



Autor: Profesores EDA

ESTRUCTURAS DE DATOS Y ALGORITMOS – TEMA 2. ESTRUCTURAS LINEALES. PROBLEMAS COLAS.

Enunciado

Implementa una clase Java para gestionar una impresora online. La impresora puede recibir peticiones desde diferentes ordeadores. Las peticiones serán impresas por orden de llegada. Cada petición incluye la siguiente información:

id (String) de la máquina que solicita la impresión (por ejemplo, "I3493") y el nombre del fichero a imprimir (por ejemplo, file1.pdf). El programa debe implementar los siguientes métodos:

- addRequest: toma una petición como entrada y la añade al conjunto de peticiones.
- printWork: coge la primera petición y muestra sus datos (id y nombre del fichero) por consola (únicamente simula la impresión de la petición). La petición debe ser eliminada del conjunto de peticiones.
- getNumRequest(): devuelve el número total de peticiones.
- showAll(): muestra todas las peticiones que no han sido impresas.
- printAll(): imprime todas las peticiones. Después de procesar la petición, ésta debe ser eliminada.
- Escribe un método main que incluya las llamadas necesarias para validar todos los métodos descritos anteriormente.

Solución

```
package unit2.queue.printer;

public class SNode {

    public Request elem;
    public SNode next;

    public SNode(Request e) {
        elem = e;
    }
}

public class Request {
```

```

    public String id;
    public String nameFile;

    public Request(String id, String nameFile) {
        this.id = id;
        this.nameFile = nameFile;
    }
}

public class PrinterQueue {

    private SNode first;
    private SNode last;
    int size;

    public boolean isEmpty() {
        return first == null;
    }

    public void addRequest(Request elem) {
        SNode node = new SNode(elem);
        if (isEmpty()) {
            first = node;
        } else {
            last.next = node;
        }
        last = node;
        size++;
    }

    public void printWork() {
        if (isEmpty()) {
            System.out.println("Queue is empty!");
            return ;
        }
        Request firstElem = first.elem;
        first = first.next;
        if (first == null) {
            last = null;
        }
        System.out.println("printing " + firstElem.id+"
"+firstElem.nameFile);
        size--;
    }
}

```

```

public void showAll() {

    for (SNode nodeIt = first; nodeIt != null; nodeIt = nodeIt.next) {
        System.out.println("request: " + nodeIt.elem.id+"
            "+nodeIt.elem.nameFile);
        }
    }

    public void printAll() {
        while (!isEmpty()) {
            printWork();
        }
    }

    public int getNumRequest() {
        return size;
    }
}

```