

Exercicis Tema 8. Programació Orientada a Objectes I

Tots els exercicis d'aquest butlletí han d'estar dins d'un **package** anomenat **tema08** (seguint les regles que hem vist a classe, ex: `com.germangascon.tema08`).

1. Crea un programa per a poder representar informació sobre cotxes. Per a desenvolupar l'exercici:

a) Crea una classe anomenada **Principal** que és la que contindrà el mètode `main()`.

b) Crea una classe anomenada **Cotxe** que represente la següent informació:

- el model
- el color
- si la pintura és metal·litzada o no
- la matrícula
- el tipus de cotxe, que pot ser MINI, UTILITARI, FAMILIAR o ESPORTIU
- l'any de fabricació
- la modalitat de l'assegurança, que pot ser a tercers o a tot risc

Fes un constructor amb valors inicials (els que tú vullgues), un altre constructor amb paràmetres, els getters, els setters i un mètode que es diga `imprimirCotxe()` que mostre el model i el color del cotxe.

- Crea varios cotxes utilitzant els diferents constructors i mostra les seues característiques per pantalla.
- Canvia alguns dels atributs dels cotxes i torna a mostrar les seues característiques.

2. Escriu un programa que dispose d'una classe per a representar les assignatures d'un cicle formatiu:

a) Una assignatura té un nom, un codi numèric i el curs en el qual s'impartix.

b) Els valors inicials han de proporcionar-se en el constructor.

c) La classe ha de tindre mètodes per obtenir els valors dels atributs.

d) El programa ha de construir un objecte amb els següents valors: nom "Programació", codi 1017, curs 1.

- A continuació, el programa ha d'imprimir els valors de l'objecte per pantalla.

3. Crea un programa per a gestionar alumnes. De cada alumne es desitja saber:

a) nia (número d'identificació de l'alumne)

b) nom

c) Cognoms

d) data de naixement

e) Grup al que pertany

f) Telèfon de contacte

El programa ha de presentar un menú que permeti gestionar alumnes similar al següent:

**** GESTIÓN ALUMNOS ****

1. Nuevo alumno ...

2. Baja de alumno ...

3. Consultas ...

0. Salir

- Nuevo alumno. Demanarà les dades de l'alumne, les validarà i l'afegirà. COMPTE! No pot haver dos alumnes amb el mateix nia.
- Baja de alumno. Demanarà el nia de l'alumne, si el nia correspon a un alumne enregistrar al sistema, demanarà un missatge de confirmació per a esborrarlo, en cas de confirmació positiva finalment serà esborrat.
- Buscar alumnos. Mostrarà el següent submenú:

**** CONSULTAS ****

1. Por grupo ...

2. Por edad ...

3. Por nia ...

4. Por apellidos ...

0. Volver al menú principal

- Por grupo. Sol·licitarà el grup i mostrarà tots els alumnes que pertanyen a eixe grup.
- Per edad. Preguntarà una edat en anys i el programa mostrarà tots aquells alumnes que tenen eixa edat. S'ha de calcular a partir de la data de naixement.
- Per nia. Preguntarà el nia, buscarà si existeix algun alumne amb eixe nia i en cas afirmatiu mostrarà les dades d'eixe alumne.
- Por apellidos. Preguntarà part de les lletres dels cognoms i buscarà aquells alumnes que els seus cognoms comencen per eixes lletres.
- Volver al menú principal. Tornarà al menú principal del programa.

A la següent taula es mostra un exemple de la sortida d'una de les consultes:

Nia	Nom	Cognoms	Data naixement	Grup	Telèfon
1036652	Juan	Mengual	20-02-1998	1DAM	965174521
1025456	María	Piera	05-03-1996	1DAM	965165854
1035622	Vicent	Tormo	18-09-1985	1DAM	632546598

CONSIDERACIONES:

Amb la finalitat d'agilitzar les proves a mesura que anem fent al programa, crea un mètode que

registre uns quants alumnes al sistema amb dades pseudo-aleatòries.

4. Crea un programa per a poder representar punts en dos dimensions (x,y). Per a desenvolupar l'exercici:

a) Crea una classe anomenada **Punt** que:

- Tinga dos atributs de tipus double.
- Tinga un constructor amb dos paràmetres de tipus double que inicialitzi els dos atributs.
- Tinga un constructor per defecte (sense paràmetres) que inicialitzi els dos atributs al valor desitjat.
- Tinga un getter i un setter per a cada un dels atributs.
- Tinga un mètode calcularDistancia que rebi un paràmetre de tipus Punt i que retorne un double.
- Tinga un mètode visualitzarPunt que mostri les coordenades del punt com es mostra en el següent exemple:

- (2.0, 6.2)

Una vegada creada la classe Punt, desde la classe Principal crea varios objectes Punt i mostra'ls per pantalla.

5. Crea una classe anomenada **Cercle** que:

- Tinga un atribut de tipus Punt (que representa el centre del cercle) i un de tipus double (que representa el radi del cercle).
- Tinga un constructor amb dos paràmetres, un de tipus Punt i un altre double que inicialitzi els dos atributs.
- Tinga un constructor amb tres paràmetres de tipus double que inicialitzi els dos atributs.
- Tinga un constructor per defecte (sense paràmetres) que inicialitzi els dos atributs al valor desitjat.
- Tinga un getter i un setter per a cada un dels atributs.
- Tinga un mètode calcularDistancia que rebi un paràmetre de tipus Punt i retorne un double (este mètode calcularà la distància des del centre del cercle fins al punt passat com a paràmetre).
- Tinga un mètode calcularArea que no rebi cap paràmetre i retorne un double.
- Tinga un mètode calcularPerimetre que no rebi cap paràmetre i retorne un double.
- Tinga un mètode visualitzarCercle que mostri la seua informació com es mostra en el següent exemple:

- Cercle de radi 2.5 cm situat en el punt (3.2, 4.0)

- NOTA: Si el centre del cercle està situat en el punt (0, 0), el missatge mostrat serà:

- Cercle de radi 3.2 cm situat en l'origen de coordenades.

Desde la classe Principal crea varios objectes Cercle i mostri on està situat, el seu perímetre, i l'àrea. Després mostra la distància que hi ha entre dos Cercles.

6. Crea un programa que permeti gestionar les bicicletes que té una tenda. De cada bicicleta es vol saber:
- la referència (que actua com a camp únic).
 - la marca
 - el model
 - el pes en Kg.
 - la grandària de les rodes en polsades
 - si té motor
 - la data de fabricació
 - el preu
 - el nombre d'existències que tenen d'eixa referència

El programa ha de mostrar un menú similar al següent:

** GESTIÓN DE BICICLETAS **

- 1.- Añadir bicicleta ...
- 2.- Vender bicicleta ...
- 3.- Consultar bicicleta ...
- 4.- Mostrar stock

0.- Salir

- *Añadir bicicleta.* Ens ha de preguntar les dades d'una bicicleta i afegir-la al sistema. S'ha de comprovar si la referència introduïda per l'usuari existeix, en cas afirmatiu, s'incrementarà l'stock d'eixa referència, si no existeix demanarà la resta de dades per donar una nova bicicleta d'alta.
- *Vender bicicleta.* Si escollim l'opció 2 ens demanarà la referència de la bicicleta que volem vendre, comprovarà que hi ha existències i en el cas que en queden, mostrarà un missatge del tipus "bicicleta xxxxx vendida correctament" i si no hi ha stock mostrarà un missatge indicant que no hi ha stock.
- *Consultar bicicleta.* Si escollim l'opció 3 ens mostrarà un submenú similar al següent:

** CONSULTA BICICLETA **

- 1.- Consultar por referencia ...
- 2.- Consultar por marca ...
- 3.- Consultar por modelo ...

0.- Volver al menú principal

- *Consultar por referencia.* Demanarà la referència i buscarà si existeix una bicicleta amb eixa referència. En cas afirmatiu mostrarà la informació associada a eixa bicicleta. Com que el camp referència asumim que és únic no pot haver cap altra bicicleta amb eixa referència.
 - *Consultar por marca.* Demanarà la marca i mostrarà les bicicletes d'eixa marca.
 - *Consultar por modelo.* Demanarà el model i mostrarà les bicicletes d'eixe model.
 - *Volver al menú principal.* Tornarà a mostrar el menú principal.
- *Mostrar stock.* Si escollim l'opció 4 del menú principal mostrarà l'stock que hi ha de cada bicicleta.
 - *Salir.* En escollir l'opció 0 eixirà del programa.

CONSIDERACIONS:

- Per evitar tindre que introduir dades cada vegada que inicies el programa per a poder fer proves, crea varies bicicletes per codi.

7. Crear una aplicació per a gestionar un xicotet centre de salut d'un poble per a atendre les urgències. Quan els pacients arriben al centre són enregistrats al sistema amb les següents dades:

- El número de la targeta sanitaria (Sip)
- Nom
- Sexe
- Edat
- Data i hora d'entrada al centre
- Sintomatologia

i esperen el seu torn a la sala d'espera.

Quan un pacient és atès es mesuren les seues constants vitals: la temperatura, les pulsacions per minut i els valors de la tensió arterial (sistòlica i diastòlica) es guarden en un atribut preRev que serà vector de 4 posicions.

Es desitja saber els pacients que estan sent atesos actualment així com un històric dels pacients atesos al llarg del mes en curs.

Sip	Nom	Sexe	Data d'entrada	Hora d'entrada	Edat	Sintomatologia	Temp	ppm	Ten Sis	Ten Dias	Data d'alta	Hora d'alta	Motiu de l'alta
94964183	Sonia Moran	M	15-12-2018	22:55	20	Dolor abdominal	36.5	75	130	60	16-12-2018	01:25	Millora
37109682	Juan Soler	V	15-12-2018	23:21	38	Mareig generalitzat	38.2	71	135	55	16-12-2018	01:39	Millora
52046314	Pedro Ferrer	V	16-12-2018	02:44	81	Dolor fort al pit	36.4	98	175	98	16-12-2018	02:59	Defunció
92367866	Sergio Signes	V	16-12-2018	05:21	45	Calfreds	40.1	115	180	110	16-12-2018	05:30	Derivació hospital

79058213	Ana Peralta	M	16-12-2018	07:50	21	Migranya intensa	36.7	68	125	62			
25152798	María Píera	M	16-12-2018	08:13	15	Contusió al braç dret							

Figura 1: exemple de l'històric de pacients

Suposem que el nombre màxim de pacients que atendrem en un dia serà de 40.

El programa mostrarà un menú similar al següent:

** URGENCIES **

1. Nuevo paciente ...

2. Atender paciente ...

3. Consultas ...

4. Alta médica ...

0. Salir

Veiem que s'ha de fer en cada cas:

- *Nuevo paciente.* Demanarà les dades del pacient i la sintomatologia (la data i hora d'entrada les obtindrem automàticament de la data i hora del sistema) i registrarà el pacient al sistema. S'ha de comprovar que el pacient no estiga ja sent atés actualment, si que poden existir varies entrades del mateix pacient però només una sense data d'alta.
- *Atender paciente.* El pacient és atés. Demanarà el sip del pacient, si es tracta d'un sip d'un pacient que està esperant a ser atés, es a dir, està enregistrat en el sistema i no te data d'alta, mostrarà les dades del pacient i demanarà les dades de les constants vitals i les registrarà al sistema.
- *Consultas.* Mostrarà un submenú similar al següent:

** CONSULTAS **

1. Por Sip ...

2. Por fechas ...

3. Estadísticas

3. Mostrar histórico mensual

0. Volver al menú principal

- *Consulta por Sip.* Sol.licitarà el Sip del pacient i, si existeix en el històric, mostrarà les dades del històric associades a eixe pacient. Recorda que un pacient pot ser atés més d'una

vegada al llarg del mes.

- *Consulta por fechas.* Sol.licitarà una data d'inici i una data finalització i mostrarà totes les dades del històric de pacients que han sigut atesos entre eixes dates. Si no se especifica data finalització, mostrarà desde la data d'inici fins la data actual.
 - *Estadísticas.* Mostrarà les següents estadístiques:
 - mitjana de la temperatura
 - mitjana de pulsacions per minut
 - mitjana dels dos valors de la tensió arterial
 - mitjana d'edad dels pacients
 - percentatge de pacients que són homes
 - percentatge de pacients que són dones
 - *Mostrar histórico mensual.* Mostrarà l'històric complet.
 - *Volver al menú principal.* Tornarà al menú principal del nostre programa.
- *Alta médica.* Demanarà el Sip del pacient i un motiu de l'alta i donarà l'alta mèdica al pacient. La data i l'hora d'alta les agafarà automàticament del sistema.
 - *Salir.* Eixirà del programa.

Amb la finalitat d'agilitzar les proves que anem fent al programa, crea un mètode que registre uns quants pacients al sistema amb dades pseudo-aleatòries.