hw16

miércoles, 18 de enero de 2023 4:33

```java
public class Ant {

    private Cell position;

    private int weight;

    private Food food;

    @Override
    public String toString() {
        return "Ant(" + "position=" + position + ", weight=" + weight + ", food=" + food + ')';
    }

    public Ant() {
        this.position = null;
        this.weight = 0;
        this.food = food;
    }

     public Ant(Cell position, int weight, Food food) {
        this.position = position;
        this.weight = weight;
        this.food = food;
    }

    public void run(){
        System.out.println("Running...");
    }

    public void caryingFood(){
        System.out.println("Carrying...");
    }

    public void createPheromone(){
        System.out.println("Droping pheromone...");
    }
```

```java
public class AntEater {

    private Cell position;
    private int antEatenCounter;
    private int runCounter;
    private int state;

    public static void run() {
        System.out.println("Running...");
    }

    @Override
    public String toString() {
        return "AntEater(" + "position=" + position + ", antEatenCounter=" + antEatenCoun
    }


     public AntEater() {
        this.position = null;
        this.antEatenCounter = 0;
        this.runCounter = 0;
        this.state = 0;
    }

    public AntEater(Cell position, int antEatenCounter, int runCounter, int State) {
        this.position = position;
        this.antEatenCounter = antEatenCounter;
        this.runCounter = runCounter;
        this.state = State;
    }
```

```java
public class Area {

    private int width;
    private int height;
    private int initNumberOfFoodPiles;
    private int tickDuration;

   private ArrayList<Cell> cells;
   private ArrayList<FoodPile> foodPile;
    private ArrayList<Colony> colonies;
    private ArrayList<AntEater> andEaters;



    public static void setup(){}

    public static void add(Colony colony){}

    public static void add(AntEater antEater){}

    public static void isAnyFoodRemaining(){}

    public static void getCell(int row, int col){}

    @Override
    public String toString() {
        return "Area(" + "width=" + width + ", height=" + height + ", initNumberOfFoodPi
    }


    public Area() {
        this.width = 0;
        this.height = 0;
        this.initNumberOfFoodPiles = 0;
        this.tickDuration = 0;
        this.cells = null;
        this.foodPile = null;
        this.colonies = null;
        this.andEaters = null;
    }
```

```java
public class Cell{
    private int row;
    private int col;

    public static void receivePheromone(PheromoneDrop pheromne){};

    public static List<Ant> getAnts(){
        ArrayList<Ant> ants =new ArrayList<>();

        return ants;
    }

    @Override
    public String toString() {
        return "Cell(" + "row=" + row + ", col=" + col + ')';
    }


     public Cell() {
        this.row = 0;
        this.col = 0;
    }

    public Cell(int row, int col) {
        this.row = row;
        this.col = col;
    }
```

```java
 */
public class Colony {

    private Nest nest;
    private ArrayList<Ant> ants;

    public void run() {
        System.out.println(x: "Running");
    }

    public void lives() {
        System.out.println(x: "Living");
    }

    @Override
    public String toString() {
        return "Colony(" + "nest=" + nest + ", ants=" + ants + '}';
    }

    public Colony() {
        this.nest = null;
        this.ants = null;
    }

    public Colony(Nest nest, ArrayList<Ant> ants) {
        this.nest = nest;
        this.ants = ants;
    }

    /**
     * @return the nest
     */
    public Nest getNest() {
        return nest;
```

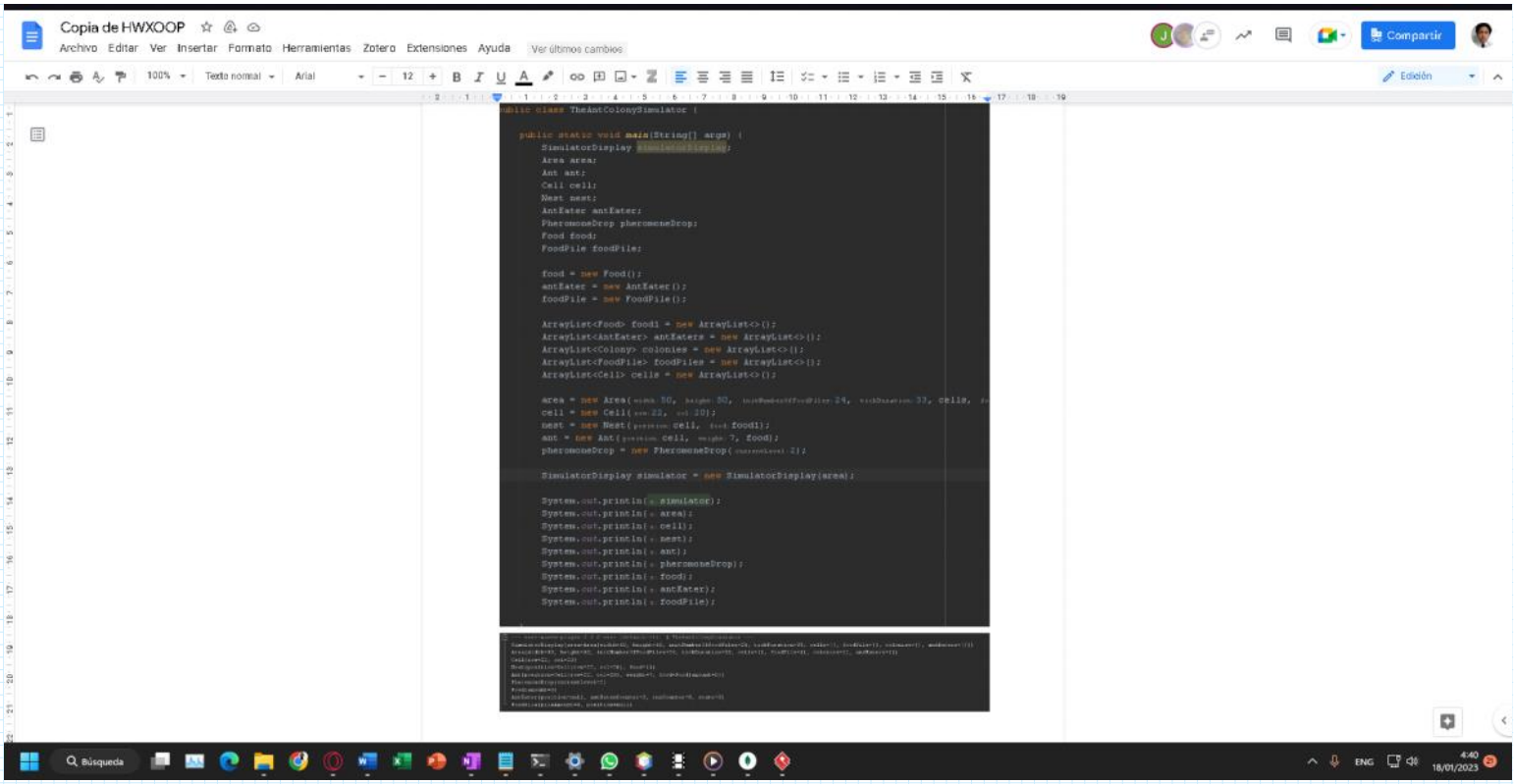
```java
 * @author Leonardo Yaranga,Progress Team, DCCO-ESPE
 */
public class Food {
    private int amount; //not specified on DC

    public Food() {
        this.amount = 0;
    }

    public Food(int amount) {
        this.amount = amount;
    }

    @Override
    public String toString() {
        return "Food(" + "amount=" + getAmount() + ')';
    }
```

```java
 */
public class FoodPile{

    private int pileAmount;  //not specified in DC
    private Cell position;

    public static Food yieldFood(int amount){
        Food food = new Food();
        return food;
    }

    public void createFood(){}

    public FoodPile() {
        this.pileAmount = 0;
        this.position = null;
    }

    public FoodPile(int pileAmount, Cell position) {
        this.pileAmount = pileAmount;
        this.position = position;
    }

    @Override
    public String toString() {
        return "FoodPile(" + "pileAmount=" + getPileAmount() + ", position=" + getPositio
    }

    /**
     * @return the pileAmount
     */
    public int getPileAmount() {
        return pileAmount;
    }
```

```java
*/
public class Nest {

    Cell position;

    ArrayList<Food> food;

    public Nest() {
        this.position = null;
        this.food = null;
    }

    public Nest(Cell position, ArrayList<Food> food) {
        this.position = position;
        this.food = food;
    }

    @Override
    public String toString() {
        return "Nest(" + "position=" + getPosition() + ", food=" + getFood() + ')';
    }

    /**
     * @return the position
     */
    public Cell getPosition() {
        return position;
    }
```

```java
public class PheromoneDrop {

    int currentLevel;

    public static void run() {
    }

    public PheromoneDrop() {
        this.currentLevel = 0;
    }

    public PheromoneDrop(int currentLevel) {
        this.currentLevel = currentLevel;
    }

    @Override
    public String toString() {
        return "PheromoneDrop(" + "currentLevel=" + getCurrentLevel() + ')';
    }

    /**
     * @return the currentLevel
     */
    public int getCurrentLevel() {
        return currentLevel;
    }

    /**
     * @param currentLevel the currentLevel to set
     */
    public void setCurrentLevel(int currentLevel) {
        this.currentLevel = currentLevel;
```

```java
 */
public class SimulatorDisplay {
    private Area area;


    public static void monitors(){
        System.out.println("Monitoring...");
    }

    @Override
    public String toString() {
        return "SimulatorDisplay(" + "area=" + area + ')';
    }


    public SimulatorDisplay() {
        this.area = null;
    }

    public SimulatorDisplay(Area area) {
        this.area = area;
    }

    /**
     * @return the area
     */
    public Area getArea() {
        return area;
```


Copia de HWXOOP
Archivo Editar Ver Insertar Formato Herramientas Zotero Extensiones Ayuda
Compartir
Búsqueda
ENG
4:40
18/01/2023