# Ejer1-

Teoría. Creamos la clase Bombilla y la clase Televisor.

# Ejer2.- Tamagotchi

Vamos a crear la clase Tamagotchi para simular el clásico juego.

|  |
| --- |
| Tamagotchi |
| nombre  cantidadVida  maxVida |
| **Tamagotchi(String nombre1, int maxVida1)**  Constructor que inicializará los valores.CantidadVida se inicializará a cero  **visualiza()**  Muestra los datos del tamagotchi por pantalla  **juega()**  Sólo puede jugar si cantidadVida>=1, esta acción restará 1 de cantidadVida y mostrará por pantalla 5 veces ***Estoy jugando.***  **come()**  Sólo puede comer si cantidadVida<maxVida, esta acción sumará 1 de cantidadVida y mostrará por pantalla 5 veces ***Estoy comiendo***  ***Nota:* Estos System.out sólo sirven para comprobar el funcionamiento del tamagotchi, no se tendrá en cuenta como muestra de resultados.** |

Desde el método principal de la clase **Ejer3** mostraremos el siguiente menú y realizaremos las acciones oportunas:

*1.*Crea tamagotchi

*2.*Juega con tu mascota virtual

*3.*Da de comer a tu mascota virtual

*4.*Ver la vida de tu mascota

*5.*Fin

## Ejer3

Vamos a crear la clase CuentaCorriente, con las siguientes propiedades y comportamiento:

|  |
| --- |
| CuentaCorriente |
| numCuenta : Será el número de la cuenta corriente.  saldo : Saldo actual de la cuenta. |
| **CuentaCorriente(StringnumCuenta)**  Inicializar numCuenta con el String pasado como parámetro  **ingresaEfectivo(double cantidad)**  Pedirá por teclado el importe que queremos ingresar y se lo sumará al saldo  **booleanretiraEfectivo(double cantidad)**  Pedirá por teclado el importe que queremos retirar y comprobará que hay saldo suficiente.Si hay saldo, restará el importe, y devolverá true, en caso contrario no retirará el dinero y devolverá false.  **visualiza()**  Mostrará por pantalla la información de la cuenta corriente: Número de cuenta y saldo. |

El método main de la clase principal **Ejer2**,creará dos cuentas corrientes: La 001 y la 002, ambas con 0 € de saldo, posteriormente mostrará el siguiente menú por pantalla:

1.Ingresar en la cuenta 001

2.Ingresar en la cuenta 002

3.Retirar de la cuenta 001

4.Retirar de la cuenta 002

5.Visualizar ambas cuentas

6.Fin

Dicho menú se ejecutará, realizando las operaciones oportunas para cada opción hasta que el usuario elija la opción de fin.

Ejer5.-Se pide realizar una clase para manejar fechas

|  |
| --- |
| Fecha |
| private int dia;  private int mes;  privateintaño; |
| // Crear el constructor necesario  privateStringgetMesLetras()  - Devuelve una cadena que contiene el mes con letras: enero, febrero ... diciembre  privatebooleanesBisiesto()  - Devuelve true si el año es bisiesto y false en caso contrario  privateintgetDiasMes()  - Devuelve el número de días que tiene el mes. Teniendo en cuenta que si el año es bisiesto el mes de febrero tendrá 29 días.  publicbooleanesCorrecta()  - Devuelve true si la fecha es correcta.  publicStringtoString( )   * Devuelve la fecha con el formato largo 12 de diciembre de 2018 |

Se pedirán fechas por teclado (dia, mes y año) , se indicará si es correcta o no y posteriormente se mostrará la fecha larga:

12 de diciembre de 2018

Cuando se introduzca un 0 en el día, el programa terminará, sin pedir el mes y el año.

Ejer4: Crea la clase Alumno

|  |
| --- |
| Alumno |
| String nombre  Float nota  int edad |
| Alumno(String n) {  //Guarda el nombre que le pasamos como parámetro en la variable nombre  }  ponNota(int n) {  // Guarda la nota que le pasamos como parámetro en la variable nota  }  toString( ) {  // los datos del alumno.  } |

En el main tendremos un array con 5 alumnos.

Tareas a realizar:

1.Pedir el nombre de los 5 alumnos y crearlos.

2.Pedir la nota de los 5 alumnos

3.Mostrar la nota media.

4.Mostrar los alumnos que tienen la nota más alta que la nota media.

5.Mostrar el alumno con la peor nota.

# Ejer6

Se pide realizar una calculadora de números racionales. Para ello se pedirá primero un número racional, numerador y denominador, después la operación a realizar: “+”, “-“, “\*” ó “/” y finalmente el segundo número racional con el que operar.

Se tendrá que crear la siguiente clase:

|  |
| --- |
| NumeroRacional |
| int numerador;  int denominador; |
| //Constructor  publicNumeroRacional suma (NumeroRacional op2) {  }  publicNumeroRacional resta (NumeroRacional op2) {  }  publicNumeroRacionalmultiplicacion (NumeroRacional op2) {  }  publicNumeroRacionaldivision (NumeroRacional op2) {  } |

**Ejer7.-** Crear la clase LineaAutobus con las siguientes características

|  |
| --- |
| LineaAutobus |
| StringnumLinea;  int frecuencia;  int [ ] paradas; |
| //constructor  //set para frecuencia  //getters  //toString mostrará dos líneas:  Línea: 34 Frecuencia: 15 minutos  Paradas 152 – 14 – 178 – 195 – 45 – 66 – 39 |

Desde el main, se crearán 4 líneas de autobús con datos aleatorios, y se mostrarán por pantalla.

Ejer8

Crear la siguiente clase:

|  |
| --- |
| NIF |
| String parteNumerica  int num;  char letraFinal, letraInicial; |
| Dni(String cadena) {  //Desglosa cada una de las partes del dni.   1. Si es dni en letraInicial pondremos una #, pero no se mostrará. 2. Si es Nie, letraInicial será la primera letra y en la primera posición de los 7 números que le corresponden, se pondrá el número adecuado. Si la letra inicial no es correcta se pondrá todo a ceros. Ver ejemplo. 3. Si hay letras en la parte numérica, se pondrá igualmente todo a ceros. 4. Si el último carácter no es una letra, se calculará   /\* Ejemplos:   |  |  |  |  |  | | --- | --- | --- | --- | --- | | cadena | letraInicial | num | parteNumerica | letraFinal | | 125544f | # | 125544 | 00125544 | F | | y22j | Y | 1000022 | 000022 | J | | H35444d | # | 0 | 00000000 | 0 | | 12jj221m | # | 0 | 00000000 | 0 | | X25 | X | 25 | 0000025 | W | | 12345 | # | 12345 | 00012345 | V |   }  Dni (intnum) {  //guarda el numero en la parte numérica y calcula la letra que la guardará en letraFinal  }  booleandniCorrecto() {  /\*devuelve true si el DNI es correcto  la letra inicial sólo puede ser ‘x’, ‘y’ , ‘z’ ó ‘#‘  x 0 se cambiará por un 0 en la partenumérica  y 1 se cambiará por un 1 en la partenumérica  z 2 se cambiará por un 2 en la partenumérica  La letra final se calcula con el resto de divider el número entre 23 y con el siguienteorden:  TRWAGMYFPDXBNJZSQVHLCKE  \*/  }  String toString() {  //muestra el Dni completo, con el formato que se indica en el enunciado |

Para comprobar el funcionamiento: <https://www.letranif.com/>

Funcionamiento de la clase principal para comprobar la clase:

Se pedirán documentos por teclado y se mostrará si es correcto o no. Si es correcto se mostrará con el siguiente formato:

DNI NNNNNNNN-L

NIE LNNNNNNN-L

Se pedirán hasta que se introduzca un documento vacío.

Ejer9.-Se pide simular múltiples lanzamientos de dos dados.  Para ello utilizaremos la siguiente clase:

|  |
| --- |
| **Lanzador** |
| int dado1, dado2 |
| **public void lanza()** {          //Genera números aleatorios para dado1 y dado2 entre 1 y 6          //Recuerda:(int)(Math.random()\*6)+1;      }  **public boolean sonIguales()** {          //Devuelve true si ambos dados son iguales, false en caso contrario      }  **public void muestraDados()** {          //Muestra el valor que tiene cada dado      } |
|  |
|  |

**Funcionamiento:**  El programa pedirá un nº para saber cuántas veces ha de lanzarse los dados, se mostrará por pantalla la simulación de todas las tiradas de los dados, y al final se mostrará el nº de veces que los dados han sido iguales.  Se volverá a pedir un nº, y se repetirá el juego hasta que el nº introducido sea cero.

Ejer10.-

|  |
| --- |
| Cliente |
| String nombre  float cuota  String deporte |
| //constructor getters & Setters |

Se tendrá un array con 10 clientes y se mostrará un menú permitiendo dar de alta y mostrar clientes. Se admiten nombres repetidos.

2ª parte.-

Baja. Se mostrará un menú como el siguiente:

000 Pepe Natación

002 Juan Alpinismo

007 Eva Submarinismo

Siendo el primer número la posición que ocupa el cliente en el array. Se pide al usuario que introduzca el número de cliente, se mostrará **toda** su información y se pedirá una confirmación de baja, una vez confirmado, se procederá a poner a null ese cliente.

Ejer11

 Crearemos la clase alumno para que tenga las notas de los 3 trimestres. El expediente tendrá el siguiente formato:LE/001, siendo 001 un contador que se irá incrementando cada vez que se crea un alumno

|  |
| --- |
| Alumno |
| String expediente  String nombre  int [] nota |
| Alumno(String n) {  //Guarda el nombre que le pasamos como parámetro en la variable nombre  }  ponNota(int n, int evaluacion) {  // Guarda la nota que le pasamos como parámetro en la variable nota en la  //evaluación indicada.  }  toString( ) {  // los datos del alumno.  } |
|  |

En el main tendremos un array con 10 alumnos.

Tareas a realizar:

1.Generar 10 nombres de manera aleatoria para los alumnos con el siguiente array:

String nombres[] = {"Ana”,”José”,”Luis”,”Eva”,”Sara”,”Julia”,”David”,”María”,”Manuel”,”Andrea”}

2.Rellenar con números aleatorios todas las notas, y secuencialmente todos los números de expediente.

3.Mostrar la nota media.

4.Mostrar los alumnos que tienen la nota más alta que la nota media.

5.Mostrar el alumno con la peor nota media