4.- Desenvolver a clase VerboseException que define obxectos Exception coas seguintes características:

- ***A clase VerboseException*** dispón de un atributo ***information*** *accesible dentro do mesmo package* que será capaz de almacenar múltiples mensaxes da clase ***ErrorMessage***.

  A clase *ErrorMessage* constará de un atributo ***String id***, un atributo ***String description*** e un atributo ***details*** **capaz de almacenar múltiples obxectos *ErrorMessage***. Además :
  - **O construtor** accesible dende calquera parte, recibirá *un String id que se almacenará no atributo id*, ***e un String*** que se almacenará no atributo *description.*
  - Terá unha versión sobrecargada do construtor no que non recibe o id, nese caso o obxecto ErrorMessage que se creará terá como id **1I**
  - Terá un método ***addDetail(Verbosity level,String msg)*** que engadirá unha mensaxe con unha id que dependerá do Verbosity. *Os valores posibles para* ***Verbosity*** *deben ser INFO, ERR, e DEBUG.*
    - ***Si o Verbosity é INFO****, se lanzará unha IllegalArgumentException("Este erro e de nivel INFO. So é válido para o obxecto VerboseException");*
    - ***Si o Verbosity é ERR****, se creará un ErrorMessage que se engadirá ao atributo details coa mesma id que o ErrorMessage pero substituíndo a letra I por E.*
    - ***Si o Verbosity é DEBUG****, se creará un ErrorMessage que se engadirá ao atributo details coa mesma id que o ErrorMessage pero substituíndo a letre I por D.*

  No atributo ***information*** *de VerboseException non se poderá almacenar máis de un obxecto ErrorMessage co mesmo id, non poden existir id duplicados. Sen embargo no atributo* ***details*** *de ErrorMessage si poderán existir varios obxectos ErrorMessage co mesmo valor de id.*

- ***A clase VerboseException*** dispón do método ***int addMensaxe(Verbosity level, int id, String msg);***, que se comportará do seguinte xeito:
  - ***Si level é INFO***, se engadirá un novo obxecto ErrorMessage de Verbosity INFO ao atributo ***information*** con id ***id+'I'***
  - ***Si level é ERR***, se engadirá un novo obxecto ErrorMessage de Verbosity ERR ao atributo ***details*** do obxecto ErrorMessage de ***information*** que teña no seu atributo ***id*** *o valor indicado en* ***id+'I'*** *co valor de id id+'E'*. Si non existe ese ErrorMessage se debe lanzar unha IllegalArgumentException coa mensaxe "Non existe erro de nivel previo"
  - ***Si level é DEBUG****, se engadirá un novo obxecto ErrorMessage de Verbosity DEBUG ao atributo* ***details*** *do obxecto ErrorMessage de* ***information*** *que teña no seu atributo* ***id*** *o valor indicado en id+'I' co valor de id id+'D'. Si non existe ese ErrorMessage se debe lanzar unha IllegalArgumentException coa mensaxe "Non existe erro de nivel previo"*

  Retornará o id do novo ErrorMessage xerado.
- *Este método estará sobrecargado: de xeito que non sexa necesario pasarlle o id (se xera por defecto o id seguinte ao mais alto dos existentes) e cun método máis que so recibe unha mensaxe, que se engadirá a Verbosity INFO*

- ***A clase VerboseException*** ao crear o obxect VerboseException engade ás mensaxes de nivel DEBUG ligadas a mensaxe con id 1 as referentes ao punto de erro, obtidas mediante o método ***getStackTrace()*** de Exception e consistirán no número de linea, o nome do método e o nome do ficheiro de cada ***StrackTraceElement*** devolto por getStackTrace. Os construtores existentes accesibles dende calquera parte serán:
  - **VerboseException(String msg):** Creará unha VerboseException con id 1, polo tanto a mensaxe almacenada en Information terá como id ***1I***.
  - **VerboseException(String msg,String description):** Creará unha VerboseException con id 1, e a mensaxe almacenada en Information msg con id 1I, engadindo a mensaxe de nivel ERR description
  - **VerboseException(Exception e):** Transformará a Exception e nunha VerboseException respetando a mensaxe e *establecendo a información de nivel DEBUG ligadas a Exception e*.

- ***A clase VerboseException*** ten un atributo estático *level* que utilizará o ***patrón State*** para definir distintos niveis de información a hora de facilitar a mensaxe de erro cando se produce unha *VerboseException*. Para isto, se almacenará en *level* un obxecto da clase ***InfoLevel, ErrorLevel ou DebugLevel*** segundo corresponda ao nivel de información desexado. Polo tanto:
  - Os obxectos das clases InfoLevel, ErrorLevel e DebugLevel son ***obxectos Level***, que teñen a capacidade de subministrar un Array de String mediante o método ***String[] messages(VerboseException e);***. Este array se creará do seguinte xeito:
    - **Para InfoLevel** constará dos Strings almacenados no atributo ***description*** dos elementos no atributo ***information*** do obxecto VerboseException recibido.

- **Para ErrorLevel** constará de cada String do atributo *information* que terá concatenados os String que lle correspondan ao id etiquetado como ***id+'E'*** precedidos por un tabulador ('\t') e o prefixo "*ERROR:*"
- **Para ErrDebug** constará de cada String do atributo *information* que terá concatenados os String que correspondan ao id etiquetado como ***id+'E'*** e os etiquetados como ***id+'D'*** precedidos por un tabulador ('\t') e o prefixo "*DEBUG: "*

- Os obxectos *Level* deben utilizar o ***Patrón Singleton***, para garantir que non vai existir máis de unha instancia da clase.
- **A clase *VerboseException*** debe ter un método *estático* **void setLevel(Verbosity level)** que estableza o nivel de información para as mensaxes de erro instanciando e colocando o valor apropiado no atributo *level. O Verbosity por defecto será INFO. O método setLevel terá a seguinte forma:*

```
public static void setLevel(Verbosity level) {
    VerboseException.level=LevelFactory.get(level);
}
```

Debes crear a clase ***LevelFactory*** que usa o **Patrón factory** de xeito que o método get retorna o obxecto Level correspondente ao Verbosity indicado no parámetro

- O método ***getMessage()*** de *VerboseException* debe estar sobreposto de xeito que visualice a información que subministre o obxecto State almacenado en *level.*

As saídas do seguinte programa deben ser:

```
public static void test() throws Exception {
    VerboseException error=new VerboseException("Probando o Verbose Exception");
    int idx=error.addMensaxe("O nome está mal");
    error.addMensaxe(ERR,idx,"Debe constar de nome e apelidos");
    error.addMensaxe("A dirección está mal");
    throw error;
}

public static void main(String[] args) throws Exception {
    VerboseException.setLevel(DEBUG);
    try {
        test();
    } catch(VerboseException e) {
        System.out.println(e.getMessage());
    }
}
```

```
// Para setLevel(INFO)
Probando o Verbose Exception
O nome está mal
A dirección está mal

// Para setLevel(ERR)
Probando o Verbose Exception
O nome está mal
        ERROR: Debe constar de nome e apelidos
A dirección está mal

// Para setLevel(DEBUG)
Probando o Verbose Exception
        DEBUG: Line 75 in test in VerboseException.java
        DEBUG: Line 85 in main in VerboseException.java
O nome está mal
        ERROR: Debe constar de nome e apelidos
A dirección está mal
```