

PRG - ETSInf. TEORIA. Curs 2016-17. Parcial 2.
5 de juny de 2017. Durada: 2 hores.

Nota: L'examen s'avalua sobre 10 punts, però el seu pes específic en la nota final de PRG és de **3 punts**.

1. 2.5 punts Es disposa d'un fitxer de text, tal que cadascuna de les seves línies ha de contenir un únic valor numèric **double** (tenint com a separador decimal, cas d'existir, el punt decimal).

No obstant això, no totes les línies tenen un valor escrit correctament, per la qual cosa es vol rebutjar aquestes. Per a això, s'ha de generar un nou fitxer que contindrà, exclusivament, les línies correctes del fitxer original. Cada línia ha de contenir un únic valor numèric **double**.

El nou fitxer s'escriurà en el mateix directori que l'original i el seu nom s'obtindrà afegint al de l'original el sufix "_nou".

Per exemple: si el nom del fitxer original era ".././valors.txt" llavors el nom del fitxer nou haurà de ser: ".././valors.txt_nou".

En el cas que el fitxer original no existeixca, o no siga accessible, no s'escriurà cap missatge. Així mateix, si alguna línia conté algun valor no correcte, s'hauran de tractar la resta de línies posteriors que puguin existir, tot i que no s'ha d'escriure cap missatge relacionat amb la línia errònia.

Es demana: Escriure un mètode que reba el nom del fitxer original com una **String** i realitzi el processament indicat.

NOTA: S'ha de tenir en compte, per a la gestió dels fitxers de sortida o d'entrada, la possible excepció: `FileNotFoundException`. Addicionalment, per a la lectura del valor numèric, s'ha de tenir en compte, bé l'excepció `InputMismatchException`, bé la `NumberFormatException`, en el cas en què s'haja utilitzat el mètode `Double.parseDouble(String)`.

Solució:

```
// Solució alternativa 1.
public static void fitxNou(String ftx) {
    File f = new File(ftx);
    Scanner s = null; PrintWriter p = null;
    try {
        s = new Scanner(f).useLocale(Locale.US);
        p = new PrintWriter(ftx + "_nou");
        while (s.hasNextLine()) {
            try {
                String lA = s.nextLine().trim();
                p.println(Double.parseDouble(lA));
            } catch (NumberFormatException n) { }
        }
    } catch (FileNotFoundException e) { }
    finally {
        if (s != null) { s.close(); }
        if (p != null) { p.close(); }
    }
}
```

```
// Solució alternativa 2.
public static void fitxNou(String ftx) {
    File f = new File(ftx);
    Scanner s = null; PrintWriter p = null;
    try {
        s = new Scanner(f).useLocale(Locale.US);
        p = new PrintWriter(ftx + "_nou");
        while (s.hasNextLine()) {
```

```

        try {
            p.println(s.nextDouble());
        } catch (InputMismatchException e) { }
        finally { s.nextLine(); }
    }
} catch (FileNotFoundException e) { }
finally {
    if (s != null) { s.close(); }
    if (p != null) { p.close(); }
}
}

```

2. **2.5 punts** **Es demana:** implementar un nou constructor en la classe `PilaIntEnla`, tal que donada una `PilaIntEnla p` torne una altra pila que siga una còpia independent de `p`. Així, si s'executa la instrucció:
- ```
PilaIntEnla res = new PilaIntEnla(PilaIntEnla p);
```
- es podrà emprar i desempilar de la pila `res` o de la pila `p` sense influir una a l'altra.
- NOTA:** Només es permet accedir als atributs de la classe, quedant prohibit l'accés als seus mètodes.

#### Solució:

```

public PilaIntEnla(PilaIntEnla p) {
 if (p.cim != null) {
 cim = new NodeInt(p.cim.dada);
 NodeInt ultim = cim;
 NodeInt aux = p.cim.seguint;
 while (aux != null) {
 ultim.seguint = new NodeInt(aux.dada);
 ultim = ultim.seguint;
 aux = aux.seguint;
 }
 talla = p.talla;
 }
}

```

3. **2.5 punts** Es vol implementar el mètode `equals(Object)` a la classe `LlistaPIIntEnla`, per definir quan la `LlistaPIIntEnla` sobre la qual s'aplique aquest mètode és o no igual a una altra que es rebrà com a argument. Donades dues `LlistaPIIntEnla` qualsevol, es consideren iguals quan:

1. Tenen els mateixos elements en les mateixes posicions i
2. el seu *Punt d'Interès* és el mateix, és a dir, es troba en la mateixa posició en les dues llistes.

**Es demana:** completar el codi següent perquè el mètode `equals(Object)` s'execute segons la definició anterior:

```

/** Retorna si l'objecte en curs és igual a o */
public boolean equals(Object o) {
 boolean res = true;
 if (!(o instanceof LlistaPIIntEnla)) { res = false; }
 else {
 LlistaPIIntEnla altra = (LlistaPIIntEnla) o;

 // COMPLETA el codi per tornar si "this" i "altra" són iguals
 }
}

```

**NOTA:** La posició del *Punt d'Interès* haurà de romandre inalterada després de l'execució del mètode.

**Solució:**

```
/** Retorna si l'objecte en curs és igual a o */
public boolean equals(Object o) {
 boolean res = true;
 if (!(o instanceof LlistaPIIntEnla)) { res = false; }
 else {
 LlistaPIIntEnla altra = (LlistaPIIntEnla) o;
 if (this.talla != altra.talla) { res = false; }
 else {
 NodeInt p = this.primer, q = altra.primer;
 while (p != null && res) {
 res = (p.dada == q.dada);
 if (res && (p == this.antPI)) { res = q == altra.antPI; }
 if (res) { p = p.seguint; q = q.seguint; }
 }
 }
 }
 return res;
}
```

4. 2.5 punts Per problemes de codificació, s'ha rebut cert text en què totes les lletres 'ç' han estat canviades pel parell de caràcters '\c' i totes les 'Ç' pel parell '\C'.

Es disposa d'una classe ja creada, `LlistaPICharEnla`, que implementa una llista amb punt d'interès, els elements de la qual són caràcters, i amb els següents mètodes públics:

|                                          |                                     |
|------------------------------------------|-------------------------------------|
| <code>public void inici()</code>         | <code>public void seguint()</code>  |
| <code>public void inserir(char e)</code> | <code>public char eliminar()</code> |
| <code>public char recuperar()</code>     | <code>public int talla()</code>     |
| <code>public boolean esBuida()</code>    | <code>public boolean esFi()</code>  |

**Es demana:** implementar un mètode estàtic, extern a la classe `LlistaPICharEnla`, que donat un objecte d'aquest tipus que conté, caràcter a caràcter, cert text amb el problema de codificació anteriorment exposat, modifiqui el contingut d'aquesta llista per codificar-la de forma correcta.

**Per exemple:** donada una llista amb el text "En PU\COL, una on\ca de xocolata es un bon dol\c.", s'haurà de modificar perquè quede "En PUÇOL, una onça de xocolata es un bon dolç".

**NOTA:** Considereu que si apareix el caràcter '\', aquest sempre anirà seguit d'una 'c' o una 'C'.

**Solució:**

```
public static void modifTrencada(LlistaPICharEnla l) {
 l.inici();
 while (!l.esFi()) {
 char c1 = l.recuperar();
 if (c1 == '\\') {
 l.eliminar();
 char c2 = l.recuperar();
 if (c2 == 'c') { l.inserir('ç'); }
 else { l.inserir('Ç'); }
 } else { l.seguint(); }
 }
}
```