

Enumeració

Definir

Una enumeració és una classe "especial" que defineix tots els valors possibles per a una variable d'aquesta classe, mitjançant un conjunt de constants.

→ Una enumeració és una forma de controlar els valors que se li pot donar a una variable.

La paraula `enum` s'usa per a definir una enumeració.

Per a les direccions de la brúixola tots els valors possibles són: nord, sud, est i oest.

```
public enum DireccioBruixola {  
    NORD, SUD, EST, OEST;  
}
```

Per als colors d'un semàfor tots els valors possibles són: roig, taronja i verd.

```
public enum ColorSemafor {  
    ROIG, TARONJA, VERD;  
}
```

Els identificadors dels valors s'escriuen en majúscula, ja que són constants.

Pots definir una enumeració amb altre tipus d'accés que no siga públic (com qualsevol classe).

L'accés als seus valors es fa amb `nomDLenumeració.nomDelValor`.

Per als dies de la setmana els valors possibles són: dilluns, dimarts, dimecres, dijous, divendres, dissabte i diumenge. El nom de l'enumeració s'usa com un tipus de dada.

```
public enum DiaSetmana {  
    DILLUNS, DIMARTS, DIMECRES, DIJOURS, DIVENDRES, DISSABTE, DIUMENGE;  
}
```

```
DiaSetmana dia;           // defineix dia de tipus DiaSetmana  
dia = DiaSetmana.DIJOURS; // assigna a dia el dijous, és a dir, DiaSetmana.DIJOURS
```

A la referència `dia`, no se li pot donar un valor que no estiga en el conjunt de valors definit en el `enum`.



Comparar

Amb un valor d'una enumeració, l'única operació que es fa és la igualtat i per a això s'usa el `==` (com si fora una primitiva).

```
DiaSetmana dia = DiaSetmana.DIUMENGE;
if (dia == DiaSetmana.DISSABTE || dia == DiaSetmana.DIUMENGE) {
    System.out.println("estem en el cap de setmana, és " + dia);
}
//visualitza estem en el cap de setmana, és DIUMENGE
```

Per defecte, quan es visualitza un element d'una enumeració obtenim el seu nom.

En un `switch`, no cal posar el nom de l'enumeració, és suficient amb el valor.

```
DiaSetmana dia = DiaSetmana.DIJOUS;
String astre = switch (dia) {
    case DILLUNS -> "Lluna";
    case DIMARTS -> "Mart";
    case DIMECRES -> "Mercuri";
    case DIJOUS -> "Júpiter";
    case DIVENDRES -> "Venus";
    case DISSABTE -> "Saturn";
    case DIUMENGE -> "Sol";
};
System.out.println(dia + " està dedicat a " + astre);
//visualitza DIJOUS està dedicat a Júpiter
```

Recorrer

Pots obtenir tots els valors d'una enumeració amb el mètode `values`. En l'exemple es recorren els valors de l'enumeració i es mostra el missatge si és dissabte o diumenge.

```
for (DiaSetmana dia : DiaSetmana.values()) {
    if (dia == DiaSetmana.DISSABTE || dia == DiaSetmana.DIUMENGE) {
        System.out.println("estem en el cap de setmana, és " + dia);
    }
}
//visualitza
estem en el cap de setmana, és DISSABTE
estem en el cap de setmana, és DIUMENGE
```

Posició

Pots obtenir la posició d'un valor dins d'una enumeració amb el mètode `ordinal`. Al primer element d'una enumeració li correspon la posició 0.

```
System.out.println(DiaSetmana.DIJOUS + ">>" + DiaSetmana.DIJOUS.ordinal());
//visualitza DIJOUS>>3
```



toString

Per defecte, el `toString` d'un valor d'una enumeració torna el seu nom, pots sobreescrivre el mètode `toString` i obtenir un text diferent.

```
public enum Dia {  
    LUNES      { public String toString(){ return "Dilluns"; } },  
    MARTES     { public String toString(){ return "Dimarts"; } },  
    MIÉRCOLES  { public String toString(){ return "Dimecres"; } },  
    JUEVES     { public String toString(){ return "Dijous"; } },  
    VIERNES    { public String toString(){ return "Divendres"; } },  
    SÁBADO     { public String toString(){ return "Dissabte"; } },  
    DOMINGO    { public String toString(){ return "Diumenge"; } },  
}
```

```
Dia hui = Dia.LUNES;  
System.out.println("hui és " + hui); // visualitza hui és Dilluns
```

Pots obtenir el nom del valor de l'enumeració amb el mètode `name`.

```
Dia hui = Dia.MIÉRCOLES;  
System.out.println("hui és " + hui + " l'element és " + hui.name());  
// visualitza hui és Dimecres l'element és MIÉRCOLES
```

Obtindre

Pots obtenir un valor d'una enumeració a partir d'un text mitjançant el mètode `valueOf`, el text ha de coincidir amb el nom de la constant de l'enumeració. Llança l'excepció `IllegalArgumentException` si el text no està definit en l'enumeració.

```
String cadena = "DIMART";  
DiaSetmana ds = DiaSetmana.valueOf(cadena);
```

Enumeracions amb camps

A Java la declaració `enum` defineix una classe, i una classe pot incloure mètodes i camps.

El constructor és privat (sols s'usa en la definició dels valors) i defineix els valors dels camps.

Cada valor de l'enumeració s'escriu seguit, entre parèntesis, pels arguments necessaris per al constructor (els valors dels camps).

En l'exemple següent, l'enumeració `Aeroport` conté els codis IATA dels aeroports, té dos camps: ciutat i país. L'enumeració té els getters dels camps i el `toString`.



```

public enum Aeroport {
//valors
    AMSC("Amsterdam", "Països Baixos"),
    ATH("Atenes", "Grècia"),
    BCN("Barcelona", "Espanya"),
    BIO("Bilbao", "Espanya"),
    CDG("París", "França"),
    CUU("Chihuahua", "Mèxic"),
    DME("Moscou", "Rússia"),
    FCO("Roma", "Itàlia");
//camps
    private String ciutat;
    private String pais;
// constructor
    private Aeroport(String ciutat, String pais) {
        this.ciutat = ciutat;
        this.pais = pais;
    }
// mètodes
    public String getCiutat() {
        return ciutat;
    }
    public String getPais() {
        return pais;
    }
    @Override
    public String toString() {
        return super.toString() + " " + ciutat + " " + pais;
    }
}

```

L'enumeració s'usa de la mateixa forma que una classe.

```

Aeroport a = Aeroport.CUU; // a és un valor de l'enumeració
if (a.getPais().equals("Mèxic")){ // obté el país de a
    System.out.println(a.toString()); // visualitza CUU Chihuahua Mèxic
}

```

L'enumeració [Planeta](#) defineix els planetes del sistema solar, té un nom (que és el valor de l'enumeració), dos camps: la massa i el radi, i quatre mètodes: obtindre la massa ([getMassa](#)), obtindre el radi ([getRadi](#)), obtindre la gravetat ([getGravetatEnLaSuperficie](#)) i obtindre el pes ([getPesEnLaSuperficie](#)).

El constructor és privat, només el pot cridar la pròpia enumeració quan crea els valors possibles de l'enumeració. El constructor té dos paràmetres la massa i el radi del planeta, cada valor de l'enumeració usará els valors entre parèntesis en la construcció.

```

public enum Planeta {
// valors, es creen cridant al constructor

```



```

MERCURI(3.303e+23, 2.4397e6),
VENUS(4.869e+24, 6.0518e6),
TERRA(5.976e+24, 6.37814e6),
MART(6.421e+23, 3.3972e6),
JUPITER(1.9e+27, 7.1492e7),
SATURN(5.688e+26, 6.0268e7),
URA(8.686e+25, 2.5559e7),
NEPTUN(1.024e+26, 2.4746e7);
//camps
private final double massa;      // en quilograms
private final double radio;      // en metres
//constructor
private Planeta(double massa, double radi) {      // es rep la massa i el radi del planeta
    this.massa = massa;
    this.radi = radi;
}
//constant G de gravitació
public static final double G = 6.67300E-11;
//mètodes
public double getGravetatEnLaSuperficie() {
    return G * massa / (radi * radi);
}
public double getPesEnLaSuperficie(double laMassa) {
    return laMassa * getGravetatEnLaSuperficie();
}
public double getMassa() {
    return massa;
}
public double getRadi() {
    return radi;
}
}

```

El codi següent visualitza el radi de la terra

```
System.out.println("el radi del planeta " + Planeta.TERRA + " és " + Planeta.TERRA.getRadi() + "m");
```

El codi següent visualitza el pes d'una persona en els diferents planetes.

```

double pes = 87.4;
System.out.println("pes en la terra " + pes + " Kg");
double massa = pes / Planeta.TERRA.getGravetatEnLaSuperficie();
for (Planeta planeta : Planeta.values()) {
    System.out.printf("El teu pes en %s és %f Kg\n", planeta, planeta.getPesEnLaSuperficie(massa));
}

```



Exercicis

1.

Realitza l'enumeració per a representar el sexe d'una persona.



2.

Realitza l'enumeració per a representar un descompte. Hi ha 4 tipus de descomptes: MENYSD4, DE4A12, DE13A60 y MESD60 (els números són edats), amb els valors corresponents 100%, 50%, 0% y 60% (és l'atribut).

En el main de la classe principal visualitza el preu final d'una entrada a un parc, una entrada val 16€, llig de teclat l'edat del comprador, assigna-li el descompte que li correspon i visualitza el preu final. Repeteix el procés mentre no s'entra un -1.

