double Pbtaper=0.90;
    protected double Pbprecision=0.80;
    protected double Pbcoupe= 0.45;
    protected double Pblift=0.22;
    protected double Pbsmash=0.15;
    protected double PbEffet=0.30;
    protected int classement;

    public Pro(String name,int classement)throws ClassementException{
        super(name);
        if (classement > 100){
            // Ici moins le classement est élevée plus le joueur est compétent
            throw new ClassementException("Mauvaise categorie, veuillez choisir

```



```

un classement moins \u00e9lev\u00e9e");
    }
    this.classement=classement;
}

public String toString(){
    return "Je suis "+this.getNom();
}

    public int getClassement(){
        return classement;
    }
    public void setClassement(int a){
        this.classement=a;
    }
public boolean coupe(){
    if((Math.random()*(1))>Pbcoupe){
        return true;
    }
    return false;
}

public boolean lift(){
    if((Math.random()*(1))>Pblift){
        return true;
    }
    return false;
}

public boolean smash(){
    if((Math.random()*(1))>Pbsmash){
        return true;
    }
    return false;
}

    public boolean renvoyerBienPro(Joueur j, boolean coupe, boolean lift, boolean
smash){//joueur adversaire en parametre
        double p=(Math.random()*(1));
        double t=(Math.random()*(1));
        if(p <= Pbtaper && t <= Pbprecision){
            double e=(Math.random()*(1-0.2)+0.2);
            if(e <= PbEffet){
                System.out.println(this.getNom()+" choisissez un effet :\n-1
pour le coup\u00e9\n-2 pour le lift\n-3 pour le smash");
                Scanner clavier = new Scanner(System.in);
                int q = Integer.parseInt(clavier.nextLine());
                switch(q){

```

```

        case 1 :
        System.out.println(this.getNom()+" cherche a faire le coup\u00e9");
            if(this.coupe()){
                System.out.println("coup\u00e9 reussi");
                j.setPbtaper(j.getPbtaper()-0.2);
                coupe=true;
            }
            else{
                System.out.println("Effet non r\u00e9ussi");
            }
        return true;
        case 2 :
        System.out.println(this.getNom()+" cherche a faire le lift");
            if(this.lift()){
                System.out.println("lift r\u00e9ussi");
                j.setPbtaper(j.getPbtaper()-0.2);
                lift=true;
            }
            else{
                System.out.println("Effet non r\u00e9ussi");
            }
        return true;

        case 3 :
        System.out.println(this.getNom()+" cherche a faire le smash");
            if(this.smash()){
                System.out.println("smash r\u00e9ussi");
                j.setPbtaper(j.getPbtaper()-0.4);
                smash=true;
            }
            else{
                System.out.println("Effet non r\u00e9ussi");
            }
        return true;
    }
}
}
if(p>Pbtaper)
    System.out.println(this.getNom()+" n'a pas pu taper la balle");
else if(t>Pbprecision)
    System.out.println("Faute pour "+this.getNom()+" La balle est sortie
du terrain");
return false;
}

public boolean renvoyerBien(){
    return true;
}

}

```

- **Effet.java :**

```
/**
 * @author J\u00e9r\u00e9my LIN | Chang Yeong HWANG
 */
public interface Effet{
    //Effets pour les pro
    public boolean lift();
    public boolean smash();
    public boolean coupe();
}
```

- **ProHomme.java :**

```
/**
 * @author J\u00e9r\u00e9my LIN | Chang Yeong HWANG
 */
public class ProHomme extends Pro implements Effet{

    public ProHomme(String name,int classement)throws ClassementException{
        super(name,classement);
    }

    public void entrainementTaper(){
        if(super.Pbtaper<1)
            super.Pbtaper=super.Pbtaper + 0.05;
    }

    public void secherEntrainement(){
        if(Pbtaper>0)
            super.Pbtaper-=0.03;
    }

    public void pause(){
        if(Pbsmash<1)
            super.Pbsmash+=0.01;
    }
    public int getClassement(){
        return super.classement;
    }
    public String toString(){
        return super.nom+ " : classement " + classement;
    }
}
```

- **ProFemme.java :**

```
/**
 * @author J\u00e9r\u00e9my LIN | Chang Yeong HWANG
 */
public class ProFemme extends Pro implements Effet{
    public ProFemme(String name,int classement)throws ClassementException{
        super(name,classement);
    }

    public void entrainementPrecision(){
        super.Pbprecision=super.Pbprecision + 0.10;
    }

    public void secherEntrainement(){
        if(Pbprecision<1)
            super.Pbprecision-=0.05;
    }

    public void pause(){
        if(Pbcoupe<1)
            super.Pbcoupe+=0.01;
        if(Pblift<1)
            super.Pblift+=0.01;
    }
    public int getClassement(){
        return super.classement;
    }

    public String toString(){
        return super.nom+ " : classement " + classement;
    }
}
```

- **Balle.java :**

```
/**
 * @author J\u00e9r\u00e9my LIN | Chang Yeong HWANG
 */
public class Balle{
    private int x,y;

    public Balle(int x, int y){
        this.x=x;
        this.y=y;
    }
}
```

```

        public Balle clone(){
            return new Balle(this.x,this.y);
        }

        public String toString(){
            return "Balle de coordonn\u00e9es (" +x+" ,"+y+" )";
        }

        public int getX(){
            return x;
        }

        public int getY(){
            return y;
        }

        public void moveBalle(int x, int y){
            this.x=x;
            this.y=y;
        }

        public void randomDrop(int maxX, int minX, int maxY, int minY){
            this.x = (int)(Math.random()*(maxX - minX) + minX);
            this.y = (int)(Math.random()*(maxY - minY) + minY);
        }
    }
}

```

- **ClassementException.java :**

```

/**
 * @author J\u00e9r\u00e9my LIN | Chang Yeong HWANG
 */
public class ClassementException extends Exception{
    public ClassementException(String s){
        super(s);
    }
}

```

- **Gestion.java :**

```

/**
 * @author J\u00e9r\u00e9my LIN | Chang Yeong HWANG
 */
import java.util.Scanner;
public class Gestion{
    private static float coteJ1=0;
    private static float coteJ2=0;
    private Gestion(){}
    public static void parisportif(Joueur j1,Joueur j2){
        System.out.println("Voulez-vous parier ? oui / non");
    }
}

```

```

Scanner clavier2 = new Scanner(System.in);
String rep = clavier2.nextLine();
switch (rep){
    case "non" :
        break;
    case "oui" :
        System.out.println("choisissez le montant a parier");
        Scanner clavier = new Scanner(System.in);
        int t = Integer.parseInt(clavier.nextLine());
        if (t<=0){
            System.out.println("Pas de paris sportif encaiss\u00e9e");
            System.exit(0);
        }
        System.out.println("Choisissez le joueur sur qui pari\u00e9e : 1-> "+j1.getNom() +" ou
2-> "+j2.getNom());
        Scanner clavier1 = new Scanner(System.in);
        int joueur = Integer.parseInt(clavier1.nextLine());
        switch(joueur){
            case 1 :
                coteJ1+=t;
                System.out.println("Vous avez pari\u00e9e "+ t+" euros sur "+j1.getNom());
                break;
            case 2 :
                System.out.println("Vous avez pari\u00e9e "+ t+" euros sur "+j2.getNom());
                coteJ2+=t;
                break;
        }
        break;
    }
}
public static float coteTotaleJ1(){
    return coteJ1;
}
public static float coteTotaleJ2(){
    return coteJ2;
}
public static float setCoteJ1(float multiplicateur){
    return coteJ1*=multiplicateur;
}
public static float setCoteJ2(float multiplicateur){
    return coteJ2*=multiplicateur;
}
}

```

- Terrain.java :

```
/**
 * @author J\u00e9r\u00e9my LIN | Chang Yeong HWANG
 */
public class Terrain{
    private static Terrain terrain=null;
    private String[][] table;

    /**
     * Constructeur standard qui cree la table de ping pong fixe
     */
    private Terrain(){
        table= new String[40][70];
        /*la longueur(L) du terrain reel vaut 274cm et sa largeur(l) 152.5cm et
        le diametre d'une balle ping pong est 4cm ainsi lorsque divise L et l par 4
        et par arrondi on obtient 68 cases sur la longueur et 38 cases sur la
        largeur.*/

        //Initialisation par des espaces*/
        for(int i = 0 ; i < table.length ; i++){
            for(int j = 0 ; j < table[0].length ; j++){
                table[i][j]=" ";
            }
        }

        //Les bords
        for(int i = 0 ; i < table[0].length ; i++){
            table[0][i] = "+ ";
            table[(table.length)-1][i] = "+ ";
        }
        for(int j = 1 ; j < (table.length-1) ; j++){
            table[j][0] = "+ ";
            table[j][table[0].length-1] = "+ ";
        }

        //le filet
        for(int k = 1 ; k < (table.length-1) ; k++){
            table[k][34] = "+ ";
            table[k][35] = "+ ";
        }
    }

    public String toString(){
        return "Je suis un Table de ping pong !";
    }
}
```

```

    }

/**
 * Methode utilisant le pattern singleton
 * @return La table de ping pong unique
 */
public static Terrain singletonTerrain(){
    if(terrain==null){
        terrain= new Terrain();
    }
    return terrain;
}

public void afficheTerrain(){
    for(int i = 0 ; i < table.length ; i++){
        for(int j = 0 ; j < table[0].length ; j++){
            System.out.printf(table[i][j]);
        }
        System.out.println("");
    }
}

public void placeBalleInTerrain(Balle b){
    table[b.getY()][b.getX()]="@ ";
}

public void removeBalleOutTerrain(){
    for(int i = 0 ; i < table.length ; i++){
        for(int j = 0 ; j < table[0].length ; j++){
            if(table[i][j]=="@ ")
                table[i][j]=" ";
        }
    }
}
}

```



- **Simulation.java :**

```
/**
 * @author J\u00e9r\u00e9my LIN | Chang Yeong HWANG
 */
import java.util.Scanner;
public class Simulation{
    private static int nbFauteTotale;
    private static Terrain ter;
    /**
     *Score du joueur 1 et du joueur 2
     */
    private int p1,p2;
    private Joueur j1,j2;
    private Balle b;
    private boolean duce;
    /**
     *Effet du joueur 1
     */
    private boolean coupe1,lift1,smash1;
    /**
     *Effet du joueur 2
     */
    private boolean coupe2,lift2,smash2;

    public Simulation(Joueur j1, Joueur j2, Terrain ter){
        this.ter=ter;
        p1=0;p2=0;
        this.j1=j1;this.j2=j2;
        duce=false;
        b=new Balle(1,1);
        nbFauteTotale=0;
        coupe1=false;
        lift1=false;
        smash1=false;
        coupe2=false;
        lift2=false;
        smash2=false;
    }

    public static String getnbFauteTotale(){
        return "Nombre de faute total durant le match : " + nbFauteTotale;
    }

    public void afficheClassement(){
        if(this.j2 != null && this.j1 != null)
```

```

        System.out.println("| Joueurs=\n-"+j1.getNom()+" est "+j1.getClassement()+"eme sur 100
joueurs\n-"+j2.getNom()+" est "+j2.getClassement()+"eme sur 100 joueurs\n");
        else
            System.out.println("Erreur d'affichage");
    }

/**
 *Methode qui place la balle sur le terrain du joueur 1
 */
    public void placementBalleJ1(){
        Balle clone = b.clone();
        clone.randomDrop(36, 69, 1, 39);
        ter.removeBalleOutTerrain();
        ter.placeBalleInTerrain(clone);
        ter.afficheTerrain();
    }

/**
 *Methode qui place la balle sur le terrain du joueur 2
 */
    public void placementBalleJ2(){
        Balle clone = b.clone();
        clone.randomDrop(1, 33, 1, 39);
        ter.removeBalleOutTerrain();
        ter.placeBalleInTerrain(clone);
        ter.afficheTerrain();
    }

/**
 *Methode qui gere le deroulement du match pour le joueur 1
 */
    public void serviceJ1(){
        for(int i=0 ; i<5 ; i++){
            if(p1==21 || p2==21){
                break;
            }
            boolean t1=false;
            if(j1 instanceof Pro){
                t1=j1.renvoyerBienPro(j2,coupe1,lift1,coupe1);
            }
            else{
                t1=j1.renvoyerBien();
            }
            if(!t1){
                nbFauteTotale++;
                //System.out.println("Service rat\u00e9");
                System.out.println("Point pour "+j2.getNom());
                p2++;
                this.placementBalleJ2();
                this.afficheScore();
            }
        }
    }

```

```

        System.out.println("");
        continue;
    }
    else{
        this.placementBalleJ1();
        boolean t2=true;
        while(t1 && t2){
            if(j2 instanceof Pro){
                t2=j2.renvoyerBienPro(j1,coupe2,lift2,smash2);
            }
            else{
                t2=j2.renvoyerBien();
            }
            if(!t2)
                break;
            this.placementBalleJ2();
            if(coupe2 || lift2){
                j2.setPbtaper(j2.getPbtaper()+0.2);
                coupe2=false;
                lift2=false;
            }
            if(smash2){
                j2.setPbtaper(j2.getPbtaper()+0.4);
                smash2=false;
            }
            if(j1 instanceof Pro){
                t1=j1.renvoyerBienPro(j2,coupe1,lift1,smash1);
            }
            else{
                t1=j1.renvoyerBien();
            }
            if(t1){
                this.placementBalleJ1();
            }
            if(coupe1 || lift1){
                j1.setPbtaper(j1.getPbtaper()+0.2);
                coupe1=false;
                lift1=false;
            }
            if(smash1){
                j1.setPbtaper(j1.getPbtaper()+0.4);
                smash1=true;
            }
        }
        if(!t1){
            System.out.println(j2.getNom()+" marque un point");
            p2++;
            this.placementBalleJ1();
            this.afficheScore();
        }
    }
}

```

```

        System.out.println("");
    }
    else if(!t2){
        System.out.println(j1.getNom()+" marque un point");
        p1++;
        this.placementBalleJ2();
        this.afficheScore();
        System.out.println("");
    }
}
if(p1==20 && p2==20){
    System.out.println("Duce!");
    duce=true;
    break;
}
}
if(p1!=21 && p2!=21)
    System.out.println("\nChangement de Service!\n");
else if(p1==21 || p2==21)
    System.out.println("Fin de match");
}

/**
 *Methode qui gere le deroulement du match pour le joueur 2
 */

public void serviceJ2(){
    for(int i=0 ; i<5 ; i++){
        if(p1==21 || p2==21){
            break;
        }
        boolean t2=false;
        if(j2 instanceof Pro){
            t2=j2.renvoyerBienPro(j1,coupe2,lift2,coupe2);
        }
        else{
            t2=j2.renvoyerBien();
        }
        if(!t2){
            nbFauteTotale++;
            //System.out.println("Service rat\u00e9");
            System.out.println("Point pour "+j1.getNom());
            p1++;
            this.afficheScore();
            this.placementBalleJ1();
            System.out.println("");
            continue;
        }
    }
}

```

```

else{
    this.placementBalleJ2();
    boolean t1=true;
    while(t1 && t2){
        if(j1 instanceof Pro){
            t1=j1.renvoyerBienPro(j2,coupe1,lift1,smash1);
        }
        else{
            t1=j1.renvoyerBien();
        }
        if(!t1)
            break;
        this.placementBalleJ1();
        if(coupe1 || lift1){
            j1.setPbtaper(j1.getPbtaper()+0.2);
            coupe1=false;
            lift1=false;
        }
        if(smash1){
            j1.setPbtaper(j1.getPbtaper()+0.4);
            smash1=true;
        }
        if(j2 instanceof Pro){
            t2=j2.renvoyerBienPro(j1,coupe2,lift2,smash2);
        }
        else{
            t2=j2.renvoyerBien();
        }
        if(t2){
            this.placementBalleJ2();
        }
        if(coupe2 || lift2){
            j2.setPbtaper(j2.getPbtaper()+0.2);
            coupe2=false;
            lift2=false;
        }
        if(smash2){
            j2.setPbtaper(j2.getPbtaper()+0.4);
            smash2=false;
        }
    }
    if(!t1){
        System.out.println(j2.getNom()+" marque un point");
        p2++;
        this.afficheScore();
        System.out.println("");
    }
    else if(!t2){
        System.out.println(j1.getNom()+" marque un point");
    }
}

```

```

                p1++;
                this.afficheScore();
                System.out.println("");
            }
        }
        if(p1==20 && p2==20){
            System.out.println("Duce!");
            duce=true;
            break;
        }
    }
    if(p1!=21 && p2!=21)
        System.out.println("\nChangement de Service!\n");
}

```

```

public void afficheScore(){
    System.out.println(j1.getNom()+" : "+ p1 + " | " + p2 + " : " + j2.getNom());
}

```

/\*\*

\* Methode qui permet de jouer s'il y a egalite entre les joueurs

\*@return Le vainqueur

\*/

```

public Joueur duce(Joueur j){
    Joueur c=j;
    while(Math.abs(p1-p2)<2){
        if(c==j1){
            boolean t1=false;
            if(j1 instanceof Pro){
                t1=j1.renvoyerBienPro(j2,coupe1,lift1,coupe1);
            }
            else{
                t1=j1.renvoyerBien();
            }
            if(!t1){
                c=j2;
                System.out.println(j2.getNom()+" marque un point");
                p2++;
                continue;}
            else{
                this.placementBalleJ1();
                System.out.println("Match point!");
                this.afficheScore();
                boolean t2=true;
                while(t1 && t2){
                    if(j2 instanceof Pro){
                        t2=j2.renvoyerBienPro(j1,coupe2,lift2,smash2);
                    }
                }
            }
        }
    }
}

```

```

        else{
            t2=j2.renvoyerBien();
        }
        if(!t2)
            break;
        this.placementBalleJ2();
        if(coupe2 || lift2){
            j2.setPbtaper(j2.getPbtaper()+0.2);
            coupe2=false;
            lift2=false;
        }
        if(smash2){
            j2.setPbtaper(j2.getPbtaper()+0.4);
            smash2=false;
        }
        if(j1 instanceof Pro){
            t1=j1.renvoyerBienPro(j2,coupe1,lift1,smash1);
        }
        else{
            t1=j1.renvoyerBien();
        }
        if(t1){
            this.placementBalleJ1();
        }
        if(coupe1 || lift1){
            j1.setPbtaper(j1.getPbtaper()+0.2);
            coupe1=false;
            lift1=false;
        }
        if(smash1){
            j1.setPbtaper(j1.getPbtaper()+0.4);
            smash1=true;
        }
    }
    if(!t1){
        System.out.println(j2.getNom()+" marque un point");
        p2++;
        this.afficheScore();
    }
    else if(!t2){
        System.out.println(j1.getNom()+" marque un point");
        p1++;
        this.afficheScore();
    }
    c=j2;
}
}
if(c==j2){
    boolean t2=false;

```

```

        if(j2 instanceof Pro){
            t2=j2.renvoyerBienPro(j1,coupe2,lift2,smash2);
        }
        else{
            t2=j2.renvoyerBien();
        }
        if(!t2){
            c=j1;
            System.out.println(j1.getNom()+" marque un point");
            p1++;
            continue;
        }
        else{
            this.placementBalleJ2();
            System.out.println("Match point!");
            this.afficheScore();
            boolean t1=true;
            while(t1 && t2){
                if(j1 instanceof Pro){
                    t1=j1.renvoyerBienPro(j2,coupe1,lift1,smash1);
                }
                else{
                    t1=j1.renvoyerBien();
                }
                if(!t1)
                    break;
                this.placementBalleJ1();
                if(coupe1 || lift1){
                    j1.setPbtaper(j1.getPbtaper()+0.2);
                    coupe1=false;
                    lift1=false;
                }
                if(smash1){
                    j1.setPbtaper(j1.getPbtaper()+0.4);
                    smash1=true;
                }
                if(j2 instanceof Pro){
                    t2=j2.renvoyerBienPro(j1,coupe2,lift2,smash2);
                }
                else{
                    t2=j2.renvoyerBien();
                }
                if(t2){
                    this.placementBalleJ2();
                }
                if(coupe2 || lift2){
                    j2.setPbtaper(j2.getPbtaper()+0.2);
                    coupe2=false;
                    lift2=false;
                }
            }
        }
    }
}

```



```

        }
        if(smash2){
            j2.setPbtaper(j2.getPbtaper()+0.4);
            smash2=false;
        }
    }
    if(!t1){
        System.out.println(j2.getNom()+" marque un point");
        p2++;
        this.afficheScore();
    }
    else if(!t2){
        System.out.println(j1.getNom()+" marque un point");
        p1++;
        this.afficheScore();
    }
    c=j1;
}

}

}
if(p1<p2){return j2;}
return j1;
}

/**
 *Methode qui execute un match classique
 *@return Le vainqueur
 */
public Joueur matchClassique(){
    Joueur res = null;
    this.afficheClassement();
    Gestion.pariSportif(j1,j2);
    System.out.println("\n\n-----\n\n");
    System.out.println("Choix du service : \n-nombre pair -> Joueur1\n-nombre impair -
> Joueur2\n");
    Scanner clavier = new Scanner(System.in);
    int t = Integer.parseInt(clavier.nextLine());
    System.out.println("\n\n Debut du Match ! \n\n");
    try{
        Thread.sleep(2000);
    }
    catch(InterruptedException ex){
        Thread.currentThread().interrupt();
    }
    while(p1<21 && p2<21){
        if(t%2==0){
            try{
                Thread.sleep(800);
            }

```

```

        catch(InterruptedException ex){
            Thread.currentThread().interrupt();
        }
        this.serviceJ2();
        if(duce){
            res=this.duce(j1);
            break;
        }
        t++;
    }
    if(t%2==1){
        try{
            Thread.sleep(800);
        }
        catch(InterruptedException ex){
            Thread.currentThread().interrupt();
        }
        this.serviceJ1();
        if(duce){
            res=this.duce(j2);
            break;
        }
        t++;
    }
}
if(p2<p1){
    System.out.println("La vainqueur de ce match est "+j1.getNom());
    j1.setClassement(j1.getClassement()-1);
    j2.setClassement(j2.getClassement()+1);
    res=j1;
    Gestion.setCoteJ1(2);
}
if(p2>p1){
    System.out.println("La vainqueur de ce match est "+j2.getNom());
    j2.setClassement(j2.getClassement()-1);
    j1.setClassement(j1.getClassement()+1);
    res=j2;
    Gestion.setCoteJ2(2);
}
this.afficheClassement();
return res;
}

```

/\*\*

\*Methode qui permet d'ordonne les joueurs selon leur classement

\*/

```

    public static Joueur[] ordonneList(Joueur[] tab){
        Joueur[] res=tab;
        for(int i=0;i<res.length;i++){

```

```

        for(int j=i;j<res.length;j++){
            if(res[i].getPointTournois()>res[j].getPointTournois()){
                Joueur temp=res[i];
                res[i]=res[j];
                res[j]=temp;
            }
        }
    }
    return res;
}

/**
 *Methode qui execute un tournoi
 */
    public void tournois(Joueur[] tabl, Terrain t){
        System.out.println("\n\n-----\n\n");
        Joueur[] tab=tabl;
        for(int test=0;test<tab.length;test++){
            if(tab[test] instanceof Amateur){
                System.out.println("Il y a un amateur dans les
participants. Impossible de debuter le tournoi");
                System.exit(0);
            }
        }
        System.out.println("\n\n Debut du Tournois ! \n\n");
        for(int i=0;i<tab.length;i++){
            for(int j=i;j<tab.length;j++){
                if(j==i)
                    continue;
                Simulation temp= new Simulation(tab[i],tab[j],t);

                for(int k=0;k<tab.length;k++){

                    if(k==i || k==j){
                        continue;
                    }

                    if(tab[k] instanceof ProHomme){
                        ((ProHomme)tab[k]).pause();

                        System.out.println(((ProHomme)tab[k]).getNom()+ " ne joue pas et prend une pause,
augmentation de la probabilit\u00e9 de smash de 0.01");
                    }

                    else if(tab[k] instanceof
ProFemme){

                        ((ProFemme)tab[k]).pause();

                        System.out.println(((ProFemme)tab[k]).getNom()+ " ne joue pas et prend une pause,
augmentation de la probabilit\u00e9 de r\u00e9ussir le lift et le coup\u00e9 de 0.01");
                    }
                }
            }
        }
    }
}

```

```

    }
    Joueur q=temp.matchClassique();
    if(q==tab[i])
        tab[i].setPointTournois(tab[i].getPointTournois()+1);
    if(q==tab[j])
        tab[j].setPointTournois(tab[j].getPointTournois()+1);
    }
}
Joueur[] result=Simulation.ordonneList(tab);
for(int k=1 ; k< result.length+1; k++){
    if(k==1){
        System.out.println("Le "+k+"er est "+result[k-1].getNom()+", il a gagne "+result[k-1].getPointTournois()+ " points et son classement final est "+result[k-1].getClassement());
    }
    else{
        System.out.println("Le "+k+"eme est "+result[k-1].getNom()+", il a gagne "+result[k-1].getPointTournois()+ " points et son classement final est "+result[k-1].getClassement());
    }
}
}
}
/**
 *Methode qui permet de s'entrainer avant le jour J
 */
public void entrainementAvantCompétition(int jours, Pro joueur){
    System.out.println(jours + " jours avant le match de "+joueur.getNom());
    for(int i=1; i<jours ; i++){
        System.out.println("Jour "+i);
        try{
            Thread.sleep(200);
        }
        catch(InterruptedException ex){
            Thread.currentThread().interrupt();
        }
        if (joueur instanceof ProHomme){
            if (Math.random()*1< 0.5){
                ((ProHomme)joueur).entrainementTaper();
                System.out.println(joueur.getNom()+" entraine sa probabilité
de taper correctement la balle de 0.05");
            }
            else{
                ((ProHomme)joueur).secherEntrainement();
                System.out.println(joueur.getNom()+" seche l'entrainement,
sa probabilit\u00e9 de taper correctement balle diminue de 0.03 !");
            }
        }
        else if(joueur instanceof ProFemme){
            if (Math.random()*1< 0.5){
                ((ProFemme)joueur).entrainementPrecision();
                System.out.println(joueur.getNom()+" augmentation de la

```

```

pr\u00e9cision de 0.1");
        }
        else{
            ((ProFemme)joueur).secherEntrainement();
            System.out.println(joueur.getNom()+" seche l'entrainement et
perds 0.05 de pr\u00e9cision !");
        }
    }
}
System.out.println("\n\n");
}
}

```

- **TestSimulation.java :**

```

/**
 * @author J\u00e9my LIN | Chang Yeong HWANG
 */
public class TestSimulation{
    public static void main(String[] args){
        System.out.println("Veuillez mettre le terminal en plein \u00e9cran");
        try{
            Thread.sleep(2000);
        }
        catch(InterruptedException ex){
            Thread.currentThread().interrupt();
        }
        Terrain table = Terrain.singletonTerrain();
        Joueur[] tab = new Joueur[4];
        try{
            Amateur j1 = new Amateur("Chan");
            ProFemme j2 = new ProFemme("Kimmy",11);
            ProHomme j3 = new ProHomme("Kevin",15);
            ProHomme j4 = new ProHomme("Tony",18);
            ProFemme j5 = new ProFemme("Jenny",25);
            Simulation match= new Simulation(j2,j3,table);
            match.entrainementAvantCompétition(7,j2);
            match.entrainementAvantCompétition(3,j3);
            tab[0]=j2;
            tab[1]=j3;
            tab[2]=j4;
            tab[3]=j5;
            match.matchClassique();
            System.out.println(match.getnbFauteTotale());
            System.out.println("La cote totale pour "+j2.getNom()+" est de
"+Gestion.coteTotaleJ1()+" euros");
            System.out.println("La cote totale pour "+j3.getNom()+" est de

```

```

"+Gestion.coteTotaleJ2()+ " euros");
    //match.tournois(tab,table);
}
catch(ClassementException e){
    System.out.println(e.toString());
}
}
}
}

```

- **TestTerrain.java :**

```

/**
    * @author J\u00e9r\u00e9my LIN | Chang Yeong HWANG
 */
public class TestTerrain{
    public static void main(String[] args){
        Terrain t= Terrain.singletonTerrain();
        t.afficheTerrain();
        Balle b1= new Balle(4,17);
        t.placeBalleInTerrain(b1);
        t.afficheTerrain();
        t.removeBalleOutTerrain();
        t.afficheTerrain();
    }
}

```