

IIP (E.T.S. de Ingeniería Informática)
Curso 2016-2017

Práctica 5. Selección: cálculo del precio de una entrada de cine

Profesores de IIP
Departamento de Sistemas Informáticos y Computación
Universitat Politècnica de València



Índice

1. Contexto y trabajo previo a la sesión de prácticas	1
2. Descripción del problema	1
3. Diseño de clases	2
4. Diseño del método que calcula el precio final	2
5. Actividades de laboratorio	3

1. Contexto y trabajo previo a la sesión de prácticas

El objetivo principal de esta práctica es trabajar con la sintaxis y la semántica de las instrucciones condicionales simples y múltiples en Java que se han visto en el “*Tema 5. Estructuras de control: selección*”. En concreto se propone diseñar una Clase Tipo de Dato que representa una entrada de cine y una Clase Programa que, utilizando información adicional, calcula el precio final de la entrada.

2. Descripción del problema

Supóngase que se está desarrollando una aplicación para la venta de entradas de cine y, en concreto, se plantea el problema de cómo calcular el precio final de una entrada a partir de su precio base, teniendo en cuenta una serie de descuentos posibles de diversa índole.

Cada entrada de cine en nuestra aplicación tiene asociado el título de la película, el cine donde se proyecta, la sesión, y su precio base. Nótese que, con el fin de simplificar el problema, no se han incluido otros datos típicos, como por ejemplo, la sala de proyección, la fecha, la butaca, etc.



Circunstancias diversas pueden hacer que el precio final de la entrada sea diferente a su precio base. Estas circunstancias, en nuestro caso, se concretan en las siguientes:

- Las personas mayores de 65 años sólo pagan el 30 % del precio base de la entrada, independientemente del día.
- El día del espectador (generalmente miércoles no festivos ni vísperas de festivos), el precio final se reduce un 20 % respecto al precio base.
- Los festivos (domingos incluidos), el precio final se incrementa un 20 % respecto al precio base.
- Las vísperas de festivo, que no sean festivos, el precio final se incrementa un 10 % respecto al precio base.
- Espectadores con tarjeta de cliente (menores de 65 años) obtienen una reducción del 20 % sobre el precio final, excepto si es el día del espectador.

El propósito de esta práctica es el diseño y uso de una clase que represente una entrada de cine, y de un programa que, a partir de datos como la edad, el tipo de día, etc., calcule su precio final.

3. Diseño de clases

Para la resolución del problema planteado, se pide implementar las siguientes clases:

- La Clase “*Tipo de Dato*” **Entrada** para representar las entradas de cine, que debe incluir:
 - Variables de instancia: **titulo** y **cine** de tipo **String** y **horaSesion** de tipo **Instante** (implementada por el alumno en la práctica anterior).
 - Variables de clase (**static**): constantes para definir el precio base, la edad de 65 años y los diferentes descuentos.
 - Métodos: constructor, **get**, **set**, **toString**, **equals**, y **precioFinal**, el cual devuelve el precio final de la entrada dados la edad del espectador (**int**) y cuatro **boolean** que indican respectivamente si es el día del espectador, si el día es festivo, si el día es víspera de festivo, y si el espectador tiene tarjeta de cliente. En la sección 4 se describe con detalle el análisis de casos necesario para la implementación del método **precioFinal**.
- La Clase Programa **VentaEntradas** con el método **main**, que debe pedir los datos de la entrada, crear el correspondiente objeto de tipo **Entrada**, pedir el resto de datos necesarios (edad del espectador, tipo de día, y tarjeta de cliente), llamar al método de instancia de la clase **Entrada** que calcula su precio final y mostrarlo por pantalla.

4. Diseño del método que calcula el precio final

Una implementación nada recomendable del método que calcula el precio final de la entrada sería utilizar una aproximación por “fuerza bruta” que considerara todas las posibles combinaciones de datos posibles, esto es, las 32 combinaciones posibles de 5 valores lógicos (edad ≥ 65 , día del espectador, festivo, víspera, tarjeta de cliente). Afortunadamente, un análisis detallado de todas las circunstancias nos permite resolver el problema con un número de comparaciones menor. En la Figura 1 se muestra un esquema de análisis que resuelve el problema del cálculo con 5 comparaciones en el peor de los casos.

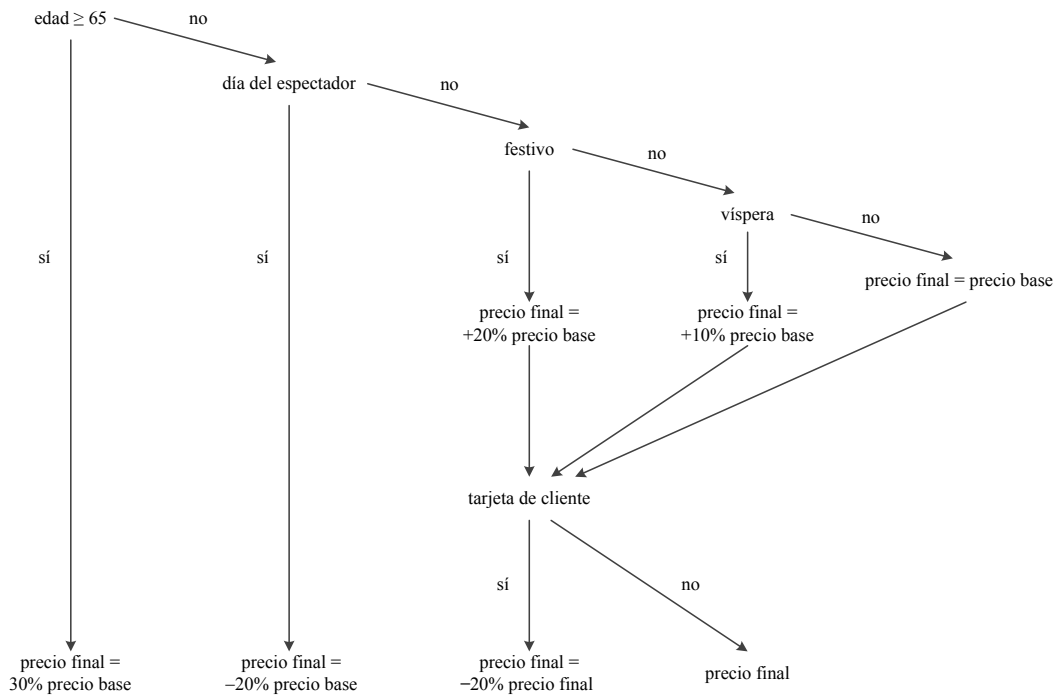


Figura 1: Esquema de análisis para determinar el precio final de la entrada.

5. Actividades de laboratorio

Actividad 1: Crear el proyecto BlueJ pract5

1. Descarga en Downloads los ficheros `Entrada.java` y `VentaEntradas.java`, disponibles en la carpeta `Recursos/Laboratorio/Práctica 5/Castellano/código` de la *PoliformaT* de IIP.
2. Abre *BlueJ* y crea un nuevo proyecto `pract5` en el directorio de trabajo de la asignatura (`iip`).
3. Añade al proyecto `pract5` las clases `Entrada.java` y `VentaEntradas.java` con la opción *Edición/Agregar clase desde archivo...* de *BlueJ*.

Actividad 2: Usar la clase Instante del paquete `iipUtil`

Para poder tener acceso a la clase `Instante`, realizada en la práctica anterior, hay que añadir la ubicación del proyecto `pract4` donde está situado el package `iipUtil` mediante la opción *Herramientas/Preferencias/Librerías* de *BlueJ*. Debes reiniciar *BlueJ* para que los cambios hechos en la librería de clases tengan efecto.

A partir de este momento, ya puedes importar en los proyectos *BlueJ* que construyas los elementos del paquete `iipUtil`, siguiendo para dicha importación la notación estándar Java. En concreto, al comienzo de la clase `Entrada` verás escrita la instrucción de importación:

```
import iipUtil.Instante;
```

Actividad 3: Completar la clase Entrada

Completa en el proyecto `pract5` la clase `Entrada`. En concreto, debes completar donde están los comentarios que así lo indican. Una vez finalizada la clase debe incluir:

1. las variables de instancia y de clase que se indican en la sección 3.
2. Un método constructor con los parámetros necesarios para inicializar las variables de instancia, es decir, un título de película, un cine, y dos enteros para la hora y los minutos de la sesión.
3. Los métodos `get` y `set` para cada una de las variables de instancia.
4. El método `toString` que devuelve la descripción de la entrada (`String`). Un ejemplo del formato de salida para una entrada en concreto sería el siguiente:

```
"Anochece en La India", proyectada en Cines Babel, a las 22:30
Precio base: 7.60 euros
```

5. El método `equals` que comprueba si dos objetos de tipo `Entrada` son iguales, es decir, si todos sus atributos coinciden.
6. El método `precioFinal` con perfil:

```
public double precioFinal(int edad, boolean diaEspec, boolean festivo,
    boolean visp, boolean tCliente)
```

que implementa el esquema de análisis de la Figura 1 para calcular el precio final de la entrada, redondeándolo a céntimos de €.

Actividad 4: Completar la clase programa VentaEntradas

Completa el método `main` de la clase `VentaEntradas`. En concreto, debes completar donde están los comentarios que así lo indican.

El método `main` de esta clase debe:

1. Solicitar y leer los datos para crear un objeto de tipo `Entrada` (la parte de lectura del título ya está incluida, completa el resto).
2. Crear un objeto `Entrada`.
3. Solicitar y leer el resto de datos en este orden: la edad del espectador, si es el día del espectador (SÍ/NO), si es festivo (SÍ/NO), si es víspera de festivo (SÍ/NO) y si el espectador tiene tarjeta de cliente (SÍ/NO). Fíjate en los ya implementados para completar los que falten.
4. Llamar al método `precioFinal` para calcular el precio final de la entrada.
5. Mostrar por pantalla el precio final de la entrada sólo con dos cifras decimales, para ello se puede usar la instrucción `System.out.printf` o el método `format` de la clase `String` (esta parte se da ya implementada).

Se puede asumir que los datos de entrada son siempre correctos, por ejemplo, que el título de la película y la sala son valores posibles, las horas y los minutos son enteros, entre 0 y 23 y entre 0 y 59, respectivamente, que las preguntas sobre el tipo de día se responden de forma apropiada, etc.

Actividad 5: Comprobar la corrección de la implementación

La Tabla 1 muestra algunos casos que se pueden utilizar para comprobar la corrección del código desarrollado (calculados a partir de un precio base de 7,60 €).

Tabla 1: Casos de prueba.

Edad	Día del espectador	Festivo	Víspera de Festivo	Tarjeta cliente	Precio final
65	SÍ	NO	NO	NO	2,28
72	NO	SÍ	NO	NO	2,28
89	NO	NO	SÍ	NO	2,28
77	NO	NO	NO	NO	2,28
34	SÍ	NO	NO	SÍ	6,08
42	SÍ	NO	NO	NO	6,08
17	NO	SÍ	NO	SÍ	7,30
27	NO	SÍ	NO	NO	9,12
53	NO	NO	SÍ	SÍ	6,69
21	NO	NO	SÍ	NO	8,36
28	NO	NO	NO	SÍ	6,08
64	NO	NO	NO	NO	7,60

Para comprobar que la clase **Entrada** funciona correctamente hay dos posibilidades:

- Utilizar el Banco de Objetos para crear un objeto de la clase **Entrada**, y sobre este objeto ir invocando el método **precioFinal** cambiando paulatinamente los valores de cada parámetro hasta completar las 12 combinaciones de la Tabla 1.
- Ejecutar el método **main** de la clase **VentaEntradas** introduciendo cada una de las combinaciones de valores descritas.

Para cada caso, el resultado de la invocación del método **precioFinal** debe ser el que se indica en la Tabla 1.

Actividad 6: Optimizar el código

Si analizas cuidadosamente el código del método **main**, te darás cuenta que hay casos en los que un dato debería pedirse o no en función de los datos ya obtenidos. Por ejemplo, si es el día del espectador, ya no es necesario preguntar si el día es festivo o víspera de festivo, o incluso si el espectador tiene o no tarjeta de cliente. Modifica el código del método **main** para no pedir al usuario más datos de los estrictamente necesarios en el cálculo del precio final de la entrada de cine.