```java
package Auxiliar;

import java.io.PrintStream;

public class Direccao {

    /**
     * Constantes de conhecimento publico
     * Estas constantes sao diferentes dos index dos sensores
     */
    public final static int frente = 0;
    public final static int tras = 1;
    public final static int esquerda = 2;
    public final static int direita = 3;
    public final static int fe = 4;
    public final static int fd = 5;
    public final static int te = 6;
    public final static int td = 7;

    /**
     * Constantes para a velocidade de rotação e tempo de rotação
     */
    private final static int velocidadeRotacao = 4;
    private final static int tempo45 = 550; //Em milisegundos

    /**
     * Constante para a direcção corrente
     */
    private int direccao;


    /**
     * Metodo construtor
     * @param direccao direcção corrente
     */
    public Direccao(int direccao) {
        this.direccao = direccao;
    }


    /**
     * Metodo que devolve a direccao  do robot
     * @return devolve um inteiro correspondente à direcção do robot
     */
    public int getDireccao() {
        return direccao;
    }

    /**
     * Metodo que devolve uma direccao oposta à direccao do argumento
     * @param d Direccao do robot
     * @return Direccao oposta à direccao que vinha no argumento
     */
    public static Direccao getDireccaoOposta(Direccao d) {
        return Direccao.getDireccaoOposta(d.getDireccao());
    }

    /**
     * Metodo que devolve uma direccao oposta
     * @param  d int correspondente à constante de direccao de conhecimento publico
     * @return Direccao oposta à direccao que vinha no argumento
     */
    public static Direccao getDireccaoOposta(int d) {
        switch(d) {
        case ValoresSensores.frente:
            return new Direccao(Direccao.tras);
```

```
        case ValoresSensores.tras:
            return new Direccao(Direccao.frente);
        case ValoresSensores.direita:
            return new Direccao(Direccao.esquerda);
        case ValoresSensores.esquerda:
            return new Direccao(Direccao.direita);
        case ValoresSensores.frenteEsquerda:
            return new Direccao(Direccao.td);
        case ValoresSensores.frenteDireita:
            return new Direccao(Direccao.te);
        default:
            return new Direccao(Direccao.tras);
        }
    }


    /**
     * Metodo para executar movimentos tendo em conta a direccao pretendida
     * @param robot HemissonRobot a ser aplicado o movimento
     */
    public void executarMovimento(HemissonRobot robot) throws Exception {
        if(this.direccao == frente) {
            return;
        } else if (this.direccao == tras) {
            robot.setSpeed(velocidadeRotacao, -1 * velocidadeRotacao);
            Thread.sleep(tempo45 * 4);
        } else if(this.direccao == esquerda) {
            robot.setSpeed(-1 * velocidadeRotacao, velocidadeRotacao);
            Thread.sleep(tempo45 * 2);
        } else if(this.direccao == fe) {
            robot.setSpeed(-1 * velocidadeRotacao, velocidadeRotacao);
            Thread.sleep(tempo45);
        } else if(this.direccao == te) {
            robot.setSpeed(-1 * velocidadeRotacao, velocidadeRotacao);
            Thread.sleep(tempo45 * 3);
        } else if(this.direccao == direita) {
            robot.setSpeed(velocidadeRotacao, -1 * velocidadeRotacao);
            Thread.sleep(tempo45 * 2);
        } else if(this.direccao == fd) {
            robot.setSpeed(velocidadeRotacao, -1 * velocidadeRotacao);
            Thread.sleep(tempo45);
        } else if(this.direccao == td) {
            robot.setSpeed(velocidadeRotacao, -1 * velocidadeRotacao);
            Thread.sleep(tempo45 * 3);
        }
        robot.setSpeed(0,0);
    }


    /**
     * Metodo que verifica qual a direccao correspondente
     * @param menor inteiro com o valor do index do sensor com um menor valor de
leitura
     * @return Direccao correspondente ao valor de entrada
     */
    public static Direccao efectuaTraducao(int menor) {
        if(menor == ValoresSensores.frente)
            return new Direccao(Direccao.frente);
        else if(menor == ValoresSensores.tras)
            return new Direccao(Direccao.tras);
        else if(menor == ValoresSensores.esquerda)
            return new Direccao(Direccao.esquerda);
        else if(menor == ValoresSensores.direita)
            return new Direccao(Direccao.direita);
        else if(menor == ValoresSensores.frenteDireita)
            return new Direccao(Direccao.fd);
```

```
            else if(menor == ValoresSensores.frenteEsquerda)
                return new Direccao(Direccao.fe);
            else
                return new Direccao(Direccao.frente);
        }

        /**
         * Metodo que imprime os valores dos sensores
         * @param []valores valores de leitura dos sensores do robot
         */
        public static void printArrayValores(int[] valores) {
            PrintStream err = System.err;
            err.println("Valores obtidos dos sensores:");
            err.print("Frente:          " + valores[ValoresSensores.frente]);
            err.print("; Frente Esquerda: " + valores[ValoresSensores.frenteEsquerda]);
            err.println("; Frente Direita:  " + valores[ValoresSensores.frenteDireita]);
            err.print("Esquerda:        " + valores[ValoresSensores.esquerda]);
            err.print("; Direita:         " + valores[ValoresSensores.direita]);
            err.println("; Tras:            " + valores[ValoresSensores.tras]);
        }

        /**
         * Metodo que dá uma correspondencia textual ao valor recebido
         * @param posicaoValor inteiro para o qual queremos a correspondência
         * @return String designação correspondente ao argumento dado
         */
        public static String getText(int posicaoValor) {
            if(posicaoValor == ValoresSensores.frente)
                return "frente";
            else if(posicaoValor == ValoresSensores.tras)
                return "tras";
            else if(posicaoValor == ValoresSensores.direita)
                return "direita";
            else if(posicaoValor == ValoresSensores.esquerda)
                return "esquerda";
            else if(posicaoValor == ValoresSensores.frenteDireita)
                return "frente direita";
            else if(posicaoValor == ValoresSensores.frenteEsquerda)
                return "frente esquerda";
            else
                return "??";
        }

    }
```