



DESARROLLO DE APLICACIONES MULTIPLATAFORMA PROGRAMACIÓN

EXAMEN 1aEV PARCIAL

11/12/2020

TEMA 4. ESTRUCTURAS DE ALMACENAMIENTO BÁSICAS (ARRAYS).

TEMA 5. PROGRAMACIÓN ORIENTADA A OBJETOS.

TURNO VERDE

Realizar los siguientes ejercicios y entregar en la tarea correspondiente del Aula Virtual, un fichero .ZIP denominado “Ex1aEv2P” con las respuestas.

Para la elaboración de los ejercicios utilizar el software “Geany”, “JDK” y Java API.
Se tendrá en cuenta para la calificación los siguientes aspectos:

- *Uso correcto de elementos básicos del lenguaje.*
- *Uso correcto de tipos de datos primitivos y clases/funciones de la API.*
- *Uso correcto de condicionales, bucles y estructuras básicas de almacenamiento.*
- *Uso correcto de conceptos de programación orientada a objetos.*
- *Formato de entrada y salida por pantalla.*
- *Correcto funcionamiento del programa (debe compilar sin errores).*
- *Interfaz de usuario “amigable”.*
- *Originalidad y eficiencia del algoritmo diseñado.*
- *Realización de todos los ejercicios (se valorará negativamente ejercicios en blanco).*

Realizar un programa para la **Gestión de un Hotel, con Clientes, Habitaciones y Reservas** de temática general. Se requiere una serie de procesos como son: dar de alta y baja Habitaciones, dar de alta y baja Reservas de Habitaciones por Clientes y diferentes tipos de Listados. Para el almacenamiento de los datos se utilizarán *Arrays*.

Los clientes estarán precargados, se podrá dar de alta y baja habitaciones, cada habitación tiene un tipo y un precio por día. Una reserva consiste en que un cliente reserve una habitación en una fecha dada (fecha de inicio), posteriormente se podrá anular o facturar la reserva indicando la fecha de finalización y el importe total que se calculará a partir de los días reservados y el precio de la habitación. Al final del enunciado tenéis un ejemplo de ejecución.

El examen dispone de una carpeta plantilla llamada “Ex1aEv2P_verde_plantilla”, antes de empezar renombrarla a “Ex1aEv2P_verde”.

Dentro de la carpeta “Ex1aEv2P_verde”, crear los archivos .java necesarios, finalmente, eliminar los ficheros .class y comprimir la carpeta completa en un fichero .zip.

Realizar los siguientes ejercicios donde se describe paso a paso la implementación de la aplicación.



EJERCICIO 1 (ptos).

Implementar una clase principal “GestionHotel”, definiendo las estructuras de almacenamiento necesarias para guardar un array de Clientes, un array de Habitaciones y un array de Reservas (máximo 10 elementos cada uno, controlando que no se exceda) con los siguientes nombres:

tClientes, tHabitaciones, tReservas

Además, dispone de un método “mostrarMenu” para mostrar un menú de opciones y conocer la opción seleccionada. El programa principal mostrará el menú, pedirá una opción y realizará el proceso correspondiente de forma repetitiva hasta que se seleccione la opción de “Salir”.

```
System.out.println("\t1.- Listado de Clientes.");  
System.out.println("\t2.- Alta de Habitación.");  
System.out.println("\t3.- Baja de Habitación.");  
System.out.println("\t4.- Listado de Habitaciones.");  
System.out.println("\t5.- Alta de Reserva.");  
System.out.println("\t6.- Baja de Reserva (Anular/Facturar).");  
System.out.println("\t7.- Listado de Reservas.");  
System.out.println("\t0.- Salir.");
```

El programa principal (*interfaz gráfica*) debe ser lo más amigable posible, simple y funcional, mostrando mensajes indicativos de los resultados de los procesos.

Los procesos de “alta” y “baja” en un array de elementos necesitan conocer previamente la posición/índice donde insertar o eliminar el elemento y además, no debe dejar “huecos”. No permitir dar de alta si el array correspondiente está lleno o dar de baja si está vacío, informando al usuario de esta situación.

EJERCICIO 2 (ptos).

Implementar una clase “Cliente” con los siguientes atributos, un constructor básico y los métodos *getters & setters* necesarios:

```
String dni           // PK  
String nombre  
LocalDate fecNac
```

Los atributos representan: el DNI del cliente, su nombre y su fecha de nacimiento.

- El método “getNombre” devolverá un *String* con el nombre del cliente en mayúsculas y sin espacios en blanco al principio y al final.
- Los métodos “get&set” de la fecha realizarán la conversión correspondiente entre *String* y *LocalDate*.



EJERCICIO 3 (ptos).

Implementar una clase “Habitacion” con los siguientes atributos, un constructor básico y los métodos *getters & setters* necesarios:

```
int id // PK
String descripcion
TIPO tipo
double precio
```

Los atributos representan: el identificador de la habitación, su descripción, un tipo de habitación y su precio por día de estancia.

- El atributo “tipo” será un enumerado que podrá tener los siguientes valores: *SIMPLE*, *DOBLE* y *OTRO*. Este dato no es obligatorio, por lo que puede tener valor nulo.
- El método “getDescripcion” devolverá un *String* con la descripción de la habitación donde la primera letra esté en mayúsculas y el resto en minúsculas. Previamente serán eliminados los posibles espacios en blanco al principio y al final.
- El método “getTipo” devolverá un *String* realizando la conversión necesaria y teniendo en cuenta que puede tener valor nulo.

EJERCICIO 4 (ptos).

Implementar una clase “Reserva” con los siguientes atributos, un constructor básico y los métodos *getters & setters* necesarios:

```
String dniCliente // PK
int idHabitacion // PK
LocalDate fecIni // PK
LocalDate fecFin
double total
```

Los atributos representan: el DNI del cliente, el identificador de la habitación, la fecha de inicio y fin de la estancia en la habitación reservada y el importe total de la estancia.

- Los métodos “get&set” de las fechas realizarán la conversión correspondiente entre *String* y *LocalDate*.
- El atributo “fecFin” no se lee por consola, si no que es establecido (fecha actual) en el proceso de “Baja de Reserva”.
- El atributo “total” no se lee por consola, si no que es calculado a partir del precio/día de la habitación en el proceso de “Baja de Reserva”.



EJERCICIO 5 (ptos).

No existe el Proceso de alta y baja de Clientes en la clase principal:

- Crear un método “preCargaDatos” en una clase “Auxiliar”, que inicialice el *array* de clientes con algunos objetos cliente por defecto. Utilizar este método en el programa principal al inicio.
- La clase “Cliente” tendrá un método “existeCliente” para conocer si existe o no, un determinado cliente en el *array* de clientes (devuelve un *boolean*).
- Implementar el proceso de “Listado de Clientes” debidamente formateado y mostrando todos los elementos del *array* aunque no tenga elementos (“null”).

EJERCICIO 6 (ptos).

Implementar el Proceso de alta y baja de Habitaciones en la clase principal:

- La clase “Habitacion” dispondrá de un atributo adicional *numHabitacion* que nos servirá como índice para inserciones y borrados en el *array* correspondiente.
- La clase “Habitacion” tendrá un método “existeHabitacion” para conocer si existe o no, una determinada habitación en el *array* de habitaciones (devuelve un *boolean*).
- La clase “Habitacion” tendrá un método “altaHabitacion” para dar de alta una habitación determinada en el *array* de habitaciones, realizando las comprobaciones previas oportunas y devolviendo si el proceso se ha realizado correctamente o no.
Comprobaciones: La habitación no debe existir previamente.
- La clase “Habitacion” tendrá un método “bajaHabitacion” para dar de baja una habitación determinada en el *array* de habitaciones, realizando las comprobaciones previas oportunas y devolviendo si el proceso se ha realizado correctamente o no.
Comprobaciones: La habitación debe existir previamente.
- Implementar el proceso de “Listado de Habitaciones” debidamente formateado y mostrando todos los elementos del *array* aunque no tenga elementos (“null”).



EJERCICIO 7 (ptos).

Implementar el Proceso de alta y baja de Reserva en la clase principal:

- La clase “Reserva” dispondrá de un atributo adicional *numReserva* que nos servirá como índice para inserciones y borrados en el *array* correspondiente.
- La clase “Reserva” tendrá un método “existeReserva” para conocer si existe o no, una determinada reserva en el *array* de reservas (devuelve un *boolean*).
- La clase “Reserva” tendrá un método “altaReserva” para dar de alta una reserva determinada en el *array* de reservas, realizando las comprobaciones previas oportunas y devolviendo si el proceso se ha realizado correctamente o no.
Comprobaciones: La reserva no debe existir previamente, el cliente debe existir y la habitación debe existir.
- La clase “Reserva” tendrá un método “bajaReserva” para dar de baja una reserva determinada en el *array* de reservas, realizando las comprobaciones previas oportunas y devolviendo si el proceso se ha realizado correctamente o no.
Comprobaciones: La reserva debe existir y no debe estar ya anulada o facturada.
- ATENCIÓN: Este proceso de baja anterior NO elimina elementos de una reserva determinada del *array*, sino que establece la fecha de fin (fecha actual) y el importe total de la reserva.
Si la fecha de fin es posterior a la fecha de inicio (*Facturar*), el importe total se calculará teniendo en cuenta el precio por día de la habitación y la diferencia en días entre la fecha de inicio y la fecha de fin. Si la fecha de fin es anterior a la fecha de inicio (*Anular*), el importe total será 0.
- Implementar el proceso de “Listado de Reservas” debidamente formateado y mostrando todos los elementos del *array* aunque no tenga elementos (“null”).



EJERCICIO 8 (ptos).

Adicionalmente, realizar lo siguiente:

- Implementar un método “esDNICorrecto” en una clase “Auxiliar” que permita conocer si es correcto el formato de un DNI en un *String* (no hace falta probarlo). Tener en cuenta todas las características/restricciones para ese formato.
- Implementar un método “esFechaCorrecta” en una clase “Auxiliar” que permita conocer si es correcto el formato “dd/mm/yyyy” en un *String* y usarlo en el método “altaReserva”. Tener en cuenta al menos, 3 o 4 características/restricciones para ese formato.
- Implementar un método “ordenarArrayReservasPorIdHabitacion” en una clase “Auxiliar” que permite ordenar el *array* de Reservas por “id” de habitación y usarlo en el proceso de “Listado de Reservas”. Utilizar algún algoritmo de ordenación visto en clase.
- El proceso de “Listado de Reservas” debe mostrar además del dni/identificador del Cliente y Habitación, sus nombres/descripciones. Implementar algún mecanismo que permita mostrar esta información teniendo en cuenta que si se elimina un cliente o habitación no se eliminará su entrada correspondiente en la reserva.
- Implementar un método “gastoAcumulado” en la clase “Cliente” que calcule el gasto total acumulado por un cliente en todas sus reservas. Utilizar este método en el proceso de “Listado de Clientes”.
- Implementar un método “numReservasSolicitadas” en la clase “Habitacion” que calcule el número total de reservas que ha tenido una habitación. Utilizar este método en el proceso de “Listado de Habitaciones”.



EJEMPLO DE EJECUCIÓN

MENU

- 1.- *Listado de Clientes.*
- 2.- *Alta de Habitacion.*
- 3.- *Baja de Habitacion.*
- 4.- *Listado de Habitaciones.*
- 5.- *Alta de Reserva.*
- 6.- *Baja de Reserva (Anular/Facturar).*
- 7.- *Listado de Reservas.*
- 0.- *Salir.*

Opcion? 1

LISTADO DE CLIENTES

00-. DNI: 1 NOMBRE: CLIENTE1 FEC.NAC.: 01/01/2001 GASTO ACUMULADO: 0
01-. DNI: 2 NOMBRE: CLIENTE2 FEC.NAC.: 02/02/2002 GASTO ACUMULADO: 0
02-. DNI: 3 NOMBRE: CLIENTE3 FEC.NAC.: 03/03/2003 GASTO ACUMULADO: 0
03-. DNI: 4 NOMBRE: CLIENTE4 FEC.NAC.: 04/04/2004 GASTO ACUMULADO: 0
04-. DNI: 5 NOMBRE: CLIENTE5 FEC.NAC.: 05/05/2005 GASTO ACUMULADO: 0
05-. null
06-. null
07-. null
08-. null
09-. null

MENU

- 1.- *Listado de Clientes.*
- 2.- *Alta de Habitacion.*
- 3.- *Baja de Habitacion.*
- 4.- *Listado de Habitaciones.*
- 5.- *Alta de Reserva.*
- 6.- *Baja de Reserva (Anular/Facturar).*
- 7.- *Listado de Reservas.*
- 0.- *Salir.*

Opcion? 2

Introduce id de Habitacion: 11

Introduce descripcion de Habitacion: hab11

Introduce tipo (0:NULL,1:SIMPLE,2:DOBLE,3:OTRO) de Habitacion: 1



Región de Murcia
Consejería de Educación
y Cultura



UNIÓN EUROPEA
Fondo Social Europeo

I.E.S. Ginés Pérez Chirinos



Av. Dr. Robles, 1
30400 Caravaca de la Cruz
Tel. (+34) 968707621 – Fax (+34) 968700202
Web: www.ieschirinos.com
Email: 30001308@murciaeduca.es



Introduce precio de Habitación: 10

Alta de Habitación correcta.

MENU

- 1.- Listado de Clientes.*
- 2.- Alta de Habitación.*
- 3.- Baja de Habitación.*
- 4.- Listado de Habitaciones.*
- 5.- Alta de Reserva.*
- 6.- Baja de Reserva (Anular/Facturar).*
- 7.- Listado de Reservas.*
- 0.- Salir.*

Opcion? 2

Introduce id de Habitación: 22

Introduce descripción de Habitación: hab22

Introduce tipo (0:NULL,1:SIMPLE,2:DOBLE,3:OTRO) de Habitación: 2

Introduce precio de Habitación: 20

Alta de Habitación correcta.

MENU

- 1.- Listado de Clientes.*
- 2.- Alta de Habitación.*
- 3.- Baja de Habitación.*
- 4.- Listado de Habitaciones.*
- 5.- Alta de Reserva.*
- 6.- Baja de Reserva (Anular/Facturar).*
- 7.- Listado de Reservas.*
- 0.- Salir.*

Opcion? 5

Introduce dni Cliente de Reserva: 1

Introduce id Habitación de Reserva: 11

Introduce fecha Ini (dd/mm/yyyy) de Reserva: 01/12/2020

Alta de Reserva correcta.

MENU

- 1.- Listado de Clientes.*



- 2.- *Alta de Habitación.*
- 3.- *Baja de Habitación.*
- 4.- *Listado de Habitaciones.*
- 5.- *Alta de Reserva.*
- 6.- *Baja de Reserva (Anular/Facturar).*
- 7.- *Listado de Reservas.*
- 0.- *Salir.*

Opcion? 5

Introduce dni Cliente de Reserva: 2

Introduce id Habitación de Reserva: 22

Introduce fecha Ini (dd/mm/yyyy) de Reserva: 01/01/2021

Alta de Reserva correcta.

MENU

- 1.- *Listado de Clientes.*
- 2.- *Alta de Habitación.*
- 3.- *Baja de Habitación.*
- 4.- *Listado de Habitaciones.*
- 5.- *Alta de Reserva.*
- 6.- *Baja de Reserva (Anular/Facturar).*
- 7.- *Listado de Reservas.*
- 0.- *Salir.*

Opcion? 7

LISTADO DE RESERVAS

00-. DNI: 1	NOMBRE: CLIENTE1	ID: 11	DESCRIPC.: Hab11	FECINI: 01/12/2020
FECFIN:	TOTAL: 0,00			
01-. DNI: 2	NOMBRE: CLIENTE2	ID: 22	DESCRIPC.: Hab22	FECINI: 01/01/2021
FECFIN:	TOTAL: 0,00			
02-. null				
03-. null				
04-. null				
05-. null				
06-. null				
07-. null				
08-. null				
09-. null				

MENU



- 1.- *Listado de Clientes.*
- 2.- *Alta de Habitación.*
- 3.- *Baja de Habitación.*
- 4.- *Listado de Habitaciones.*
- 5.- *Alta de Reserva.*
- 6.- *Baja de Reserva (Anular/Facturar).*
- 7.- *Listado de Reservas.*
- 0.- *Salir.*

Opcion? 6

Introduce dni Cliente de Reserva: 1

Introduce id Habitación de Reserva: 11

Introduce fec Ini (dd/mm/yyyy) de Reserva: 01/12/2020

Baja de Reserva correcta.

MENU

- 1.- *Listado de Clientes.*
- 2.- *Alta de Habitación.*
- 3.- *Baja de Habitación.*
- 4.- *Listado de Habitaciones.*
- 5.- *Alta de Reserva.*
- 6.- *Baja de Reserva (Anular/Facturar).*
- 7.- *Listado de Reservas.*
- 0.- *Salir.*

Opcion? 6

Introduce dni Cliente de Reserva: 2

Introduce id Habitación de Reserva: 22

Introduce fec Ini (dd/mm/yyyy) de Reserva: 01/01/2021

Baja de Reserva correcta.

MENU

- 1.- *Listado de Clientes.*
- 2.- *Alta de Habitación.*
- 3.- *Baja de Habitación.*
- 4.- *Listado de Habitaciones.*
- 5.- *Alta de Reserva.*
- 6.- *Baja de Reserva (Anular/Facturar).*
- 7.- *Listado de Reservas.*
- 0.- *Salir.*



Opcion? 7

LISTADO DE RESERVAS

00-. DNI: 1 NOMBRE: CLIENTE1 ID: 11 DESCRIPC.: Hab11 FECINI: 01/12/2020
FECFIN: 08/12/2020 TOTAL: 70,00
01-. DNI: 2 NOMBRE: CLIENTE2 ID: 22 DESCRIPC.: Hab22 FECINI: 01/01/2021
FECFIN: 08/12/2020 TOTAL: 0,00
02-. null
03-. null
04-. null
05-. null
06-. null
07-. null
08-. null
09-. null

MENU

- 1.- Listado de Clientes.
- 2.- Alta de Habitación.
- 3.- Baja de Habitación.
- 4.- Listado de Habitaciones.
- 5.- Alta de Reserva.
- 6.- Baja de Reserva (Anular/Facturar).
- 7.- Listado de Reservas.
- 0.- Salir.

Opcion? 1

LISTADO DE CLIENTES

00-. DNI: 1 NOMBRE: CLIENTE1 FEC.NAC.: 01/01/2001 GASTO ACUMULADO: 70
01-. DNI: 2 NOMBRE: CLIENTE2 FEC.NAC.: 02/02/2002 GASTO ACUMULADO: 0
02-. DNI: 3 NOMBRE: CLIENTE3 FEC.NAC.: 03/03/2003 GASTO ACUMULADO: 0
03-. DNI: 4 NOMBRE: CLIENTE4 FEC.NAC.: 04/04/2004 GASTO ACUMULADO: 0
04-. DNI: 5 NOMBRE: CLIENTE5 FEC.NAC.: 05/05/2005 GASTO ACUMULADO: 0
05-. null
06-. null
07-. null
08-. null
09-. null

MENU



Región de Murcia
Consejería de Educación
y Cultura



UNIÓN EUROPEA
Fondo Social Europeo

I.E.S. Ginés Pérez Chirinos



Av. Dr. Robles, 1
30400 Caravaca de la Cruz
Tel. (+34) 968707621 – Fax (+34) 968700202
Web: www.ieschirinos.com
Email: 30001308@murciaeduca.es



- 1.- Listado de Clientes.
- 2.- Alta de Habitación.
- 3.- Baja de Habitación.
- 4.- Listado de Habitaciones.
- 5.- Alta de Reserva.
- 6.- Baja de Reserva (Anular/Facturar).
- 7.- Listado de Reservas.
- 0.- Salir.

Opcion? 4

LISTADO DE HABITACIONES

00-. ID: 11 DESCRIPCION: Hab11	TIPO: SIMPLE PRECIO: 10,00 RESERVAS: 1
01-. ID: 22 DESCRIPCION: Hab22	TIPO: DOBLE PRECIO: 20,00 RESERVAS: 1
02-. null	
03-. null	
04-. null	
05-. null	
06-. null	
07-. null	
08-. null	
09-. null	

MENU

- 1.- Listado de Clientes.
- 2.- Alta de Habitación.
- 3.- Baja de Habitación.
- 4.- Listado de Habitaciones.
- 5.- Alta de Reserva.
- 6.- Baja de Reserva (Anular/Facturar).
- 7.- Listado de Reservas.
- 0.- Salir.

Opcion? 0