**Programació Orientada a Objecte**

**Pràctica 3**

Miquel Rodríguez Juvany

Eloi Egea

Grup 101

2021-22

**Índex**

Participació.......................................................................................................................2

Practica3Sessio1 MedicamentPindoles.java.....................................................................3

Practica3Sessio1 Malalt.java.............................................................................................4

Practica3Sessio1 Prova.java..............................................................................................7

Practica3Sessio2 MedicamentPindoles.java.....................................................................9

Practica3Sessio2 Malalt.java...........................................................................................10

Practica3Sessio2 ExceptionMedicament.java.................................................................14

Practica3Sessio2 Prova.java............................................................................................15

**Participació**

Amb aquesta pràctica hem tingut bastants problemes.

A la Sessió 1 vam fer amb facilitat la classe MedicamentPindoles. Amb la classe Malalt el primer problema va ser entendre el fet que el magatzem havia d’estar ordenat perquè no sabíem com fer-ho i ens vam equivocar moltes vegades. Els mètodes que més vam arrastrar per problemes van ser comprarMedicamentPindoles, eliminarMedicamentPindoles, numMedicamentsPerQueden i els privats troba i crear. També vam trobar problemes a l’hra de redefinir l’equals perquè no sabíem implementar els mètodes split i sort. Quan creiem que ja havíem resolt els problemes trobats, a l’hora de fer el main a la classe Prova vam veure que tots els mètodes mencionats anteriorment estaven malament i vam trigar molt a poder corregir-los.

A la Sessió 2 no hem tingut problema a l’hora d’implementar les interfícies Comparable i el corresponent mètode compareTo però sí que hem trobat molts problemes amb medicamentMenys, llistatOrdenatAscendent, Bombolla i llistatOrdenatDescendent. Per a aplicar les excepcions hem tingut menys problemes, hem determinat fins a 10 excepcions: 7 IllegalArgumentException, 2 ClassCastException i 1 ExceptionMedicament.

Hem trobat complicat observar el main amb errors perquè cada error impedia al programa avançar.

**Sessió 1**

**MedicamentPindoles.java**

**package** Practica3Sessio1;

**public** **class** MedicamentPindoles {

**private** **final** String medicament;

**private** **final** **int** unitats;

**private** **int** preses;

**public** MedicamentPindoles (String nom, **int** unitats) {

**this**.medicament = nom;

**this**.unitats = unitats;

**this**.preses = 0;

}

**public** **int** quantesUnitatsQueden() {

**return** (**this**.unitats-**this**.preses);

}

**public** **void** prendrePindola() {

**if** (**this**.preses < **this**.unitats) {

**this**.preses++;

}

}

**public** **int** getPindolesPreses() {

**return** **this**.preses;

}

**public** String getNom() {

**return** **this**.medicament;

}

**public** **boolean** equals(Object med) {

**if** (**this** == med) {

**return** **true**;

} **else** **return** **false**;

}

**public** String toString() {

**return** ("Nom del medicament " + **this**.medicament + " amb " + **this**.quantesUnitatsQueden() + " píndoles de " + **this**.unitats);

}

}

**Malalt.java**

**package** Practica3Sessio1;

//import java.util;

**import** java.util.Arrays;

**public** **class** Malalt {

**private** String nom; // Atribut 1

**private** MedicamentPindoles[] magatzem; // Atribut 2

**private** **int** medicaments; // Atribut 3

**private** **final** **int** inc; // Atribut 4

**public** Malalt (String nom, **int** max, **int** inc) {

**this**.nom = nom;

**this**.magatzem = **new** MedicamentPindoles[max];

**this**.medicaments = 0;

**this**.inc = inc;

}

**public** String getNom() {**return** **this**.nom;}

**public** **int** getMedicaments() {**return** **this**.medicaments;}

**public** **int** getIncrement() {**return** **this**.inc;}

**public** MedicamentPindoles getMedicamentPindoles(**int** quin) {

**if** (quin >= 0 && quin < **this**.medicaments) {

**return** **this**.magatzem[quin];

} **else** **return** **null**;

}

**public** MedicamentPindoles getMedicamentPindolesNoBuit() {

MedicamentPindoles ret = **null**;

**for** (**int** i=0; i<**this**.medicaments; i++) {

**if** (**this**.magatzem[i] != **null**) {

ret = **this**.magatzem[i];

} **else** ret = **null**;

}

**return** ret;

}

**public** **void** comprarMedicamentPindoles(MedicamentPindoles p) {

**int** pos = 0;

**boolean** trobat = **false**;

**if** (**this**.medicaments >= **this**.magatzem.length) {

MedicamentPindoles[] nouMagatzem = **new** MedicamentPindoles[**this**.medicaments + **this**.inc];

**for** (**int** j = 0; j < **this**.medicaments; j++) {

nouMagatzem[j] = **this**.magatzem[j];

}

**this**.magatzem = nouMagatzem;

}

// buscar posicio a la que va

**for** (**int** i = 0; i < **this**.medicaments && !trobat; i++) {

**if** (magatzem[i].getNom().compareTo(p.getNom()) > 0) {

// hem trobat el lloc

pos = i;

trobat = **true**;

} **else** **if** (magatzem[i].getNom().compareTo(p.getNom()) == 0) {

**if** (magatzem[i].quantesUnitatsQueden() < p.quantesUnitatsQueden()) {

pos = i;

trobat = **true**;

}

}

}

**if** (!trobat)

pos = **this**.medicaments;

// fer desplaçament

**for** (**int** i = **this**.medicaments; i >= pos; i--) {

**this**.magatzem[i + 1] = **this**.magatzem[i];

}

**this**.magatzem[pos] = p;

**this**.medicaments++;

}

**public** **void** comprarMedicamentPindoles(String p, **int** pindoles) {

MedicamentPindoles obj = **new** MedicamentPindoles(p, pindoles);

**this**.comprarMedicamentPindoles(obj);

}

**public** **int** totalPindolesQueden() {

**int** queden = 0;

// Seria fins el limit o fins l'ultim omplert

**for** (**int** i=0; i<**this**.medicaments; i++) {

queden += **this**.magatzem[i].quantesUnitatsQueden();

}

**return** queden;

}

**public** **int** totalPindolesQueden(String nom) {

**int** queden = 0;

**for** (**int** i=0; i<**this**.medicaments; i++) {

**if** (**this**.magatzem[i].getNom().equals(nom)) {

queden += **this**.magatzem[i].quantesUnitatsQueden();

}

}

**return** queden;

}

**public** **int** maximPindoles() {

String nom;

**int** queden, quedenAnt = 0;

**for** (**int** i=0; i<**this**.medicaments; i++) {

nom = **this**.magatzem[i].getNom();

queden = **this**.totalPindolesQueden(nom);

**if** (queden >= quedenAnt) {

quedenAnt = queden;

}

}

**return** quedenAnt;

}

**public** String numMedicamentsPerQueden() {

String retorn = "";

**int**[] quants, queden;

**int** num = 0;

queden = **new** **int**[**this**.medicaments];

quants = **new** **int**[**this**.medicaments];

**for** (**int** i = 0; i < **this**.medicaments; i++) {

//if (this.magatzem[i].quantesUnitatsQueden() > num) {

**int** x = *troba*(**this**.magatzem[i], queden, num);

**if**(x != -1) {

quants[x]++;

} **else** {

queden[num]=**this**.magatzem[i].quantesUnitatsQueden();

quants[num]=1;

num++;

}

}

retorn = *crear*(queden, quants, num);

**return** retorn;

}

**private** **static** **int** troba(MedicamentPindoles caixa, **int**[] queden, **int** quants) {

**for** (**int** i = 0; i < quants; i++) {

**if** (caixa.quantesUnitatsQueden() == queden[i]) {

**return** i;

}

}

**return** -1;

}

**private** **static** String crear(**int**[] queden, **int**[] quants, **int** quantes) {

String retorn="";

**for**(**int** i=0; i<quantes;i++) {

retorn = "Amb " + queden[i] + " pindoles queden " + quants[i] + " medicaments - ";

}

**return** retorn;

}

// A la hora d'eliminar un altre element fem el tractament tot i que tingui 0 píndoles

**public** **int** eliminarMedicamentPindoles() {

**int** retorn = 0;

**for** (**int** i=0; i<**this**.medicaments; i++) {

**if** (**this**.magatzem[i].quantesUnitatsQueden()==0) {

**for** (**int** j=i; j<**this**.medicaments; j++) {

**this**.magatzem[j] = **this**.magatzem[j++];

}

**this**.magatzem[**this**.medicaments] = **null**;

**this**.medicaments--;

retorn++;

}

}

**return** retorn;

}

**public** MedicamentPindoles[] donaMedicamentsBuits() {

**int** j=0;

MedicamentPindoles[] retorn;

retorn = **new** MedicamentPindoles[**this**.medicaments];

**for** (**int** i=0; i<**this**.medicaments; i++) {

**if** (**this**.magatzem[i].quantesUnitatsQueden()==0) {

retorn[j] = **this**.magatzem[i];

j++;

}

}

**return** retorn;

}

**public** String toString() {

String retorn = "";

**for** (**int** i = 0; i < **this**.medicaments; i++) {

retorn += (i+1) + ". "+**this**.magatzem[i].toString()+"\n";

}

**return** retorn;

}

**public** **boolean** equals(Object o) {

String[] arrayA, arrayB;

**if** (((Malalt) o) **instanceof** Malalt) {

String a = **this**.numMedicamentsPerQueden();

String b = ((Malalt) o).numMedicamentsPerQueden();

arrayA = a.split("-", **this**.medicaments);

arrayB = b.split("-", ((Malalt) o).getMedicaments());

Arrays.*sort*(arrayA);

Arrays.*sort*(arrayB);

a = Arrays.*toString*(arrayA);

b = Arrays.*toString*(arrayB);

**if** (a == b) {

**return** **true**;

} **else**

**return** **false**;

} **else**

**return** **false**;

}

}

**Prova.java**

**package** Practica3Sessio1;

**public** **class** Prova {

**private** **static** **void** prendreTotsElsMedicamentsComprats(Malalt m) {

**for** (**int** i=0; i<m.getMedicaments(); i++) {

MedicamentPindoles a = m.getMedicamentPindolesNoBuit();

*prendreDelMedicament*(a);

}

}

**private** **static** **void** prendreDelMedicament(MedicamentPindoles m) {

**for** (**int** i = 0; i < m.quantesUnitatsQueden(); i++) {

m.prendrePindola();

}

}

**public** **static** **void** main(String [] args) {

Malalt maria = **new** Malalt("Maria Fernandez", 10, 10);

MedicamentPindoles enelapril60 = **new** MedicamentPindoles("ENELAPRIL", 60);

MedicamentPindoles ibuprofeno20 = **new** MedicamentPindoles("IBUPROFENO", 20);

maria.comprarMedicamentPindoles(enelapril60);

maria.comprarMedicamentPindoles(enelapril60);

maria.comprarMedicamentPindoles(ibuprofeno20);

Malalt joan = **new** Malalt("Joan Moll", 15, 10);

MedicamentPindoles auxina30 = **new** MedicamentPindoles("AUXINA", 30);

MedicamentPindoles auxina35 = **new** MedicamentPindoles("AUXINA", 35);

joan.comprarMedicamentPindoles(auxina30);

joan.comprarMedicamentPindoles(auxina30);

joan.comprarMedicamentPindoles(auxina35);

joan.comprarMedicamentPindoles(auxina35);

joan.comprarMedicamentPindoles(ibuprofeno20);

joan.comprarMedicamentPindoles(ibuprofeno20);

System.***out***.println("El malalt " + maria.getNom() + " té:");

System.***out***.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

System.***out***.println(maria.toString());

System.***out***.println("Llistat per unitats");

System.***out***.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

System.***out***.println(maria.numMedicamentsPerQueden());

System.***out***.println();

System.***out***.println("El malalt " + joan.getNom() + " té:");

System.***out***.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

System.***out***.println(joan.toString());

System.***out***.println("Llistat per unitats");

System.***out***.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

System.***out***.println(joan.numMedicamentsPerQueden());

System.***out***.println();

System.***out***.println("Els malalts es prenen totes les píndoles!!!!!");

System.***out***.println();

*prendreTotsElsMedicamentsComprats*(maria);

*prendreTotsElsMedicamentsComprats*(maria);

System.***out***.println("La " + maria.getNom() + " es pren totes les píndoles que té");

System.***out***.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

System.***out***.println("El malalt " + maria.getNom() + " té:");

System.***out***.println(maria.toString());

System.***out***.println("Llistat per unitats");

System.***out***.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

System.***out***.println(maria.numMedicamentsPerQueden());

System.***out***.println();

System.***out***.println("La " + joan.getNom() + " es pren totes les píndoles que té");

System.***out***.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

System.***out***.println("El malalt " + joan.getNom() + " té:");

System.***out***.println(joan.toString());

System.***out***.println("Llistat per unitats");

System.***out***.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

System.***out***.println(joan.numMedicamentsPerQueden());

}

}

**Sessió 2**

**MedicamentPindoles.java**

**package** Practica3Sessio2;

**public** **class** MedicamentPindoles **implements** Comparable {

**private** **final** String medicament;

**private** **final** **int** unitats;

**private** **int** preses;

**public** MedicamentPindoles (String nom, **int** unitats) {

**if** (!(nom **instanceof** String)) **throw** **new** IllegalArgumentException("Nom no valid");

**this**.medicament = nom;

**if** (unitats < 0) **throw** **new** IllegalArgumentException("Nombre no valid, unitats inferior a 0");

**this**.unitats = unitats;

**this**.preses = 0;

}

**public** **int** quantesUnitatsQueden() {

**return** (**this**.unitats-**this**.preses);

}

**public** **void** prendrePindola() **throws** ExceptionMedicament {

**if** (**this**.unitats == **this**.preses) **throw** **new** ExceptionMedicament("No hi ha pindoles que prendre");

**if** (**this**.preses < **this**.unitats) {

**this**.preses++;

}

}

**public** **int** getPindolesPreses() {

**return** **this**.preses;

}

**public** String getNom() {

**return** **this**.medicament;

}

**public** **boolean** equals(Object med) {

**if** (**this** == med) {

**return** **true**;

} **else** **return** **false**;

}

**public** String toString() {

**return** ("Nom del medicament " + **this**.medicament + " amb " + **this**.quantesUnitatsQueden() + " píndoles de " + **this**.unitats);

}

**public** **int** compareTo(Object o) {

**if**(!(o **instanceof** MedicamentPindoles)) **throw** **new** ClassCastException();

**if** (**this**.unitats < ((MedicamentPindoles)o).unitats) {

**return** -1;

} **else** **if** (**this**.unitats == ((MedicamentPindoles)o).unitats) {

**return** 0;

} **else** **return** 1;

}

}

**Malalt.java**

**package** Practica3Sessio2;

//import java.util;

**import** java.util.Arrays;

**public** **class** Malalt **implements** Comparable {

**private** String nom; // Atribut 1

**private** MedicamentPindoles[] magatzem; // Atribut 2

**private** **int** medicaments; // Atribut 3

**private** **final** **int** inc; // Atribut 4

**public** Malalt (String nom, **int** max, **int** inc) {

**if**(!(nom **instanceof** String)) **throw** **new** IllegalArgumentException("Nom no valid");

**this**.nom = nom;

**if**(max < 0) **throw** **new** IllegalArgumentException("Nombre no valid, unitats inferior a 0");

**this**.magatzem = **new** MedicamentPindoles[max];

**this**.medicaments = 0;

**if**(inc < 0) **throw** **new** IllegalArgumentException("Nombre no valid, increment inferior a 0");

**this**.inc = inc;

}

**public** String getNom() {**return** **this**.nom;}

**public** **int** getMedicaments() {**return** **this**.medicaments;}

**public** **int** getIncrement() {**return** **this**.inc;}

**public** MedicamentPindoles getMedicamentPindoles(**int** quin) {

**if** (quin >= 0 && quin < **this**.medicaments) {

**return** **this**.magatzem[quin];

} **else** **throw** **new** IllegalArgumentException();

}

**public** MedicamentPindoles getMedicamentPindolesNoBuit() {

MedicamentPindoles ret = **null**;

**for** (**int** i=0; i<**this**.medicaments; i++) {

**if** (**this**.magatzem[i] != **null**) {

ret = **this**.magatzem[i];

} **else** ret = **null**;

}

**return** ret;

}

**public** **void** comprarMedicamentPindoles(MedicamentPindoles p) {

**int** pos = 0;

**boolean** trobat = **false**;

**if** (**this**.medicaments >= **this**.magatzem.length) {

MedicamentPindoles[] nouMagatzem = **new** MedicamentPindoles[**this**.medicaments + **this**.inc];

**for** (**int** j = 0; j < **this**.medicaments; j++) {

nouMagatzem[j] = **this**.magatzem[j];

}

**this**.magatzem = nouMagatzem;

}

// buscar posicio a la que va

**for** (**int** i = 0; i < **this**.medicaments && !trobat; i++) {

**if** (magatzem[i].getNom().compareTo(p.getNom()) > 0) {

// hem trobat el lloc

pos = i;

trobat = **true**;

} **else** **if** (magatzem[i].getNom().compareTo(p.getNom()) == 0) {

**if** (magatzem[i].quantesUnitatsQueden() < p.quantesUnitatsQueden()) {

pos = i;

trobat = **true**;

}

}

}

**if** (!trobat)

pos = **this**.medicaments;

// fer desplaçament

**for** (**int** i = **this**.medicaments; i >= pos; i--) {

**this**.magatzem[i + 1] = **this**.magatzem[i];

}

**this**.magatzem[pos] = p;

**this**.medicaments++;

}

**public** **void** comprarMedicamentPindoles(String p, **int** pindoles) {

**if** (pindoles < 0) **throw** **new** IllegalArgumentException();

MedicamentPindoles obj = **new** MedicamentPindoles(p, pindoles);

**this**.comprarMedicamentPindoles(obj);

}

**public** **int** totalPindolesQueden() {

**int** queden = 0;

// Seria fins el limit o fins l'ultim omplert

**for** (**int** i=0; i<**this**.medicaments; i++) {

queden += **this**.magatzem[i].quantesUnitatsQueden();

}

**return** queden;

}

**public** **int** totalPindolesQueden(String nom) {

**int** queden = 0;

**for** (**int** i=0; i<**this**.medicaments; i++) {

**if** (**this**.magatzem[i].getNom().equals(nom)) {

queden += **this**.magatzem[i].quantesUnitatsQueden();

}

}

**return** queden;

}

**public** **int** maximPindoles() {

String nom;

**int** queden, quedenAnt = 0;

**for** (**int** i=0; i<**this**.medicaments; i++) {

nom = **this**.magatzem[i].getNom();

queden = **this**.totalPindolesQueden(nom);

**if** (queden >= quedenAnt) {

quedenAnt = queden;

}

}

**return** quedenAnt;

}

**public** String numMedicamentsPerQueden() {

String retorn = "";

**int**[] quants, queden;

**int** num = 0;

queden = **new** **int**[**this**.medicaments];

quants = **new** **int**[**this**.medicaments];

**for** (**int** i = 0; i < **this**.medicaments; i++) {

//if (this.magatzem[i].quantesUnitatsQueden() > num) {

**int** x = *troba*(**this**.magatzem[i], queden, num);

**if**(x != -1) {

quants[x]++;

} **else** {

queden[num]=**this**.magatzem[i].quantesUnitatsQueden();

quants[num]=1;

num++;

}

}

retorn = *crear*(queden, quants, num);

**return** retorn;

}

**private** **static** **int** troba(MedicamentPindoles caixa, **int**[] queden, **int** quants) {

**for** (**int** i = 0; i < quants; i++) {

**if** (caixa.quantesUnitatsQueden() == queden[i]) {

**return** i;

}

}

**return** -1;

}

**private** **static** String crear(**int**[] queden, **int**[] quants, **int** quantes) {

String retorn="";

**for**(**int** i=0; i<quantes;i++) {

retorn = "Amb " + queden[i] + " pindoles queden " + quants[i] + " medicaments - ";

}

**return** retorn;

}

// A la hora d'eliminar un altre element fem el tractament tot i que tingui 0 píndoles

**public** **int** eliminarMedicamentPindoles() {

**int** retorn = 0;

**for** (**int** i=0; i<**this**.medicaments; i++) {

**if** (**this**.magatzem[i].quantesUnitatsQueden()==0) {

**for** (**int** j=i; j<**this**.medicaments; j++) {

**this**.magatzem[j] = **this**.magatzem[j++];

}

**this**.magatzem[**this**.medicaments] = **null**;

**this**.medicaments--;

retorn++;

}

}

**return** retorn;

}

**public** MedicamentPindoles[] donaMedicamentsBuits() {

**int** j=0;

MedicamentPindoles[] retorn;

retorn = **new** MedicamentPindoles[**this**.medicaments];

**for** (**int** i=0; i<**this**.medicaments; i++) {

**if** (**this**.magatzem[i].quantesUnitatsQueden()==0) {

retorn[j] = **this**.magatzem[i];

j++;

}

}

**return** retorn;

}

**public** String toString() {

String retorn = "";

**for** (**int** i = 0; i < **this**.medicaments; i++) {

retorn += (i+1) + ". "+**this**.magatzem[i].toString()+"\n";

}

**return** retorn;

}

**public** **boolean** equals(Object o) {

String[] arrayA, arrayB;

**if** (((Malalt) o) **instanceof** Malalt) {

String a = **this**.numMedicamentsPerQueden();

String b = ((Malalt) o).numMedicamentsPerQueden();

arrayA = a.split("-", **this**.medicaments);

arrayB = b.split("-", ((Malalt) o).getMedicaments());

Arrays.*sort*(arrayA);

Arrays.*sort*(arrayB);

a = Arrays.*toString*(arrayA);

b = Arrays.*toString*(arrayB);

**if** (a == b) {

**return** **true**;

} **else**

**return** **false**;

} **else**

**return** **false**;

}

**public** **int** compareTo(Object o) {

**if**(!(o **instanceof** MedicamentPindoles)) **throw** **new** ClassCastException();

**if** (**this**.totalPindolesQueden() < ((Malalt)o).totalPindolesQueden()) {

**return** -1;

} **else** **if** (**this**.totalPindolesQueden() == ((Malalt)o).totalPindolesQueden()) {

**return** 0;

} **else** **return** 1;

}

**public** String medicamentMenys() {

String retorn = "j";

**int** j = 0;

MedicamentPindoles aux = **this**.magatzem[j];

**for** (**int** i = 1; i < **this**.medicaments; i++) {

**if** (aux.compareTo(**this**.magatzem[i]) == -1) {

retorn = **this**.magatzem[i].getNom();

} **else** **if** (aux.compareTo(**this**.magatzem[i]) == 0) {

**if** (**this**.magatzem[i].getNom().compareTo(aux.getNom()) >= 0) {

retorn = **this**.magatzem[i].getNom();

} **else** {

retorn = **this**.magatzem[i].getNom();

}

}

j++;

retorn = "Hola";

}

**return** retorn;

}

**public** **void** llistaOrdenatAscendent() {

MedicamentPindoles[] Ascendent = **new** MedicamentPindoles[**this**.medicaments];

**for** (**int** j = 0; j < **this**.medicaments; j++) {

Ascendent[j] = **this**.magatzem[j]; // no copia bé els objectes a cada posicio

}

Bombolla(Ascendent);

**for** (**int** i = 0; i < **this**.medicaments; i++) {

System.***out***.println("El medicament és " + Ascendent[i].getNom() + " i té "

+ Ascendent[i].quantesUnitatsQueden() + "pindoles");

}

}

**private** **void** Bombolla(Object Comparable[]) {

MedicamentPindoles aux;

**if** (((MedicamentPindoles[]) Comparable) **instanceof** MedicamentPindoles[]) { // si es array ficar que l'objecte

// sigui array

**for** (**int** i = 1; i <= **this**.medicaments; i++) {

**for** (**int** j = **this**.medicaments; j >= i; j--) {

**if** (((MedicamentPindoles) Comparable[j])

.quantesUnitatsQueden() < ((MedicamentPindoles) Comparable[j - 1]).quantesUnitatsQueden()) {

aux = ((MedicamentPindoles) Comparable[j]);

Comparable[i] = Comparable[j - 1];

Comparable[j - 1] = aux;

}

}

}

}

}

**public** **void** llistatOrdenatDescendent() {

**int** x = **this**.medicaments;

MedicamentPindoles[] Descendent = **new** MedicamentPindoles[**this**.medicaments];

MedicamentPindoles aux;

**for** (**int** i = 0; i <= **this**.medicaments; i++) {

Descendent[i] = **this**.magatzem[i];

}

Arrays.*sort*(Descendent);

**for** (**int** i = 0; i < **this**.medicaments; i++) {

**for** (**int** j = **this**.medicaments; j >= i; j--) {

aux = Descendent[i];

Descendent[i] = Descendent[j];

Descendent[j] = aux;

}

}

**for** (**int** i = 0; i < **this**.medicaments; i++) {

System.***out***.println(Descendent[i].getNom() + "-" + Descendent[i]);

}

}

}

**ExceptionMedicament.java**

**package** Practica3Sessio2;

**public** **class** ExceptionMedicament **extends** Exception {

**public** ExceptionMedicament (String message) {

System.***out***.println(message);

}

}

**Prova.java**

**package** Practica3Sessio2;

**import** java.util.Random.\*;

**public** **class** Prova {

**private** **static** **void** prendreTotsElsMedicamentsComprats(Malalt m) **throws** ExceptionMedicament {

**for** (**int** i=0; i<m.getMedicaments(); i++) {

MedicamentPindoles a = m.getMedicamentPindolesNoBuit();

*prendreDelMedicament*(a);

}

}

**private** **static** **void** prendreDelMedicament(MedicamentPindoles m) **throws** ExceptionMedicament {

**for** (**int** i = 0; i < m.quantesUnitatsQueden(); i++) {

m.prendrePindola();

}

}

**private** **static** String malaltAleatori (**int** i) {

**final** String[] malalts;

**final** String malalt1 = "Maria Fernandez";

**final** String malalt2 = "Joan Moll";

**final** String malalt3 = "David Jimenez";

**final** String malalt4 = "Guillem Mateo";

**final** String malalt5 = "Francesc Serrano";

**final** String malalt6 = "Martin Pedro";

**final** String malalt7 = "Perico de los Palotes";

**final** String malalt8 = "Aleix Zamora";

malalts = **new** String[8];

}

**public** **static** **void** main(String [] args) **throws** ExceptionMedicament {

Malalt error1 = **new** Malalt("Maria Fernandez", -5, 10);

Malalt joan = **new** Malalt("Joan Moll", 15, 10);

Malalt pepito = **new** Malalt("xxxxxx", 10, 10);

MedicamentPindoles ibuprofeno20 = **new** MedicamentPindoles("IBUPROFENO", 20);

error1.comprarMedicamentPindoles(ibuprofeno20);

MedicamentPindoles medicamentBuit = **new** MedicamentPindoles("AUXINA", 0);

MedicamentPindoles medicamentError = **new** MedicamentPindoles("ERROR", -4);

*prendreTotsElsMedicamentsComprats*(error1);

joan.comprarMedicamentPindoles(medicamentError);

pepito.comprarMedicamentPindoles(medicamentBuit);

*prendreTotsElsMedicamentsComprats*(pepito);

}

**public** **static** **void** main1(String [] args) {

}

}