

PROGRAMACIÓN JAVA

Nivel: **Avanzado**

Puntos: **7**

Se pide desenvolver no `package csv` a clase **CSVRow** que xestionará unha liña de datos dun ficheiro CSV.

As follas de cálculo poden gardar a información en formato **CSV**. O formato CSV é un formato de Texto no que a primeira liña ou **cabeceira** indica o nome das columnas (campos) mentres que as seguintes liñas ou **liñas de datos** conteñen os valores para cada campo. Cada liña representa un **registro**, e dicir, un conxunto de valores para cada unha das columnas definidas na primeira fila. Vexamos un exemplo:

isbn,	título,	autor,	autor,	xenero,	editorial,	lingua
84-89660-01-8,	Comunicaciones y Redes	William Stallings,	Otro Autor,	Libro de Texto,	Prentice Hall,	Español

Os nomes das columnas non distinguen entre maiúsculas e minúsculas, sendo considerada a columna **Isbn** igual que a columna **isbn**. As columnas se poden repetir, como neste caso que a Publicación ten dúas columnas **autor**.

Se deben almacenar os valores de cada columna nunha estrutura de datos que permita recuperalos rápidamente, tendo en conta que unha columna poderá almacenar varios valores (como no exemplo anterior a columna **autor**) e que non debemos distinguir entre maiúsculas e minúsculas (deberíamos poder recuperar a mesma información indicando a columna “título” ou “Titulo”). Tanto os nomes das columnas como a información deberían ser String sen espazos nin comiñas nin o principio nin o final (ver método **trim** en **String**)

A clase **CSVRow** terá os atributos necesarios para gardar a información da cabeceira e almacenar e recuperar a información dun xeito efectivo, e os seguintes métodos:

- **Construtores**

- **CSVRow(String header)**; : Recibe un String que define a cabeceira do CSV (como por exemplo “**isbn, título, autor, autor, xenero, editorial, lingua**”). Este construtor debe dividir o String utilizando “,” como separador (ver método **split** da clase **String**) e coa información resultante inicializar a estrutura de datos que almacenará a información de cada campo.
- **CSVRow(List<String> header)**; : Recibe un **ArrayList** cos nomes das columnas, que utilizaremos para inicializar a estrutura de datos que almacenará a información de cada campo.
- **CSVRow(String[] header)**; : Recibe un **String[]** cos nomes das columnas, que utilizaremos para inicializar a estrutura de datos que almacenará a información de cada campo.

Nota: Debemos gardar a información orixinal da cabeceira xa que a necesitaremos con posterioridade e ter en conta que non se debe distinguir entre maiúsculas e minúsculas para recuperar a información e que os String non deben ter espazos nin comiñas no inicio nin no final.

- **Métodos**

- **String getHeader()**; : Devolve un String representando a cabeceira do CSV. Cada elemento aparecerá entre comiñas e sen espazos no inicio e final de cada String: “**isbn**”, “**título**”, “**autor**”, “**autor**”, “**xenero**”, “**editorial**”, “**lingua**”.
- **void setLine(String data)**; : Recibe unha liña de datos do CSV. Debe separar os valores utilizando “,” como separador e almacenar a información asociada a columna correspondente. Si liña de datos ten distinto número de elementos que a cabeceira se producirá unha **IllegalArgumentException** coa mensaxe “*Liña de datos errónea*”. Se deben limpar os datos anteriores e colocar os datos novos que se irán asignando ao valor de cabeceira a que correspondan, por exemplo: “**84-89660-01-8,Comunicaciones y Redes William Stallings,Otro Autor,Libro de Texto,Prentice Hall,Español**”, sería “troceada” e almacenada segundo a cabeceira “**isbn, título, autor, autor, xenero, editorial e lingua**”, almacenando asociado á columna **autor** dous valores: “William Stallings” e “Otro Autor”.
NOTA: coidado limpar os datos anteriores no inicio
- **String[] getValues(String field)**; : Retorna un array con todos os valores asociados a un campo. Si o campo non existe lanza unha **IllegalArgumentException** coa mensaxe axeitada. Por exemplo, cos datos do exemplo, **getValues(“autor”)** retornaría o array [“William Stallings”, “Otro Autor”]
- **String getValue(String field)**; : Retorna o primeiro valor asociado a un campo. Si o campo non existe lanza unha **IllegalArgumentException** coa mensaxe axeitada. Si o campo non ten valor asociado retorna **null**. Por exemplo **getValue(“autor”)** retornaría o String “William Stallings”.
- **String[] setValue(String field,String[] value)**;
- **String[] setValue(String field,String value)**; : Substitúe os valores asociados ó campo **field** polo **value** indicado retornando o valor antigo. Si o campo non existe lanza unha **IllegalArgumentException** coa mensaxe axeitada
- **String[] addValue(String field,String value)**; : Engade o valor **value** aos valores asociados a **field**. Retorna o contido anterior. Si o campo non existe lanza unha **IllegalArgumentException** coa mensaxe axeitada
- **toString()**; : Esta clase Debe ter correctamente sobreposto este método de xeito que retorne un **String** de datos compatible co CSV indicado pola cabeceira utilizada. No caso do exemplo retornaría: “**84-89660-01-8**”, “**Comunicaciones y Redes**”, “**William Stallings**”, “**Otro Autor**”, “**Libro de Texto**”, “**Prentice Hall,Español**”. No caso de que a Publicación tivera un so autor, sería algo como “**84-89660-01-8**”, “**Comunicaciones y Redes**”, “**William Stallings**”, “”, “**Libro de Texto**”, “**Prentice Hall,Español**”. **O valor de cada campo debe aparecer entre comiñas**

```

public class CSVRowTest {
    // Contemplamos un máximo de tres autores.
    static String strheader="isbn,titulo,autor,autor,autor,xenero,editorial,lingua,paxinas";
    static String[] header={"isbn","titulo","autor","autor","autor","xenero","editorial","lingua","paxinas"};
    static List<String> lheader=Arrays.asList(header);
    static String testline="978-1-4842-4827-0,Building Microservices Applications on Microsoft Azure,"
        + "Harsh Chawla,Hemant Kathuria,,Libro de Texto,Apress,Inglés,618";

    public static void main(String[] args) {
        System.out.println("Test con constructor String");
        CSVRow row=new CSVRow(strheader);
        showTest(row);
        System.out.println("\nTest con constructor Array");
        row=new CSVRow(header);
        showTest(row);
        System.out.println("\nTest con constructor List");
        row=new CSVRow(lheader);
        showTest(row);
    }

    private static void showTest(CSVRow row) {
        System.out.println("Row Header: "+row.getHeader());
        row.addValue("isbn","00-0000-00-00");
        row.addValue("titulo","Libro de Proba");
        row.addValue("autor","Autor de Proba 1");
        row.addValue("xenero","Test de CSVRow");
        row.addValue("lingua","galego");
        row.addValue("autor","Autor de Proba 2");
        row.addValue("paxinas","0");
        row.addValue("autor","Autor de Proba 3");
        System.out.println("Row Data: "+row);
        System.out.println("Autores: "+Arrays.toString(row.getValues("Autor")));
        System.out.println("Titulo:"+row.getValue("titulo")+" Editorial:"+row.getValue("editorial"));
        System.out.println("--> Cambiando autores");
        row.setValue("autor",new String[] {"autor1","autor2"});
        System.out.println("Row Data: "+row);
        System.out.println("Autores: "+Arrays.toString(row.getValues("autor")));
        System.out.println("Titulo:"+row.getValue("titulo")+" Editorial:"+row.getValue("editorial"));
        System.out.println("--> Poñendo información a partir de String");
        row.setLine(testline);
        System.out.println("Row Data: "+row);
        System.out.println("Autores: "+Arrays.toString(row.getValues("autor")));
        System.out.println("Titulo:"+row.getValue("Titulo")+" Editorial:"+row.getValue("eDitoRial"));

        try {
            System.out.println("Campo Inexistente: "+row.getValue("blabla"));
        } catch (IllegalArgumentException e) {
            System.out.println("ERROR: "+e.getMessage());
        }
    }
}

```

```

Test con constructor String
Row Header: "isbn","titulo","autor","autor","autor","xenero","editorial","lingua","paxinas"
Row Data: "00-0000-00-00","Libro de Proba","Autor de Proba 1","Autor de Proba 2","Autor de Proba 3","Test de CSVRow","","galego","0"
Autores: [Autor de Proba 1, Autor de Proba 2, Autor de Proba 3]
Titulo:Libro de Proba Editorial:null
--> Cambiando autores
Row Data: "00-0000-00-00","Libro de Proba","autor1","autor2","","Test de CSVRow","","galego","0"
Autores: [autor1, autor2]
Titulo:Libro de Proba Editorial:null
--> Poñendo información a partir de String
Row Data: "978-1-4842-4827-0","Building Microservices Applications on Microsoft Azure","Harsh Chawla","Hemant Kathuria","","Libro de Texto","Apress","Inglés","618"
Autores: [Harsh Chawla, Hemant Kathuria]
Titulo:Building Microservices Applications on Microsoft Azure Editorial:Apress
ERROR: O campo blabla non existe

Test con constructor Array
Row Header: "isbn","titulo","autor","autor","autor","xenero","editorial","lingua","paxinas"
Row Data: "00-0000-00-00","Libro de Proba","Autor de Proba 1","Autor de Proba 2","Autor de Proba 3","Test de CSVRow","","galego","0"
Autores: [Autor de Proba 1, Autor de Proba 2, Autor de Proba 3]
Titulo:Libro de Proba Editorial:null
--> Cambiando autores
Row Data: "00-0000-00-00","Libro de Proba","autor1","autor2","","Test de CSVRow","","galego","0"
Autores: [autor1, autor2]
Titulo:Libro de Proba Editorial:null
--> Poñendo información a partir de String
Row Data: "978-1-4842-4827-0","Building Microservices Applications on Microsoft Azure","Harsh Chawla","Hemant Kathuria","","Libro de Texto","Apress","Inglés","618"
Autores: [Harsh Chawla, Hemant Kathuria]
Titulo:Building Microservices Applications on Microsoft Azure Editorial:Apress
ERROR: O campo blabla non existe

Test con constructor List
Row Header: "isbn","titulo","autor","autor","autor","xenero","editorial","lingua","paxinas"
Row Data: "00-0000-00-00","Libro de Proba","Autor de Proba 1","Autor de Proba 2","Autor de Proba 3","Test de CSVRow","","galego","0"
Autores: [Autor de Proba 1, Autor de Proba 2, Autor de Proba 3]
Titulo:Libro de Proba Editorial:null
--> Cambiando autores
Row Data: "00-0000-00-00","Libro de Proba","autor1","autor2","","Test de CSVRow","","galego","0"
Autores: [autor1, autor2]
Titulo:Libro de Proba Editorial:null
--> Poñendo información a partir de String
Row Data: "978-1-4842-4827-0","Building Microservices Applications on Microsoft Azure","Harsh Chawla","Hemant Kathuria","","Libro de Texto","Apress","Inglés","618"
Autores: [Harsh Chawla, Hemant Kathuria]
Titulo:Building Microservices Applications on Microsoft Azure Editorial:Apress
ERROR: O campo blabla non existe

```

Criterios de Avaliación

• Os atributos de CSVRow son axeitados para a tarefa e están correctamente definidos	1
• Os construtores de CSVRow son correctos	1,50
• setLine é correcto	2
• getHeader é correcto	0,50
• getValues e getValue son correctas	1
• setValue e addValue son correctas	1

PD: Si se desenvolveu a clase *Tokenizer* en outro exercicio, se valorará o seu uso.