PROYECTOS EDUCATIVOS

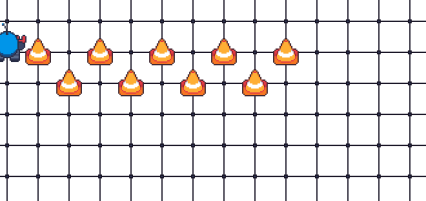
I – IDENTIFICACIÓN

RESULTADOS ESPERADOS: definir e implementar métodos simples, métodos con parámetros, y utilizar sobrecarga de métodos en Java

II – TEMA: Juego del Triangulo

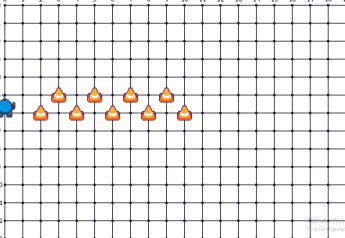
III – DESCRIPCIÓN DEL PROYECTO

**Planteamiento:** Tenemos dos jugadores, el primer jugador tipo RobotJugador1 distribuye conos formando las siguientes figuras (triángulos invertidos).



El jugador (jugador1) distribuye conos formando estos triángulos, y después vuelve a su posición original.

El segundo jugador tipo RobotJugador2 (jugador2) modificar la orientación de los triángulos anteriores, quedando de esta manera:



Definir los métodos necesarios para lograr el objetivo.

Dadas las clases definidas

import uni.robot.\*;

/\*\*

\* <p>Title: Ejercicio RobotJugador1.</p>

\* <p>Description: Utilizado para la cátedra de Introducción a la Programación I.</p> \* <p>Company: Universidad Nacional de Itapúa.</p>

\* @author not attributable

\* @version 1.0

\*/

public class RobotJugador1 extends Robot {

/\*\*

\* Constructor simple

\* @param miMundo Mundo

\*/

public RobotJugador1(Mundo miMundo){

super(miMundo);

}

/\*\*

\* Constructor con parámetros

\* @param unMundo Mundo

\* @param fila int

\* @param columna int

\* @param dir int

\* @param capBol int

\* @param cantConos int

\*/

public RobotJugador1(Mundo unMundo, int fila, int columna, int dir, int capBol, int cantConos) {

super(unMundo, fila, columna, dir, capBol, cantConos);

}

//Definir los métodos necesarios y documentar

}

import uni.robot.\*;

/\*\*

\* <p>Title: Ejercicio RobotJugador2.</p>

\* <p>Description: Utilizado para la cátedra de Introducción a la Programación I.</p> \* <p>Company: Universidad Nacional de Itapúa.</p>

\* @author not attributable

\* @version 1.0

\*/

public class RobotJugador2 extends RobotJugador1 {

/\*\*

\* Constructor simple

\* @param miMundo Mundo

\*/

public RobotJugador2(Mundo miMundo){

super(miMundo);

}

/\*\*

\* @param unMundo Mundo

\* @param fila int

\* @param columna int

\* @param dir int

\* @param capBol int

\* @param cantConos int

\*/

public RobotJugador2(Mundo unMundo, int fila, int columna, int dir,

int capBol, int cantConos) {

super(unMundo, fila, columna, dir, capBol, cantConos);

}

//Definir los métodos necesarios y documentar

}

IV – CAPACIDADES (competencias) A SER DESARROLLADAS

Al trabajar con estos conceptos, desarrollarás varias competencias clave:

1. **Resolución de Problemas**: Saber cómo y cuándo usar métodos simples y sobrecargados te ayuda a resolver problemas de programación de manera más eficiente.

2. **Organización y Modularidad**: La capacidad para definir métodos con parámetros y sobrecargarlos ayuda a organizar tu código de manera modular y a hacer que tu código sea más limpio y mantenible.

3. **Conceptos de Programación Orientada a Objetos (OOP)**: Los métodos y la sobrecarga son aspectos fundamentales de OOP, que incluyen el encapsulamiento y el polimorfismo.

4. **Reusabilidad y Mantenibilidad del Código**: Desarrollar métodos que pueden ser reutilizados y que tienen sobrecarga adecuada contribuye a escribir código más limpio y fácil de mantener.