

PROGRAMACIÓ ORIENTADA A OBJECTES

TRIMESTRE 2 – CURS 2021/2022

PRÀCTICA 3. SESSIÓ 2

Objectiu: Interfície Comparable. Ordenació de taules: mètode de la Bombolla.

Tractament d'errors: Excepcions. Crear les nostres classes d'excepcions, subclasses de la classe Exception. Usar la jerarquia de classes d'excepcions proporcionada per l'API de java:

NullPointerException
IllegalArgumentException
ClassCastException
IndexOutOfBoundsException
NegativeArraySizeException
ArithmeticException.

Durada: Una sessió

Lliurament: Llistat imprès dels fonts i penjar el projecte al Moodle

Data Lliurament: Abans de la Pràctica 4

Important: Imprescindible per aprovar la pràctica seguir les pautes de l'enunciat.

Afegim funcionalitat a les classes definides en la sessió anterior.

La classe MedicamentPíndoles ha d'implementar la interfície Comparable:

S'han de poder comparar objectes MedicamentPíndoles tenint en compte el número de píndoles que queden al medicament. Cal indicar si a un medicament li queden més, menys o igual nombre de píndoles que a un altre. **Un medicament és més petit que altre si li queden menys píndoles, són iguals si li queden les mateixes i més gran en qualsevol altre cas.** Adoneu-vos que la igualtat implementada amb el mètode equals no coincideix amb aquest criteri (no és recomanable però en aquesta pràctica és així).

La classe Malalt ha d'implementar la interfície Comparable:

Un malalt és menor que un altre si li queden menys píndoles, més gran si n'hi queden més, i iguals si li queden el mateix nombre de píndoles al magatzem. Per saber quantes píndoles li queden a un objecte Malalt cal sumar el número de píndoles que hi ha en cadascun dels medicaments que té al magatzem, hi ha un mètode a la classe Malalt que fa aquest càlcul.

La classe Malalt ha de tenir més mètodes:

- Afegiu a la classe Malalt un mètode **MedicamentPindoles medicamentMenys()** per trobar el medicament del malalt que té menys píndoles. Si dos o més medicaments tenen el mateix número de píndoles cal retornar el medicament que té nom de medicament menor (alfabèticament), a igualtat de noms és irrellevant el retornat. Per fer el tractament **obligatòriament féu crides al mètode compareTo de la classe MedicamentPindoles**. Adoneu-vos de que **no podeu usar** el mètode equals per comparar la igualtat de medicaments ja que el criteri d'igualtat que té redefint la classe MedicamentPindoles **no lliga** amb el de compareTo.
- Afegiu a la classe Malalt un mètode, en direm **void llistatOrdenatAscendent()** per a que faci una visualització a pantalla de tots els medicaments de l'objecte. Aquest llistat ha d'estar ordenat en funció del número de píndoles que li queden a cada medicament. L'ordenació ha de ser ascendent, de menys a més número de píndoles que li queden a cada medicament. El mètode d'ordenació que heu d'implementar és la **Bombolla** i aquest **ha d'invocar al mètode compareTo de la classe MedicamentPindoles**. En el llistat, de cada medicament, ha de mostrar: nom del medicament i número de píndoles que li queden del total. **Imprescindible** que el mètode print/ln necessari per confeccionar el llistat tingui com a paràmetre un objecte MedicamentPindoles. Per realitzar l'ordenació és imprescindible programar un mètode privat amb el següent prototipus:

```
private void Bombolla(Comparable[])
```
- Afegiu a la classe Malalt altre mètode, en direm **void llistatOrdenatDescendent()** per a que faci una visualització a pantalla de tots els medicaments de l'objecte. En aquest cas el llistat ha de ser descendent, ordenat en funció del número de píndoles que li queden a cada medicament però de més a menys. Per fer l'ordenació invoqueu al **mètode sort de la llibreria de Java**. En el llistat, de cada medicament, ha de mostrar: nom del medicament i número de píndoles que li queden del total. **Imprescindible** que el mètode print/ln necessari per confeccionar el llistat tingui com a paràmetre un objecte MedicamentPindoles.

Aquests mètodes són consultors, han d'extreure les dades sol·licitades pel mètode però **NO** han de modificar els atributs dels objectes

java.util

Class Arrays

sort

```
public static void sort(Object[] a)
```

Sorts the specified array of objects into ascending order, according to the natural ordering of its elements. All elements in the array must implement the Comparable interface. Furthermore, all elements in the array must be mutually comparable (that is, `e1.compareTo(e2)` must not throw a `ClassCastException` for any elements `e1` and `e2` in the array).

This sort is guaranteed to be *stable*: equal elements will not be reordered as a result of the sort.

Parameters:

a - the array to be sorted

Throws:

`ClassCastException` - if the array contains elements that are not *mutually comparable* (for example, strings and integers)

`IllegalArgumentException` - (optional) if the natural ordering of the array elements is found to violate the `Comparable` contract

Compte!!!!
Si la taula a ordenar no està completament plena el mètode dona error

Escriure la classe ExceptionMedicament

Creeu una subclasse de la classe Exception.

Poseu-hi un mètode constructor amb un únic paràmetre corresponent al missatge explicatiu de la situació d'error que s'ha donat.

Recordeu que les excepcions d'aquesta nova classe seran de captura/resposta obligatòria.

Afegir el tractament d'errors

Cal tractar totes les situacions anormals que es puguin donar en les implementacions de tots els mètodes demanats en les classes d'ambdues sessions de la pràctica 3.

Per cada classe, per cadascun dels seus mètodes cal determinar si pot o no generar una situació anormal, en cas afirmatiu cal preparar el mètode per llançar un objecte Exception. Algunes situacions, entre d'altres que haureu de determinar vosaltres, són:

- El mètode **public void prendrePindola()** de la classe MedicamentPindoles implementat en la sessió anterior no fa res si no queda cap píndola per prendre, ara caldrà llançar una excepció si es dona aquesta situació. El mètode sol·licita que es prengui una píndola, si el medicament no en té cap no pot fer-ho, no pot fer el que se li demana que faci, és una situació anormal que controlareu mitjançant l'ús d'excepcions. L'excepció en aquesta situació, ha de ser un objecte de la classe ExceptionMedicament. Totes les situacions d'error referents a tractaments envers objectes de la classe Malalt i MedicamentPindoles llançaran objectes ExceptionMedicament.
- Els constructors i mètodes que reben paràmetres que no són el que s'espera: han de **llançar** una excepció del tipus **IllegalArgumentException**. **No farem confiança amb l'usuari.**
- En la implementació dels mètode **compareTo** si el paràmetre no és del tipus que s'espera ha de **llançar** una excepció del tipus **ClassCastException**.
- **..... resta de situacions anormals que heu de determinar vosaltres.....**

Si considereu convenient crear més classes d'excepcions per controlar situacions anormals podeu fer-ho!

La classe Prova

Escriviu **dos** mètodes **main**:

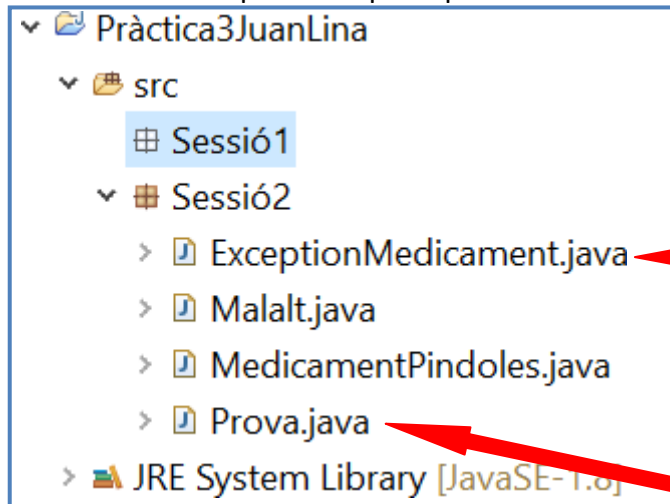
- Un mètode **main** per il·lustrar la gestió que heu fet de les situacions d'errors. **Provoqueu** totes les situacions incorrectes per recollir les excepcions i visualitzar a pantalla les diferents situacions anormals que es generin.
- Altre **main** on es generin aleatòriament objectes Malalt. **Per cada malalt** mostreu el correcte funcionament dels mètodes que visualitzen a pantalla els llistats ordenats, invocant als mètodes corresponents de la classe Malalt. Tots els valors numèrics que es requereixin genereu-los aleatòriament usant **la classe Random** (prohibit usar la classe Math), i pels String requerits, noms de medicament i noms de malalts agafeu-los aleatòriament **d'un/s atribut/s final estàtic** que ubicareu a la mateixa classe. Caldrà un vector amb noms de medicaments i altre amb noms de persones malaltes.

L'usuari **només** indicarà el número de objectes Malalt que es volen crear. Un cop mostrats els llistats de cada malalt, ordeneu els objectes Malalt creats invocant el mètode **sort del paquet java.util** i mostreu-ne el resultat a pantalla.

Es valorarà l'anàlisi descendent aplicat.

Organització

Dins del mateix projecte creeu un nou paquet on copiareu les classes de la sessió anterior i sobre aquestes còpies aplicareu el demanat en aquesta sessió.



Classe a afegir, únicament tindrà constructor amb un paràmetre corresponent al missatge explicatiu de la situació anormal que s'ha generat.

Dos mètodes main per il·lustrar el programat en aquesta sessió. Està explicat a l'enunciat.

Què se us subministra?

L'enunciat amb les especificacions i línies a seguir per el desenvolupament del disseny de classes.

Transparències: Mètode de la Bombolla.

Què s'ha de lliurar i com?

S'ha de lliurar la carpeta que conté el projecte Eclipse amb el vostre desenvolupament de la pràctica. La carpeta s'ha de lliurar amb tot el seu contingut i comprimida amb ZIP o RAR.

També s'ha de lliurar un **llistat en paper** del codi desenvolupat (no la classe Prova). El format de lliurament d'aquest codi ha de seguir el patró indicat en la presentació de l'assignatura: amb portada, índex, número de pàgina, tabulació ... En **aquest llistat** cal que indiqueu:

- la distribució de la feina entre els dos estudiants. És a dir, el grau de participació de cada membre del grup en la realització d'aquesta activitat.
- Si el programa no funciona cal que indiqueu quina/es parts no funcionen explicant que és el que passa.

On s'ha de lliurar?

El lliurament del projecte es farà a través de la plataforma Moodle i no s'acceptarà cap altra via. Feu atenció a la data i hora límit.

El lliurament en paper es farà directament a la professora a **l'inici de la primera sessió de la pràctica 4.**

Quan s'ha de lliurar?

El lliurament es podrà fer fins el **dia indicat a sota**. Tingueu present que a partir d'aquesta hora el sistema bloquejarà, de manera automàtica, la possibilitat de lliurament.

Lliurament Moodle → 10 Març a les 23:50h

Lliurament en paper → 11 de Març inici sessió de pràctiques

Pauta de correcció:

En aquesta pràctica de dues sessions resoldreu un únic enunciat que s'anirà ampliant sessió a sessió.

Aquesta sessió 2 de la pràctica 3 té un pes del 30% de la qualificació total de la pràctica 3.

Valoració (sobre 10 punts):

- 1.- Mètodes nous de la classe MedicamentPindoles i classe **Malalt (6 punts)**
- 2.- Control excepcions **(2.5 punts)**
- 2.- Escriptura de la classe **Prova (1.5 punts)**

La correcció valorarà:

- 1.- **Descomposició funcional** aplicada, tant per un reaprofitament del codi com per la claredat de la solució mostrada. És imprescindible que cada tasca amb una funcionalitat pròpia tingui el seu propi procediment.
- 2.- L'aplicació dels **esquemes** adients (recorregut i cerca). El **magatzem està ordenat, per tant tingueu-ho en compte a l'aplicar l'esquema de cerca.**
- 3.- La **robustesa**, és a dir el control de què les dades demanades a l'usuari siguin correctes.
- 4.- **Eficiència** en quan a no fer càlculs repetits amb unes mateixes dades.

La qualificació dependrà dels ítems indicats prèviament. **Obligatòriament heu de seguir les pautes de programació** donades en l'escriptura del programa. La vostra solució s'ha d'ajustar a l'enunciat de l'exercici.