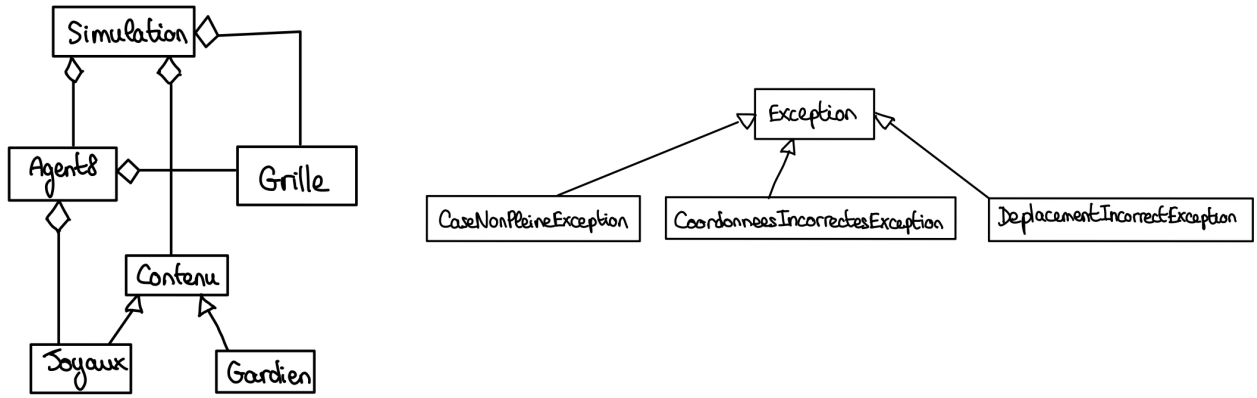
# 1.Projet LU2IN002 - 2023-2024

| |
|---|
| Numéro du groupe de TD/TME : Groupe 8 |

| | |
|---|---|
| *Nom :* WENG | *Nom :* PENG |
| *Prénom :* Julie | *Prénom :* Kairui |
| *N° étudiant :* 21209248 | *N° étudiant :* 21219046 |

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

| *Checklist des éléments utilisés:* | *Nom(s) des classe(s) correspondante(s)* |
|---|---|
| Classe contenant un tableau ou une ArrayList | • Agent8<br>• Simulation |
| Classe avec membres et méthodes statiques | • Agent8 |
| Classe abstraite et méthode abstraite | |
| Interface | |
| Classe avec un constructeur par copie ou clone() | |
| Définition de classe étendant Exception | • CaseNonPleineException<br>• CoordonneesIncorrectesException<br>• DeplacementIncorrectException |
| Gestion des exceptions | • Agent8<br>• Simulation<br>• TestGrille<br>• TestSimulation |

| Utilisation du pattern singleton | |
|---|---|

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

## CLASSE AGENT8

```
public class Agent8{
   private int x;
   private int y;
   private int taille;
   private Joyaux [] sac;
   private static int nbJoyaux = 0;
   private Grille g;

   public Agent8(int x,int y,int taille,Grille g) {
      this.x = x;
      this.y = y;
      this.taille = taille;
      this.g = g;
      sac = new Joyaux[taille];
   }

   public void seDeplacer(int xnew,int ynew) throws DeplacementIncorrectException{
      try {
         if (!g.sontValides(xnew,ynew)) {
            throw new DeplacementIncorrectException("tentative de mauvais placement");
         }
         x = xnew;
         y = ynew;
         if ( !g.caseEstVide(x,y) ) {
            Contenu c = g.getCase(x,y);
            if ( c instanceof Joyaux ) {
               if ( nbJoyaux < taille ) {    //si la sac non vide
                  Joyaux j = (Joyaux) (g.videCase(x,y));
                  sac[nbJoyaux] = j;
                  nbJoyaux++;
               } else {        //sinon
                  System.out.println("la sac est vide");
               }
            } else if ( c instanceof Gardien ) {
               for ( int i=0;i<nbJoyaux;i++ ) {
                  sac[i] = null;
               }
               System.out.println("occupée par un gardien");
               nbJoyaux = 0;
```

```
            }
        }
    } catch (CoordonneesIncorrectesException e) {
        System.out.println("Erreur: "+e.getMessage());
    } catch (CaseNonPleineException e) {
        System.out.println("Erreur: "+e.getMessage());
    }
}

public void seDeplacer(int xnew, int ynew, int f) throws DeplacementIncorrectException{
    try {
        if (!g.sontValides(xnew,ynew)) {
            throw new DeplacementIncorrectException("tentative de mauvais placement");
        }
        x = xnew;
        y = ynew;
        if ( !g.caseEstVide(x,y) ) {
            Contenu c = g.getCase(x,y);
            if ( c instanceof Joyaux ) {
                if ( nbJoyaux < taille ) {    //si la sac non vide
                    Joyaux j = (Joyaux) (g.videCase(x,y));
                    sac[nbJoyaux] = j;
                    nbJoyaux++;
                } else {        //sinon
                    System.out.println("la sac est vide");
                }
            } else if ( c instanceof Gardien ) {
                Gardien gard = (Gardien) (g.getCase(x,y));
                if ( gard.getNbVie() <= f ) {
                    gard = (Gardien) (g.videCase(x,y));
                } else {
                    for ( int i=0;i<nbJoyaux;i++ ) {
                    sac[i] = null;
                    }
                    System.out.println("occupée par un gardien");
                    nbJoyaux = 0;
                    gard.baisserVie(f);
                }
            }
        }
    } catch (CoordonneesIncorrectesException e) {
        System.out.println("Erreur: "+e.getMessage());
    } catch (CaseNonPleineException e) {
        System.out.println("Erreur: "+e.getMessage());
    }
}

public int fortune() {
```

```
        int prix = 0;
        for ( int i=0;i<nbJoyaux;i++ ){
            prix += sac[i].getPrix();
        }
        return prix;
    }

    public void contenuSac() {
        if ( nbJoyaux == 0 ) {
            System.out.println("la sac est vide");
        } else {
            for ( int i=0;i<nbJoyaux;i++ ) {
                System.out.println(sac[i].toString() + " prix : " + sac[i].getPrix());
            }
        }
    }

    public int getX() { return x; }
    public int getY() { return y; }

    public String toString() {
        return "agent : ("+x+" , "+y+")" + " fortune : " + this.fortune();
    }
}
```

---

**CLASSE CASENONPLEINEEXCEPTION**

```
public class CaseNonPleineException extends Exception{
        public CaseNonPleineException(String m) {
                super(m);
        }
}
```

---

**CLASSE COORDONNEESINCORRECTESEXCEPTION**

```
public class CoordonneesIncorrectesException extends Exception{
        public CoordonneesIncorrectesException(String m) {
                super(m);
        }
}
```

---

**CLASSE DEPLACEMENTINCORRECTEXCEPTION**

```
public class DeplacementIncorrectException extends Exception{
        public DeplacementIncorrectException(String m) {
```

```
                super(m);
        }
}
```

---

## CLASSE GARDIEN

```
public class Gardien extends Contenu{
   private int nbVie;

   public Gardien(String nom, int quantite){
      super(nom,quantite);
      nbVie = (int)(Math.random()*201);
   }

   public int getNbVie() { return nbVie; }

   public void baisserVie(int f) {
      nbVie -= f;
   }
}
```

---

## CLASSE JOYAUX

```
public class Joyaux extends Contenu{
   private int prix;

   public Joyaux(String nom, int quantite){
      super(nom,quantite);
      prix = (int)(Math.random()*8000+1);
   }

   public int getPrix() { return prix; }

}
```

---

## CLASSE SIMULATION

```
import java.io.*;

public class Simulation{
        private Agent8 agent;
        private Grille g;
        private Contenu [] tab;
```

```
public Simulation(Contenu[] tab, int taille, int m, Grille g){
	this.tab = new Contenu[tab.length];
	for ( int k=0;k<tab.length;k++ ) {
		this.tab[k] = tab[k];
	}
	this.g = g;
	int cpt = 0;
	try{
		while ( cpt<m ) {
			int i = (int)(Math.random() * g.nbLignes);
			int j = (int)(Math.random() * g.nbColonnes);
			if (g.caseEstVide(i,j) && g.sontValides(i,j)){
				g.setCase(i,j,tab[cpt]);
				cpt++;
			}

		}
	} catch (CoordonneesIncorrectesException e) {
		System.out.println("Erreur: "+e.getMessage());
	}
	int i = (int)(Math.random() * g.nbLignes);
	int j = (int)(Math.random() * g.nbColonnes);
	agent = new Agent8(i,j,taille,g);
}

public String toString() {
	g.affiche(3);
	return g.toString() + agent.toString();
}

public void lance(int nbEtapes, File f) throws IOException{
	int i=1;
	while (i<=nbEtapes){
		int x = agent.getX();
		int y = agent.getY();
		double n = Math.random();    //probabilite de coordonnee
		if ( n<0.25 ){
			x++;
		} else if ( 0.25<=n && n<0.5 ) {
			x--;
		} else if ( 0.5<=n && n<0.75 ){
			y++;
		} else {
			y--;
		}
		while (!(g.sontValides(x,y))) {
			n = Math.random();
			x = agent.getX();
```

```
            y = agent.getY();
            if ( n<0.25 ){
                        x++;
            } else if ( 0.25<=n && n<0.5 ) {
                        x--;
            } else if ( 0.5<=n && n<0.75 ){
                        y++;
            } else {
                        y--;
            }
}

n = Math.random();
String s = "";
if (!f.exists()){
            System.out.println("le fichier n'existe pas");
            return ;
}
else {
            FileOutputStream out = null ;
            if ( n<0.3 ) {                          //probabilite d'obtenir force
                        int force = (int)(Math.random() * 91 + 10);
                        try{
                                    agent.seDeplacer(x,y,force);
                                    s = "Etape "+i+" : "+this.toString() + "\n";
                                    out = new FileOutputStream(f,true);
                                    out.write(s.getBytes());
                                    System.out.println(s);

                        } catch (DeplacementIncorrectException e) {
                                    System.out.println("Erreur: "+e.getMessage());
                        } finally {
                                    if ( out!=null ){
                                                out.close();
                                    }
                        }
            }
            else {
                        try{
                                    agent.seDeplacer(x,y);
                                    s = "Etape "+i+" : "+this.toString() + "\n";
                                    out = new FileOutputStream(f,true);
                                    out.write(s.getBytes());
                                    System.out.println(s);
                        } catch (DeplacementIncorrectException e) {
                                    System.out.println("Erreur: "+e.getMessage());
                        } finally {
                                    if ( out!=null ){
```

```
                                                out.close();
                                        }
                                }
                        }
                }
                i++;
        }
    }
}
```

---

**CLASSE TESTSIMULATION**

```
import java.io.*;
public class TestSimulation{

    public static void main(String [] args) throws IOException{
        Joyaux ja = new Joyaux("A",2);
        Joyaux jb = new Joyaux("B",3);
        Joyaux jc = new Joyaux("C",4);
        Joyaux jd = new Joyaux("D",5);
        Joyaux je = new Joyaux("E",3);
        Joyaux jf = new Joyaux("F",4);
        Joyaux jg = new Joyaux("G",5);
        Gardien d1 = new Gardien ("G1",1);
        Gardien d2 = new Gardien ("G2",3);
        Gardien d3 = new Gardien ("G3",5);

        Contenu [] tab = {d1,d2,d3,ja,jb,jc,jd,je,jf,jg};    // len = 10
        Grille g = new Grille(10,10);
        Simulation s = new Simulation(tab,10,9,g);            // m = 9

        File f = new File("resultat.txt");
        FileOutputStream out = null ;
        String msg = "WENG Julie et PENG Kairui, Groupe 8 \n";
        if (!f.exists()) {
            f.createNewFile();
        }
        f.delete();
        try {
            out = new FileOutputStream(f,true);
            out.write(msg.getBytes());
        } catch (IOException e) {
            System.out.println(e.getMessage());
        } finally {
            if ( out!=null ) { out.close();}
        }
```

```
                s.lance(50,f);


        }
}
```

---