

PPE3 Mission n°1-Etape n°4

Documentation Technique :

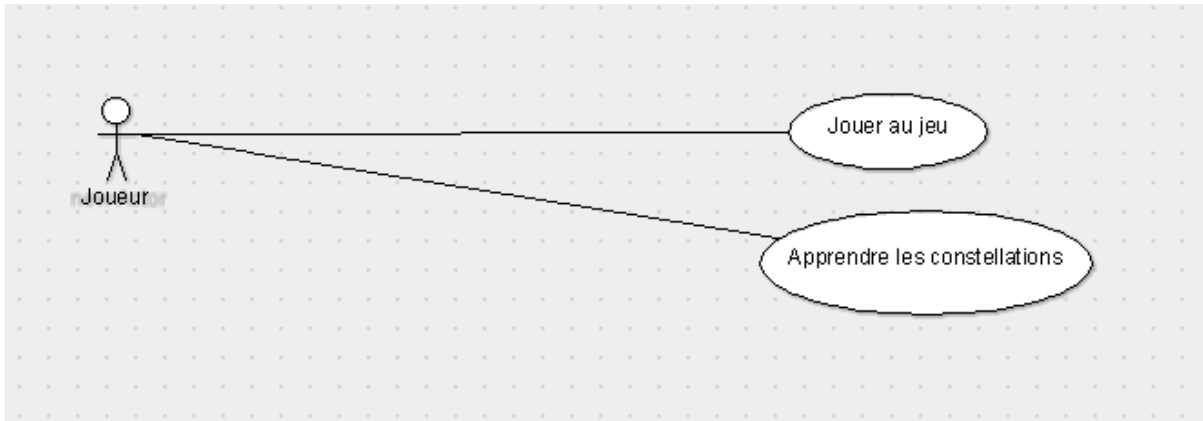


Figure 1 Diagramme de cas d'utilisation de l'application Jeu Sérieux

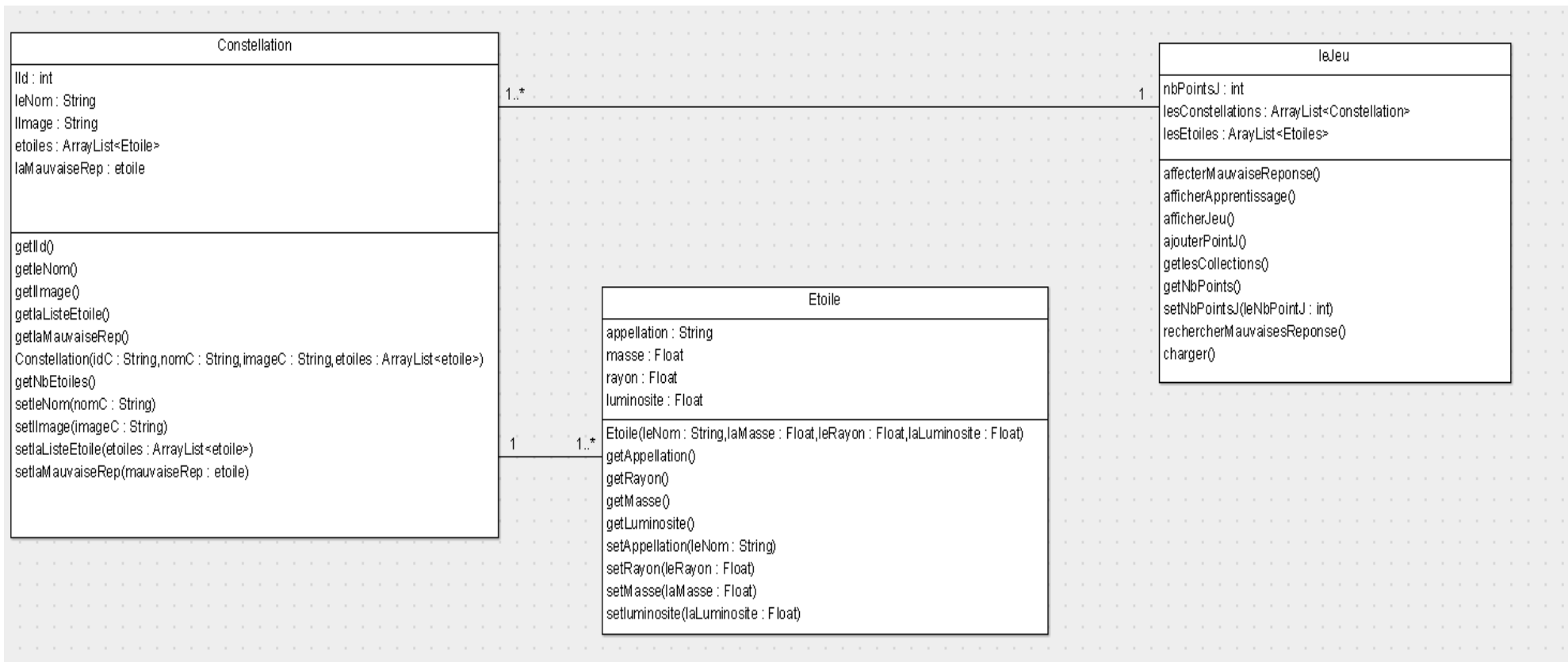


Figure 2 Diagramme de classe de l'application Jeu Sérieux

Fichier constellation.xml complété :

```
<?xml version="1.0" encoding="UTF-8"?>
<ciel>
  <constellation id="1">
    <nom>Lion</nom>
    <image>lion.gif</image>
    <lesetoiles>
      <etoile masse="2.3" rayon="3.6" >
        <appellation>Denebola </appellation>
        <luminosite>60</luminosite>
      </etoile>
      <etoile masse="2.2" rayon="2" >
        <appellation>Zosma </appellation>
        <luminosite>23</luminosite>
      </etoile>
      <etoile masse="2.9" rayon="4.3" >
        <appellation>Chertan </appellation>
        <luminosite>120</luminosite>
      </etoile>
      <etoile masse="3.5" rayon="3.15">
        <appellation>Regulus </appellation>
        <luminosite>150</luminosite>
      </etoile>
    </lesetoiles>
  </constellation>
  <constellation id="2">
    <nom>Gemeaux</nom>
    <image>gemeaux.gif</image>
    <lesetoiles>
      <etoile masse="1.86" rayon="8.8" >
        <appellation>Pollux </appellation>
        <luminosite>32</luminosite>
      </etoile>
      <etoile masse="2.15" rayon="2.3" >
        <appellation>Castor </appellation>
        <luminosite>30</luminosite>
      </etoile>
      <etoile masse="2.8" rayon="4.4" >
        <appellation>Alhena </appellation>
        <luminosite>160</luminosite>
      </etoile>
      <etoile masse="2.8" rayon="4.4" >
        <appellation>Mebсутa </appellation>
        <luminosite>160</luminosite>
      </etoile>
    </lesetoiles>
  </constellation>

```

```
  <constellation id="5">
    <nom>Taureau</nom>
    <image>taureau.jpg</image>
    <lesetoiles>
      <etoile masse="4.5" rayon="6" >
        <appellation>El Nath </appellation>
        <luminosite>700</luminosite>
      </etoile>
      <etoile masse="2.7" rayon="12.6" >
        <appellation>Ain </appellation>
        <luminosite>97</luminosite>
      </etoile>
      <etoile masse="1.13" rayon="45.1" >
        <appellation>Aldebaran </appellation>
        <luminosite>425</luminosite>
      </etoile>
      <etoile masse="2.7" rayon="13.4">
        <appellation>Hyadum </appellation>
        <luminosite>85</luminosite>
      </etoile>
    </lesetoiles>
  </constellation>
  <constellation id="6">
    <nom>Cygne</nom>
    <image>cygne.jpg</image>
    <lesetoiles>
      <etoile masse="20" rayon="220" >
        <appellation>Deneb </appellation>
        <luminosite>60000</luminosite>
      </etoile>
      <etoile masse="3.15" rayon="4.7" >
        <appellation>Delta </appellation>
        <luminosite>180</luminosite>
      </etoile>
      <etoile masse="0" rayon="0" >
        <appellation>Sadr </appellation>
        <luminosite>0</luminosite>
      </etoile>
      <etoile masse="5" rayon="0">
        <appellation>Eta </appellation>
        <luminosite>950</luminosite>
      </etoile>
    </lesetoiles>
  </constellation>
</ciel>
```

```
  <constellation id="3">
    <nom>Lyre</nom>
    <image>lyre.jpg</image>
    <lesetoiles>
      <etoile masse="0" rayon="0" >
        <appellation>Eta Lyrae</appellation>
        <luminosite>0</luminosite>
      </etoile>
      <etoile masse="2.11" rayon="1.9" >
        <appellation>Vega </appellation>
        <luminosite>37</luminosite>
      </etoile>
      <etoile masse="0.3" rayon="15.2" >
        <appellation>Beta </appellation>
        <luminosite>6500</luminosite>
      </etoile>
      <etoile masse="0" rayon="0">
        <appellation>Gamma </appellation>
        <luminosite>0</luminosite>
      </etoile>
    </lesetoiles>
  </constellation>
  <constellation id="4">
    <nom>Orion</nom>
    <image>orion.jpg</image>
    <lesetoiles>
      <etoile masse="16" rayon="0" >
        <appellation>Meissa </appellation>
        <luminosite>63000</luminosite>
      </etoile>
      <etoile masse="9" rayon="5.7" >
        <appellation>Bellatrix </appellation>
        <luminosite>4000</luminosite>
      </etoile>
      <etoile masse="7.7" rayon="1180" >
        <appellation>Betelgeuse </appellation>
        <luminosite>6300</luminosite>
      </etoile>
      <etoile masse="28" rayon="20">
        <appellation>Alnitak </appellation>
        <luminosite>100000</luminosite>
      </etoile>
    </lesetoiles>
  </constellation>

```

Code Java commenté de la classe constellation :

```
package LesClasses;
import java.util.ArrayList;
/**
 * <p>
 * Cette classe mémorise en mémoire centrale (RAM) une constellation caractérisée par les données suivantes :<br/>
 * <ul>
 * <li> L'identifiant d'une constellation</li>
 * <li> Le nom d'une constellation</li>
 * <li> Le chemin de l'image d'une constellation</li>
 * <li> Le nom d'une étoile qui n'appartient pas à la constellation</li>
 * <li> Une collection d'objets de la classe étoile</li>
 * </ul>
 * </p>
 * @author Benjamin LARTIGUE - Julien VANELIAN
 */
public class Constellation {
    /**
     * lId : Identifiant de la Constellation
     * @see Constellation#getId()
     * @see Constellation#setId(int)
     */
    private int lId;
    /**
     * leNom : Nom de la constellation
     * @see Constellation#getleNom()
     * @see Constellation#setleNom(java.lang.String)
     */
    private String leNom;
    /**
     * lImage : Chemin d'accès de l'image de la constellation
     * @see Constellation#getlImage()
     * @see Constellation#setlImage(java.lang.String)
     */
    private String lImage;
    /**
     * laMauvaiseRep : Objet de la classe Etoile qui n'appartient pas à la collection
     * @see Constellation#getlaMauvaiseRep()
     * @see Constellation#setlaMauvaiseRep(LesClasses.Etoile)
     */
}
```

```

private Etoile laMauvaiseRep;
/**
 * laListeEtoile : collection d'étoiles appartenant à la constellation
 * @see Constellation#getLaListeEtoiles()
 * @see Constellation#setLaListeEtoiles(java.util.ArrayList)
 */
private ArrayList<Etoile> laListeEtoiles;

/**
 * Constructeur de la classe Constellation
 * @param idC : identifiant de la constellation
 * @param nomC : identifiant de la constellation
 * @param imageC : chemin d'accès de l'image de la constellation
 * @param etoiles : collection d'étoiles
 */
public Constellation(int idC, String nomC, String imageC, ArrayList<Etoile> etoiles){
    lId = idC;
    leNom = nomC;
    lImage = imageC;
    laMauvaiseRep = null;
    laListeEtoiles = etoiles;
}

/**
 * Accesseur en consultation de la donnée privée lId
 * @return lId
 */
public int getlId() {
    return lId;
}

/**
 * Accesseur en modification de la donnée privée lId
 * @param leNvId : Le nouvel id de la constellation
 */
public void setlId(int leNvId) {
    lId = leNvId;
}

```

```
/**
 * Accesseur en consultation de la donnée privée leNom
 * @return leNom
 */
public String getleNom() {
    return leNom;
}
/**
 * Accesseur en modification de la donnée privée leNom
 * @param leNvNom : Le nouveau nom de la constellation
 */
public void setleNom(String leNvNom) {
    leNom = leNvNom;
}
/**
 * Accesseur en consultation de la donnée privée lImage
 * @return lImage
 */
public String getlImage() {
    return lImage;
}
/**
 * Accesseur en modification de la donnée privée lImage
 * @param laNvImage : Le nouveau chemin vers l'image
 */
public void setlImage(String laNvImage) {
    lImage = laNvImage;
}
/**
 * Accesseur en consultation de la donnée privée laListeEtoiles
 * @return laListeEtoiles
 */
public ArrayList<Etoile> getLaListeEtoiles() {
    return laListeEtoiles;
}
}
```

```

/**
 * Accesseur en modification de la donnée privée LaListeEtoiles
 * @param laNvListeEtoiles : La nouvelle liste d'étoiles
 */
public void setLaListeEtoiles(ArrayList<Etoile> laNvListeEtoiles) {
    laListeEtoiles = laNvListeEtoiles;
}

/**
 * Accesseur en consultation de la donnée privée laMauvaiseRep
 * @return laMauvaiseRep
 */
public Etoile getlaMauvaiseRep() {
    return laMauvaiseRep;
}

/**
 * Accesseur en modification de la donnée privée laMauvaiseRep
 * @param laNvMauvaiseRep : La nouvelle mauvaise réponse
 */
public void setlaMauvaiseRep(Etoile laNvMauvaiseRep) {
    laMauvaiseRep = laNvMauvaiseRep;
}

/**
 * Méthode de type fonction qui calcule et renvoie (entier) le nombres d'étoiles dans la constellation
 * @return nbEtoiles
 */
public int getNbEtoiles(){
    int nbEtoiles = 0 ;
    for (Etoile recupEtoile:laListeEtoiles)
    {
        nbEtoiles++;
    }
    return nbEtoiles;
}
}

```

Code Java commenté de la classe Etoile :

```
package LesClasses;

/**
 * <p>
 * Cette classe mémorise en mémoire centrale (RAM) une étoile caractérisée par les données suivantes :<br/>
 * <ul>
 * <li> L'appellation d'une étoile</li>
 * <li> La masse de l'étoile</li>
 * <li> Le rayon de l'étoile</li>
 * <li> La luminosité de l'étoile</li>
 * </ul>
 * </p>
 * @author Benjamin LARTIGUE - Julien VANELIAN
 */

public class Etoile {

    /**
     * appellation : Nom de l'étoile
     * @see Etoile#getAppellation()
     * @see Etoile#setAppellation(java.lang.String)
     */
    String appellation;

    /**
     * masse : Masse de l'étoile
     * @see Etoile#getMasse()
     * @see Etoile#setMasse(float)
     */
    float masse;

    /**
     * rayon : Rayon de l'étoile
     * @see Etoile#getRayon()
     * @see Etoile#setRayon(float)
     */
    float rayon;

    /**
     * luminosite : Luminosité de l'étoile
     * @see Etoile#getLuminosite()
     * @see Etoile#setLuminosite(float)
     */
}
```



```
float luminosite;

/**
 * Constructeur de la classe Etoile
 *
 * @param leNom : Nom de l'étoile
 */
public Etoile(String leNom) {
    appellation = leNom;
    masse = 0;
    rayon = 0;
    luminosite = 0;
}

/**
 *
 * @param leNom : Nom de l'étoile
 * @param laMasse : La masse de l'étoile
 * @param leRayon : Le rayon l'étoile
 * @param laLuminosite : La luminosite de l'étoile
 */
public Etoile(String leNom, float laMasse, float leRayon, float laLuminosite) {
    appellation = leNom;
    masse = laMasse;
    rayon = leRayon;
    luminosite = laLuminosite;
}

/**
 * Accesseur en consultation de la donnée privée appellation
 * @return appellation
 */
public String getAppellation() {
    return appellation;
}
```

```

* Accesseur en modification de la donnée privée appellation
* @param laNvappellation : La nouvelle appellation
*/
public void setAppellation(String laNvappellation) {
    appellation = laNvappellation;
}
/**
* Accesseur en consultation de la donnée privée masse
* @return masse
*/
public float getMasse() {
    return masse;
}
/**
* Accesseur en modification de la donnée privée masse
* @param laNvmasse : La nouvelle masse
*/
public void setMasse(float laNvmasse) {
    masse = laNvmasse;
}
/**
* Accesseur en consultation de la donnée privée rayon
* @return rayon
*/
public float getRayon() {
    return rayon;
}
/**
* Accesseur en modification de la donnée privée rayon
* @param leNvrayon : La nouveau rayon
*/
public void setRayon(float leNvrayon) {
    rayon = leNvrayon;
}

/**
* Accesseur en consultation de la donnée privée luminosité
* @return luminosité
*/
public float getLuminosite() {
    return luminosite;
}
/**
* Accesseur en consultation de la donnée privée luminosité
* @param laNvluminosite : La nouvelle luminosité
*/
public void setLuminosite(float laNvluminosite) {
    luminosite = laNvluminosite;
}

```

Code Java commenté de la classe LeJeu :

```
package LesClasses;
import java.io.File;
import java.util.ArrayList;
import java.util.Iterator;
import java.util.List;
import java.util.Random;
import javax.swing.JOptionPane;
import lesIHM.FenetreAffichageConstellation;
import lesIHM.FenetreJeuConstellation;
import org.jdom2.Element;
import org.jdom2.input.SAXBuilder;
/**
 * <p>
 * Cette classe mémorise en mémoire centrale (RAM) le jeu, qui utilise les données suivantes :<br/>
 * <ul>
 * <li> La liste des constellations</li>
 * <li> La liste des étoiles</li>
 * <li> le nombre de bonnes réponses</li>
 * </ul>
 * </p>
 * @author Benjamin LARTIGUE - Julien VANELIAN
 */
public class LeJeu {
    /**
     * LesConstellations : La liste des constellations
     * @see LeJeu#getLesConstellations()
     */
    private static ArrayList<Constellation>LesConstellations;
    /**
     * LesEtoiles : La liste des étoiles
     */
    private static ArrayList<Etoile> LesEtoiles;
    /**
     * nbPointsJ : Nombre de réponses justes
     * @see LeJeu#getNbPointsJ()
     * @see LeJeu#setNbPointsJ(int)
     */
    private static int nbPointsJ;
```

```
* Méthode de type procédure qui valorise les objets constellations et étoiles à partir du fichier XML
* @throws Exception
*/
public static void charger() throws Exception
{
    LesConstellations = new ArrayList<Constellation>();

    Element racine;
    org.jdom2.Document document;

    Etoile objEtoile;
    Constellation objConstellation;

    int idConstell;
    String nomConstell;
    String imgConstell;
    String nomEtoile;
    Float lumiEtoile;
    Float masseEtoile;
    Float rayonEtoile;

    Element lesEtoiles;
    Iterator i, j;

    List lesNoeudsConstellations;
    List lesNoeudsEtoiles;

    SAXBuilder sxb=new SAXBuilder();
    document=sxb.build(new File("constellation.xml"));

    racine=document.getRootElement();

    lesNoeudsConstellations=racine.getChildren("constellation");
    i = lesNoeudsConstellations.iterator();
```

```

// Parcours de tous les noeuds<constellation>
// la méthode hasNext renvoie true s'il y a encore un élément à parcourir
while(i.hasNext())
{
    Element laConstellation = (Element)i.next(); // Incrémente l'itérateur i (Tant qu'il le peut)
    // Récupération des valeurs contenues dans la balise <nom>
    // puis dans l'attribut id
    LesEtoiles = new ArrayList<Etoile>();
    lesEtoiles = laConstellation.getChild("lesetoiles");
    lesNoeudsEtoiles = lesEtoiles.getChildren(); //On récupère toutes les étoiles
    j = lesNoeudsEtoiles.iterator(); //Creation de l'itérateur j
    while(j.hasNext())
    {
        Element lEtoile = (Element)j.next(); // Incrémente l'itérateur j (Tant qu'il le peut)
        nomEtoile = lEtoile.getChildText("appellation");
        lumiEtoile = Float.parseFloat(lEtoile.getChildText("luminosite"));
        masseEtoile = lEtoile.getAttribute("masse").getFloatValue();
        rayonEtoile = lEtoile.getAttribute("rayon").getFloatValue();
        objEtoile = new Etoile(nomEtoile, masseEtoile, rayonEtoile, lumiEtoile);

        LesEtoiles.add(objEtoile);

    }
    nomConstell = laConstellation.getChild("nom").getText();
    idConstell = laConstellation.getAttribute("id").getIntValue();
    imgConstell = laConstellation.getChild("image").getText();

    objConstellation = new Constellation(idConstell, nomConstell, imgConstell, LesEtoiles);

    LesConstellations.add(objConstellation);
}
}

Accesseur en consultation de la donnée privée LesConstellations
@return LesConstellations
/

```

```

    /
public static ArrayList<Constellation> getLesConstellations()
{
    return LesConstellations;
}
/**
 * Méthode de type fonction qui recherche les mauvaises réponses et les renvoie dans un ArrayList contenant des Etoi
 * @return lesMauvaisesRep
 */
private static ArrayList <Etoile> rechercherMauvaisesReponses() {
    ArrayList <Etoile> lesMauvaisesRep;
    int taille, numConstellationAlea, numEtoileAlea;
    Constellation objConstelAlea=null;
    Etoile objEtoileAlea;
    taille= LesConstellations.size();
    lesMauvaisesRep=new ArrayList<Etoile>();

    Random nbRand = new Random();
    int i=0;
    for (Constellation recupConstellation:LesConstellations)
    {
        do
        {
            numConstellationAlea = nbRand.nextInt(taille);
        }while(!(numConstellationAlea != i));

        objConstelAlea = LesConstellations.get(numConstellationAlea);

        numEtoileAlea = nbRand.nextInt(4);

        objEtoileAlea=objConstelAlea.getLaListeEtoiles().get(numEtoileAlea);
        System.out.println(objEtoileAlea.getAppellation());
        lesMauvaisesRep.add(objEtoileAlea);
        i++;
    }
}

```

Return lesMauvaisesReponses ;

}

```

/**
 * Méthode de type procedure qui utilise rechercherMauvaisesReponses() et les affecte aux const
 */
public static void affecterMauvaisesReponses()
{
    ArrayList<Etoile> CollectionMauvaisesEtoiles;
    int numEtoileAlea, i=0;
    Random nbRand = new Random();
    Etoile laMauvaiseEtoile;

    CollectionMauvaisesEtoiles = rechercherMauvaisesReponses();

    for (Constellation recupConstellation:LesConstellations)
    {
        numEtoileAlea = nbRand.nextInt(4);
        recupConstellation.getLaListeEtoiles().remove(numEtoileAlea);
        laMauvaiseEtoile = CollectionMauvaisesEtoiles.get(i);
        recupConstellation.setlaMauvaiseRep(laMauvaiseEtoile);
        recupConstellation.getLaListeEtoiles().add(numEtoileAlea, laMauvaiseEtoile);
        i++;
    }
}

/**
 * Méthode de type procedure qui affiche une constellation par fenetre toutes les 60 secondes
 * @throws Exception
 */
public static void afficherApprentissage() throws Exception
{
    for (Constellation uneConstel:LesConstellations)
    {
        FenetreAffichageConstellation laFenetre = new FenetreAffichageConstellation(uneConstel)
        //laFenetre.setResizable(false);
        laFenetre.setVisible(true);
        laFenetre.setLocationRelativeTo(null);
        laFenetre.setResizable(false);
        Thread.sleep(60000);
        laFenetre.dispose();
    }
}

```

```

/**
 * Méthode de type procédure qui affiche une constellation par fenetre avec les boutons toutes l
 * @throws Exception
 */
public static void afficherJeu() throws Exception
{
    int i = 0;
    boolean continuer;
    LeJeu.charger();
    LeJeu.affecterMauvaisesReponses();

    nbPointsJ = 0;
    LesConstellations = LeJeu.getLesConstellations();

    do
    {
        Constellation uneConstellation = LesConstellations.get(i);
        FenetreJeuConstellation laFenetre;
        laFenetre = new FenetreJeuConstellation(uneConstellation);
        laFenetre.setResizable(false);
        laFenetre.setLocationRelativeTo(null);
        laFenetre.setVisible(true);
        i++;
        laFenetre.lancerCompteur();

        continuer = laFenetre.getContinuer();
        laFenetre.dispose();

    }while(i < LesConstellations.size() && continuer);

    if (!continuer)
    {
        JOptionPane.showMessageDialog(null, "Arrête de dormir.");
    }
}

```

```

/**
 * Méthode de type procédure qui ajoute un point de bonne réponse
 */
public static void ajouterPointJ()
{
    nbPointsJ += 1;
}
/**
 * Méthode de type fonction qui renvoie le nombre de points de bonne réponse
 * @return nbPointsJ
 */
public static int getNbPointsJ() {
    return nbPointsJ;
}
/**
 * Méthode de type procédure qui affecte le nombre de points de bonne réponse donné à la don
 * @param nbPointsJ : le nombre de points justes
 */
public static void setNbPointsJ(int nbPointsJ) {
    LeJeu.nbPointsJ = nbPointsJ;
}

```

```

}

```