

```

package Tarefa;

import hemissonComm.HemissonRobot;

public abstract class TarefaRobot extends Thread {

    /**
     * variaveis correspondentes aos index dos sensores para o chão
     */
    public static final int esquerda = 5;
    public static final int direita = 3;

    /**
     * variavel correspondete ao HemissonRobot corrente
     */
    protected HemissonRobot robot;

    /**
     * variavel para o avaliador do ambiente
     */
    protected Avaliador aval;

    /**
     * variavel Random
     */
    protected Random rand;

    /**
     * variavel para guardar os valores lidos dos sensores
     */
    protected int[] valoresSensores;

    /**
     *
     */
    private boolean execute;

    /**
     * Metodo Contrutor
     * @param robot HemissonRobot correspondente ao robot corrente
     * @param aval correspondente ao avaliador do ambiente corrente
     */
    public TarefaRobot(HemissonRobot robot, Avaliador aval) {
        super();
        this.robot = robot;
        this.aval = aval;
        this.rand = new Random();
        this.valoresSensores = null;
        this.execute = true;
    }

    /**
     * Metodo que representa um turno de execução
     * @throws Exception
     */
    public abstract void runOnce() throws Exception;

    /**
     * Metodo Run da classe
     */
    public void run() {
        try {

```

```
        while(this.toExecute()) {
            runOnce();
        }
    } catch (Exception e) {
        System.err.println("Ocorreu uma exceção: " + e.getMessage());
        this.execute = false;
    }
}

/**
 * Metodo que verifica se a tarefa está em execução
 * @return boolean execute?
 */
public boolean toExecute() {
    return execute;
}

/**
 * Metodo que interrompe a execução de uma tarefa
 */
public void interrupt() {
    this.execute = false;
}

/**
 * Metodo que faz o start à tarefa
 */
public final void start() {
    super.start();
}
}
```