1.- Descripción de la tarea.

Caso práctico

Gerd Altmann Pixabay License.

María está entrenándose en la programación orientada a objetos. Está empezando a realizar pruebas, utilizando métodos, creando objetos, en fin, familiarizándose poco a poco con el lenguaje Java y alguna de las bibliotecas o librerías que lleva incorporadas el paquete básico de desarrollo.

Ahora mismo está construyendo pequeñas aplicaciones para usar algunas clases que han realizado algunos de sus compañeros (clases `Dado` y `CuentaBancaria`), así como las clases relacionadas con el uso de las fechas y las horas en Java (`LocalDate`, `LocalTime`, `LocalDateTime`). También está practicando con el uso de la **documentación JavaDoc** que se ha generado para esas clases y está observando lo útil que puede llegar a ser disponer de este tipo de documentación. ¡Le están literalmente "salvando la vida"!

¿Qué te pedimos que hagas?

Las actividades a realizar se centrarán en el uso y manipulación de objetos mediante operaciones sencillas, trabajando con métodos de las clases `CartonBingo`, `Bombo` y `LocalDate` / `LocalTime` para obtener una serie de resultados, **siendo obligatorio el uso de objetos de esas clases** según corresponda en cada ejercicio.

Para entender cómo funcionan todas estas clases es fundamental que consultes la **documentación JavaDoc** de cada una de ellas, bien consultando la documentación de la API de Java (en el caso de la clases `LocalDate` y `LocalTime` ya que son clases **propias de Java** como puede ser también la clase Scanner que ya conoces) o bien a través de la documentación JavaDoc que se te proporciona en esta tarea (para el caso de las clases `CartonBingo` y `Bombo` ya que son clases propias y no pertenecen a la API de Java).

Para ello, es fundamental que consultes la **documentación JavaDoc** de cada una de esas clases, bien a través de la documentación de la API de Java (en el caso de las clases `LocalDate` / `LocalTime`) bien a través de la documentación JavaDoc que se te proporciona en esta tarea (para el caso de las clases `CartonBingo` y `Bombo`).

Debes usar obligatoriamente el proyecto NetBeans que se te proporciona en el apartado **información de interés** de esta tarea, el cual incluye una **biblioteca específica para esta tarea** con las clases `CartonBingo` y `Bombo`, las cuales, como decíamos, no forman parte de la API de Java.

1.1.- Ejercicio 1: creación y uso de cartones de bingo.

La clase `CartonBingo` **trata de modelar el funcionamiento de un cartón en un juego de bingo. Este cartón contiene, entre otros atributos, una cantidad determinada de números que se deberán ir tachando si una bola con ese número es extraída del bombo del juego.**

Es **muy importante** que le eches un vistazo a la **documentación JavaDoc** de esta clase antes de empezar a trabajar con ella. Así podrás ver qué tipo de información contiene, qué constructores proporciona, qué restricciones se le pueden aplicar, cuáles son los métodos disponibles, etc. Recuerda que tendrás que hacer el import correspondiente para poder usar esta clase en tu programa.

Cartón del bingo BINGOTRASS. Fran Jiménez (elaboración propia). (CC BY)

Para practicar con la creación y uso de objetos de tipo `CartonBingo`, vamos a crear distintos cartones: debemos crear un cartón para cada uno de los tres personajes de nuestras historias *(María, Ada y Juan)*. En los tres casos se tratará de objetos (instancias) de la clase `CartonBingo`.

En el proyecto de trabajo que se proporciona para esta tarea dispones de un programa llamado **Ejercicio1** donde tendrás que llevar a cabo las siguientes acciones paso a paso:

Presentación del ejercicio. Mediante un mensaje inicial se presenta el ejercicio y se muestra la fecha actual de ejecución. Ten en cuenta que esta fecha debe calcularse automáticamente y debe reflejar la **fecha actual** en la que se ejecute el programa ***(no es un texto fijo)***. Consultar la clase `LocalDate` en la API de Java te será de utilidad.

Declarar tres variables referencia a objetos instancia de la clase `CartonBingo`. Como sabes, debes utilizar un nombre descriptivo para cada una de ellas que permita identificarlas fácilmente en el código. Solo debes declarar las variables, **no les asignes ninguna instancia aún**.

Instanciar objetos de la clase `CartonBingo` a los cuales apuntarán cada una de las variables declaradas anteriormente.

> Recuerda que cuando decimos que una variable de tipo referencia va a "apuntar" a un objeto significa simplemente que va a almacenar en una variable (de tipo referencia, obviamente) **aquello que devuelva el operador new** tras la invocación al **constructor** (es decir, una referencia a la zona de memoria donde se encuentran realmente los atributos del objeto).

Antes de crear los cartones correctamente, haremos un par de **intentos con errores** para comprobar que **funciona el lanzamiento de excepciones** por parte de sus constructores:

Intenta crear un cartón con una fecha de sorteo NO válida (utiliza la API de LocalDate para establecer la fecha de la semana pasada a partir de la fecha actual). Recuerda que para instanciar un objeto de una determinada clase debes invocar a un constructor de esa clase mediante el operador new (analiza la API de la clase `CartonBingo` y utiliza un constructor adecuado que te permita establecer la fecha de sorteo del cartón).

Dado que, en este caso, sabemos que la llamada al constructor va a fallar seguro, debes usar un bloque try-catch para capturar la excepción y gestionar ese error. Si este error no lo gestiona tu código, lo gestionará la propia máquina virtual de Java y el programa se **detendrá abruptamente** ante el error, cosa que **nunca debe suceder**. Tu código **siempre debe tener previsto** este tipo de posibles errores a través de la captura de las excepciones que puedan ocurrir.

Utilizando el constructor adecuado, intenta crear otro cartón con una cantidad de números no válida (por ejemplo, **30 números**). Nuevamente, el programa lanzará una excepción de error que deberás gestionar correctamente para evitar que tu programa "aborte".

Una vez hemos probado algunos errores que puede lanzar el constructor de la clase vamos a crear cartones correctamente:

Utilizando el constructor más adecuado, crea **un cartón para María con 18 números y fecha de sorteo 5 de marzo de 2023.** Lo que devuelva el operador new al invocar al constructor es lo que

tendrás que asignar a la variable referencia que represente el cartón de María, con la cual podrás manipular este objeto.

De nuevo utilizando el constructor más adecuado, crea ahora **un cartón para Ada**, sin indicar nada más y asigna la nueva instancia creada a la variable que represente al cartón de Ada.

Por último, crea el **cartón de Juan utilizando los valores por omisión** (constructor sin parámetros) y asigna la nueva instancia creada a la variable que represente al cartón de Juan.

Obtén la siguiente información de los distintos cartones creados y mostrarla por pantalla. Para ello tendrás que usar algunos de los métodos de la clase `CartonBingo`:

Total de cartones que se han creado hasta el momento.
Identificador del cartón de María.
Cantidad de números que tiene el cartón de Ada.
Fecha de sorteo del cartón de Juan y del cartón de María.
Lista de números del cartón de Ada.
Número de días que faltan para el sorteo en el que participará el cartón de María (observa que el número de días siempre se compara con la fecha actual, por lo que la salida que obtengas puede ser distinta a las salidas de ejemplo).
Total de cartones que participan en el sorteo de hoy.

Ahora llevaramos a cabo distintas **operaciones sobre los cartones que hemos creado**:

Marca los números **desde el número 20 hasta el número 35 ambos incluidos** en cada uno de los tres cartones.
Muestra la **lista de números que se han podido marcar correctamente en cada uno de los tres cartones** tras realizar la operación anterior.
Indica, de los tres cartones, **a cuál de ellos le quedan menos números.**

Para finalizar, vamos a mostrar el **estado final de los cartones**:

Muestra el estado final del cartón de María.
Muestra el estado final del cartón de Ada.
Muestra el estado final del cartón de Juan.

Resultado del ejercicio

En este ejercicio no hay que introducir datos de entrada, por tanto la salida no dependerá de ninguna entrada del usuario por teclado. Sin embargo, conviene recordar que la clase CartonBingo durante la creación de cada uno de los cartones genera los números que contendrá el cartón de forma aleatoria.

Por ello, es **prácticamente imposible que todos aquellos resultados relacionados con estos números** (cantidad de números pendientes en un cartón, listado de números marcados en un cartón, cartón con menos números pendientes, etc.) **coincidan exactamente** con los resultados que se muestran en los ejemplos de ejecución (aunque sí deben ser similares la salida de tu programa y la de los ejemplos).

Aquí tienes una muestra de un ejemplo de ejecución:

Mostrar retroalimentación

```
Ejercicio 1. Creación y uso de cartones de bingo
------------------------------------------------
Fecha ACTUAL de ejecución: 05/11/2022

Creación de los cartones para cada uno de los personajes (uso de constructores)
-------------------------------------------------------------------------------
Intentando crear un cartón para el sorteo celebrado hace una semana (29/10/2022).
¡ERROR! Fecha 29/10/2022 no válida, no se puede crear un cartón para una fecha inválida o para un sortec
```

```
Intentando crear un cartón con 30 números (cantidad no válida)
¡ERROR! Cantidad de números no válida, no se puede crear un cartón con 30 números (debe tener un mínimo

Creamos ahora cartones válidos para nuestros tres jugadores...

Creando un cartón de 18 números y fecha de sorteo del 5 de Marzo de 2023 para María:
+ Se ha creado correctamente un cartón para María con 18 números!

Creando un cartón para Ada, sin indicar nada más:
+ Se ha creado correctamente un cartón para Ada con 15 números!

Creando un cartón para Juan, utilizando los valores por defecto:
+ Se ha creado correctamente un cartón para Desconocido con 15 números!


Obtención de información de los cartones creados
------------------------------------------------
Total de cartones que se han creado:  3
Identificador del cartón de María:  C-000
Cantidad de números que tiene el cartón de Ada:  15 números
Fecha de sorteo del cartón de Juan (05/11/2022) y del cartón de María (05/03/2023)
Lista de números del cartón de Ada:  05 06 08 12 15 16 20 22 30 31 36 39 43 44 45
Número de días que faltan hasta el sorteo en el que participa María:  120
Cartones participan en el sorteo de hoy:  2

Realización de algunas operaciones con los cartones creados
-----------------------------------------------------------
Marcamos los números desde el 20 hasta el número 35 ambos incluidos en los tres cartones...
De los números anteriores, mostramos la lista de números que sí se han podido marcar en cada uno de los
- En el cartón de María estaban los números: 21 27 28 29 31 33 35
- En el cartón de Ada estaban los números: 20 22 30 31
- En el cartón de Juan estaban los números: 27 29 32 33 35

Al jugador/a que le quedan menos números pendientes de marcar en su cartón es... Juan, que le quedan 10

Estado final de todos los cartones
----------------------------------

+---------------------------------+
| N18          BINGOTRASS - María |
+---------------------------------+
|   07  10  19  ··  XX  ··  44    |
|   08  11  XX  XX  XX  39  46    |
|   ··  12  XX  XX  XX  42  49    |
+---------------------------------+
|  ID: C-000 - SORTEO:05/03/2023  |
+---------------------------------+

+---------------------------------+
| N15            BINGOTRASS - Ada |
+---------------------------------+
|   ··  06  ··  12  XX  XX  43    |
|   05  ··  08  15  XX  36  44    |
|   ··  ··  ··  16  XX  39  45    |
+---------------------------------+
|  ID: C-001 - SORTEO:05/11/2022  |
+---------------------------------+

+---------------------------------+
| N15    BINGOTRASS - Desconocido |
+---------------------------------+
|   ··  05  12  XX  XX  XX  41    |
|   04  ··  14  XX  ··  ··  44    |
|   ··  07  17  XX  ··  39  47    |
+---------------------------------+
|  ID: C-002 - SORTEO:05/11/2022  |
+---------------------------------+

El programa ha finalizado!!
```

Otro ejemplo de ejecución podría ser el siguiente (fíjate bien que la salida es igual a la anterior, pero varían los números de cada cartón y los resultados relacionados con esos números)

Mostrar retroalimentación

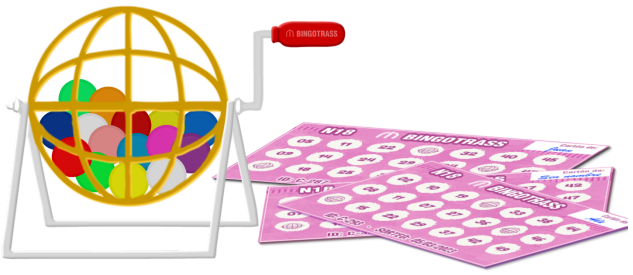
```
Ejercicio 1. Creación y uso de cartones de bingo
------------------------------------------------
Fecha ACTUAL de ejecución: 05/11/2022

Creación de los cartones para cada uno de los personajes (uso de constructores)
--------------------------------------------------------------------------------
Intentando crear un cartón para el sorteo celebrado hace una semana (29/10/2022).
¡ERROR! Fecha 29/10/2022 no válida, no se puede crear un cartón para una fecha inválida o para un sorteo

Intentando crear un cartón con 30 números (cantidad no válida)
¡ERROR! Cantidad de números no válida, no se puede crear un cartón con 30 números (debe tener un mínimo

Creamos ahora cartones válidos para nuestros tres jugadores...

Creando un cartón de 18 números y fecha de sorteo del 5 de Marzo de 2023 para María:
+ Se ha creado correctamente un cartón para María con 18 números!

Creando un cartón para Ada, sin indicar nada más:
+ Se ha creado correctamente un cartón para Ada con 15 números!

Creando un cartón para Juan, utilizando los valores por defecto:
+ Se ha creado correctamente un cartón para Desconocido con 15 números!


Obtención de información de los cartones creados
------------------------------------------------
Total de cartones que se han creado:  3
Identificador del cartón de María:  C-000
Cantidad de números que tiene el cartón de Ada:  15 números
Fecha de sorteo del cartón de Juan (05/11/2022) y del cartón de María (05/03/2023)
Lista de números del cartón de Ada:  01 03 05 06 10 17 21 23 24 25 27 34 38 40 46
Número de días que faltan hasta el sorteo en el que participa María:  120
Cartones participan en el sorteo de hoy:  2

Realización de algunas operaciones con los cartones creados
-----------------------------------------------------------
Marcamos los números desde el 20 hasta el número 35 ambos incluidos en los tres cartones...
De los números anteriores, mostramos la lista de números que sí se han podido marcar en cada uno de los
- En el cartón de María estaban los números: 20 21 23 27 32
- En el cartón de Ada estaban los números: 21 23 24 25 27 34
- En el cartón de Juan estaban los números: 21 26 27 30 31 32

Al jugador/a que le quedan menos números pendientes de marcar en su cartón es... Juan, que le quedan 9 n

Estado final de todos los cartones
----------------------------------

+--------------------------------+
| N18          BINGOTRASS - María |
+--------------------------------+
|   ··  05  10  14  XX  XX  38   |
|   ··  06  12  17  XX  36  41   |
|   01  08  ··  XX  XX  37  47   |
+--------------------------------+
|  ID: C-000 - SORTEO:05/03/2023  |
+--------------------------------+
```

```
+--------------------------------+
| N15            BINGOTRASS - Ada |
+--------------------------------+
|   01  06  17  XX  XX  ··  46   |
|   03  10  XX  XX  XX  38  ··   |
|   05  ··  ··  ··  XX  40  ··   |
+--------------------------------+
|  ID: C-001 - SORTEO:05/11/2022  |
+--------------------------------+

+--------------------------------+
| N15    BINGOTRASS - Desconocido |
+--------------------------------+
|   02  07  ··  ··  19  XX  XX   |
|   04  ··  ··  ··  XX  XX  47   |
|   ··  09  13  15  XX  XX  49   |
+--------------------------------+
|  ID: C-002 - SORTEO:05/11/2022  |
+--------------------------------+

El programa ha finalizado!!
```

Recuerda que para resolver esta actividad debes utilizar obligatoriamente instancias de la clase **CartonBingo** junto con sus métodos.

1.2.- Ejercicio 2: ¡a jugar! Uso de las clases CartonBingo y Bombo para simular una partida

En el ejercicio anterior hemos conocido el funcionamiento de la clase `CartonBingo`. En este ejercicio **utilizaremos esa clase junto con la clase** `Bombo` para **simular un sorteo** real de bingo con **DOS jugadores participantes**.



Fran Jiménez (elaboración propia). (CC BY)

En este ejercicio crearemos **un cartón para cada uno de los dos jugadores** e iremos **sacando bolas del bombo** y **marcándolas en cada uno de los cartones**. Cuando un jugador consiga una **línea completa** en su cartón **podrá cantar ¡Línea!**, mientras que cuando un jugador consiga marcar **todos los números** de su cartón cantará **¡Bingo!** y ganará la partida.

Para resolver este ejercicio deberás consultar y utilizar la documentación JavaDoc de la clase `CartonBingo` y de la clase `Bombo` que se proporciona.

En el proyecto de trabajo que se proporciona para esta tarea dispones de un programa llamado **Ejercicio2** donde tendrás que llevar a cabo las siguientes acciones paso a paso:

Presentación del ejercicio. Mediante un mensaje inicial se presenta el ejercicio y se muestra la fecha actual de ejecución. Ten en cuenta que esta fecha debe calcularse automáticamente y debe reflejar la fecha actual en la que se ejecute el programa (no es un texto fijo). Consultar la clase `LocalDate` en la API de Java te será de utilidad.

A continuación, se debe pedir el nombre de los jugadores que participarán. Se creará un cartón con **15 números** para que cada jugador pueda participar en el **sorteo de hoy**. Utiliza el **constructor más adecuado** para realizar esta tarea.

Una vez creados los cartones, se creará un **bombo para el sorteo de hoy**.

Una vez creados los cartones y el bombo, **comienza el juego, "jugamos para Línea"**. Mientras ninguno de los jugadores cante ¡Línea! repetiremos las siguientes acciones:

Utilizando métodos de la clase `Bombo` simularemos la **extracción de una bola**. El **sistema debe "cantar" la bola** extraída.

Una vez extraída la bola, trataremos de **marcar su número** en el **cartón de cada jugador**.

A continuación, **comprobaremos si alguno de los jugadores puede cantar ¡Línea!**:

Si algún jugador **canta ¡Línea!**, debemos **comprobar si la línea cantada por el jugador es correcta** o no (revisa la API de ambas clases a ver qué método te puede servir para ello):

Si la línea NO es correcta se informa de ello y **se continúa "jugando para Línea"**

Si la línea es correcta, **termina el "juego para Línea"**. **A partir de ahora ningún jugador podrá cantar ¡Línea!**

A continuación, **jugaremos "para Bingo"**. Mientras ninguno de los dos jugadores cante **¡Bingo!** repetiremos las siguientes acciones:

Utilizando métodos de la clase `Bombo` simularemos la **extracción de una bola**. El sistema debe "cantar" la bola extraída.

Una vez extraída la bola, trataremos de **marcar su número en el cartón de cada jugador**.

A continuación, comprobaremos si alguno de los jugadores **puede cantar ¡Bingo!**:

Si algún jugador canta ¡Bingo! debemos **comprobar si el bingo cantado por el jugador es correcto o no**.

Si el bingo no es correcto se informa de ello y se continúa “jugando para Bingo”

Si el bingo es correcto, termina el juego y **se declara como ganador a ese jugador.**

Por último, se mostrará **un resumen del sorteo**.

Ejemplos de ejecución

En este ejercicio los únicos datos de entrada que se deben introducir son los nombres de los dos jugadores. Al igual que en el ejercicio anterior, **cada cartón se generará con números aleatorios**. El **desarrollo** de la partida **variará** en cada ejecución **según las bolas que vayan saliendo** del bombo, así que **no te fijes en los datos exactos** de la salida pero **sí presta atención al desarrollo de la ejecución**.

A continuación, se muestran un par de ejemplos de ejecución. Tu salida, obviando los datos aleatorios **debería ser similar** a la siguiente:

Primera ejecución

Mostrar retroalimentación

```
Ejercicio 2. ¡A jugar!
----------------------
Fecha ACTUAL de ejecución: 06/11/2022

Introduce el nombre del/la primer/a jugador/a:
Carmen

Introduce el nombre del/la segundo/a jugador/a:
Manuel

Creando el cartón de Carmen con 15 números para el sorteo de hoy...
+ Se ha creado correctamente un cartón para Carmen con 15 números!

+--------------------------------+
| N15         BINGOTRASS - Carmen |
+--------------------------------+
|   ··  14  20  ··  36  38  43   |
|   10  15  25  31  ··  ··  44   |
|   11  17  ··  35  ··  39  48   |
+--------------------------------+
|  ID: C-000 - SORTEO:06/11/2022 |
+--------------------------------+

Creando el cartón de Manuel con 15 números para el sorteo de hoy...
+ Se ha creado correctamente un cartón para Manuel con 15 números!

+--------------------------------+
| N15         BINGOTRASS - Manuel |
+--------------------------------+
|   04  ··  ··  25  34  41  46   |
|   05  24  ··  ··  37  44  49   |
|   15  ··  ··  31  40  45  50   |
+--------------------------------+
|  ID: C-001 - SORTEO:06/11/2022 |
+--------------------------------+

Creando el bombo para el sorteo de hoy...
Se ha creado un bombo correctamente (bolas del 1 al 50) para el sorteo de 06/11/2022!

Comienza el juego, jugamos para "Línea"...
```

```
24  -  VEINTICUAAAATRO!!... DOS, CUATRO
30  -  TREIIIINTA!!... TRES, CERO
22  -  VEINTIDOOOOS!!... DOS, DOS
47  -  CUAREEEENTA Y SIETE!!... CUATRO, SIETE
25  -  VEINTICIIIINCO!!... DOS, CINCO
06  -  SEEEEIS!!... CERO, SEIS
19  -  DIECINUEEEEVE!!... UNO, NUEVE
44  -  CUAREEEENTA Y CUATRO!!... CUATRO, CUATRO
41  -  CUAREEEENTA Y UNO!!... CUATRO, UNO
15  -  QUIIIINCE!!... UNO, CINCO
14  -  CATOOOORCE!!... UNO, CUATRO
13  -  TREEEECE!!... UNO, TRES
04  -  CUAAAATRO!!... CERO, CUATRO
43  -  CUAREEEENTA Y TRES!!... CUATRO, TRES
10  -  DIEEEEZ!!... UNO, CERO
23  -  VEINTITREEEES!!... DOS, TRES
26  -  VEINTISEEEEIS!!... DOS, SEIS
31  -  TREIIIINTA Y UNO!!... TRES, UNO

***** Carmen: LINEEEEAAA! *****

Verificamos la línea cantada por Carmen...

+--------------------------------+
| N15         BINGOTRASS - Carmen |
+--------------------------------+
|   ··  XX  20  ··  36  38  XX   |
|   XX  XX  XX  XX  ··  ··  XX   |
|   11  17  ··  35  ··  39  48   |
+--------------------------------+
|  ID: C-000 - SORTEO:06/11/2022 |
+--------------------------------+
La línea cantada por Carmen es VÁLIDA



Seguimos para "Bingo"...
16  -  DIECISEEEEIS!!... UNO, SEIS
28  -  VEINTIOOOOCHO!!... DOS, OCHO
49  -  CUAREEEENTA Y NUEVE!!... CUATRO, NUEVE
40  -  CUAREEEENTA!!... CUATRO, CERO
46  -  CUAREEEENTA Y SEIS!!... CUATRO, SEIS
03  -  TREEEES!!... CERO, TRES
39  -  TREIIIINTA Y NUEVE!!... TRES, NUEVE
12  -  DOOOOCE!!... UNO, DOS
38  -  TREIIIINTA Y OCHO!!... TRES, OCHO
35  -  TREIIIINTA Y CINCO!!... TRES, CINCO
05  -  CIIIINCO!!... CERO, CINCO
01  -  UUUUNO!!... CERO, UNO
02  -  DOOOOS!!... CERO, DOS
27  -  VEINTISIEEEETE!!... DOS, SIETE
34  -  TREIIIINTA Y CUATRO!!... TRES, CUATRO
18  -  DIECIOOOOCHO!!... UNO, OCHO
37  -  TREIIIINTA Y SIETE!!... TRES, SIETE
36  -  TREIIIINTA Y SEIS!!... TRES, SEIS
48  -  CUAREEEENTA Y OCHO!!... CUATRO, OCHO
32  -  TREIIIINTA Y DOS!!... TRES, DOS
42  -  CUAREEEENTA Y DOS!!... CUATRO, DOS
08  -  OOOOCHO!!... CERO, OCHO
09  -  NUEEEEVE!!... CERO, NUEVE
45  -  CUAREEEENTA Y CINCO!!... CUATRO, CINCO
20  -  VEEEEINTE!!... DOS, CERO
33  -  TREIIIINTA Y TRES!!... TRES, TRES
50  -  CIENCUEEEENTA!!... CINCO, CERO

***** Manuel: BIIIINGOOO! *****

Verificamos el bingo cantado por Manuel...
```

```
+---------------------------------+
| N15         BINGOTRASS - Manuel |
+---------------------------------+
|   XX  ··  ··  XX  XX  XX  XX    |
|   XX  XX  ··  ··  XX  XX  XX    |
|   XX  ··  ··  XX  XX  XX  XX    |
+---------------------------------+
|  ID: C-001 - SORTEO:06/11/2022  |
+---------------------------------+
El bingo cantado por Manuel es VÁLIDO
Manuel ha GANADO el juego!!



-----------------------------------
   RESUMEN DEL SORTEO 06/11/2022
-----------------------------------
Bolas NO extraídas hasta el momento:
07 11 17 21 29
-----------------------------------
 Bolas extraídas hasta el momento:
01 02 03 04 05 06 08 09 10 12
13 14 15 16 18 19 20 22 23 24
25 26 27 28 30 31 32 33 34 35
36 37 38 39 40 41 42 43 44 45
46 47 48 49 50
-----------------------------------
- Cantidad de bolas extraídas: 45
- Quedan aún en el bombo: 5
-----------------------------------
- Ganador/a Línea: Carmen
- Ganador/a Bingo: Manuel
-----------------------------------

El sorteo ha finalizado!!
```

Segunda ejecución

Mostrar retroalimentación

```
Ejercicio 2. ¡A jugar!
----------------------
Fecha ACTUAL de ejecución: 06/11/2022

Introduce el nombre del/la primer/a jugador/a:
Patricia

Introduce el nombre del/la segundo/a jugador/a:
Alejandro

Creando el cartón de Patricia con 15 números para el sorteo de hoy...
+ Se ha creado correctamente un cartón para Patricia con 15 números!

+---------------------------------+
| N15       BINGOTRASS - Patricia |
+---------------------------------+
|   02  12  18  27  ··  40  ··    |
|   09  14  22  33  38  41  ··    |
|   10  16  26  ··  ··  45  ··    |
+---------------------------------+
```

```
|  ID: C-000 - SORTEO:06/11/2022  |
+---------------------------------+

Creando el cartón de Alejandro con 15 números para el sorteo de hoy...
+ Se ha creado correctamente un cartón para Alejandro con 15 números!

+---------------------------------+
| N15      BINGOTRASS - Alejandro |
+---------------------------------+
|   01  04  08  18  19  21  35    |
|   ··  ··  15  ··  20  30  42    |
|   ··  06  17  ··  ··  33  48    |
+---------------------------------+
|  ID: C-001 - SORTEO:06/11/2022  |
+---------------------------------+

Creando el bombo para el sorteo de hoy...
Se ha creado un bombo correctamente (bolas del 1 al 50) para el sorteo de 06/11/2022!

Comienza el juego, jugamos para "Línea"...


46  -  CUAREEEENTA Y SEIS!!... CUATRO, SEIS
31  -  TREIIIINTA Y UNO!!... TRES, UNO
40  -  CUAREEEENTA!!... CUATRO, CERO
33  -  TREIIIINTA Y TRES!!... TRES, TRES
42  -  CUAREEEENTA Y DOS!!... CUATRO, DOS
18  -  DIECIOOOOCHO!!... UNO, OCHO
22  -  VEINTIDOOOOS!!... DOS, DOS
44  -  CUAREEEENTA Y CUATRO!!... CUATRO, CUATRO
10  -  DIEEEEZ!!... UNO, CERO
47  -  CUAREEEENTA Y SIETE!!... CUATRO, SIETE
15  -  QUIIIINCE!!... UNO, CINCO
30  -  TREIIIINTA!!... TRES, CERO
26  -  VEINTISEEEEIS!!... DOS, SEIS
43  -  CUAREEEENTA Y TRES!!... CUATRO, TRES
34  -  TREIIIINTA Y CUATRO!!... TRES, CUATRO
01  -  UUUUNO!!... CERO, UNO
50  -  CIENCUEEEENTA!!... CINCO, CERO
14  -  CATOOOORCE!!... UNO, CUATRO
06  -  SEEEEIS!!... CERO, SEIS
12  -  DOOOOCE!!... UNO, DOS
23  -  VEINTITREEEES!!... DOS, TRES
11  -  OOOONCE!!... UNO, UNO
07  -  SIEEEETE!!... CERO, SIETE
38  -  TREIIIINTA Y OCHO!!... TRES, OCHO
37  -  TREIIIINTA Y SIETE!!... TRES, SIETE
17  -  DIECISIEEEETE!!... UNO, SIETE
27  -  VEINTISIEEEETE!!... DOS, SIETE
05  -  CIIIINCO!!... CERO, CINCO
32  -  TREIIIINTA Y DOS!!... TRES, DOS
39  -  TREIIIINTA Y NUEVE!!... TRES, NUEVE
24  -  VEINTICUAAAATRO!!... DOS, CUATRO
16  -  DIECISEEEEIS!!... UNO, SEIS
02  -  DOOOOS!!... CERO, DOS

***** Patricia: LINEEEEAAA! *****

Verificamos la línea cantada por Patricia...

+---------------------------------+
| N15       BINGOTRASS - Patricia |
+---------------------------------+
|   XX  XX  XX  XX  ··  XX  ··    |
|   09  XX  XX  XX  XX  41  ··    |
|   XX  XX  XX  ··  ··  45  ··    |
+---------------------------------+
|  ID: C-000 - SORTEO:06/11/2022  |
+---------------------------------+
La línea cantada por Patricia es VÁLIDA
```

```
Seguimos para "Bingo"...
20  -  VEEEEINTE!!... DOS, CERO
29  -  VEINTINUEEEEVE!!... DOS, NUEVE
48  -  CUAREEEENTA Y OCHO!!... CUATRO, OCHO
08  -  OOOOCHO!!... CERO, OCHO
13  -  TREEEECE!!... UNO, TRES
41  -  CUAREEEENTA Y UNO!!... CUATRO, UNO
03  -  TREEEES!!... CERO, TRES
25  -  VEINTICIIIINCO!!... DOS, CINCO
19  -  DIECINUEEEEVE!!... UNO, NUEVE
49  -  CUAREEEENTA Y NUEVE!!... CUATRO, NUEVE
36  -  TREIIIINTA Y SEIS!!... TRES, SEIS
45  -  CUAREEEENTA Y CINCO!!... CUATRO, CINCO
04  -  CUAAAATRO!!... CERO, CUATRO
35  -  TREIIIINTA Y CINCO!!... TRES, CINCO
28  -  VEINTIOOOOCHO!!... DOS, OCHO
21  -  VEINTIUUUUNO!!... DOS, UNO

***** Alejandro: BIIIINGOOO! *****

Verificamos el bingo cantado por Alejandro...

+--------------------------------+
| N15      BINGOTRASS - Alejandro |
+--------------------------------+
|   XX  XX  XX  XX  XX  XX  XX   |
|   ..  ..  XX  ..  XX  XX  XX   |
|   ..  XX  XX  ..  ..  XX  XX   |
+--------------------------------+
|  ID: C-001 - SORTEO:06/11/2022  |
+--------------------------------+
El bingo cantado por Alejandro es VÁLIDO
Alejandro ha GANADO el juego!!



----------------------------------
   RESUMEN DEL SORTEO 06/11/2022
----------------------------------
Bolas NO extraídas hasta el momento:
09
----------------------------------
 Bolas extraídas hasta el momento:
01 02 03 04 05 06 07 08 10 11
12 13 14 15 16 17 18 19 20 21
22 23 24 25 26 27 28 29 30 31
32 33 34 35 36 37 38 39 40 41
42 43 44 45 46 47 48 49 50
----------------------------------
- Cantidad de bolas extraídas: 49
- Quedan aún en el bombo: 1
----------------------------------
- Ganador/a Línea: Patricia
- Ganador/a Bingo: Alejandro
----------------------------------

El sorteo ha finalizado!!
```

1.3.- Ejercicio 3: ¡vamos al cine!

Cines Trassierra. Fran Jiménez (elaboración propia). (CC BY)

Nuestra empresa, **Cines Trassierra**, ofrece los últimos estrenos y una política de precios muy competitiva. Partiendo de un **precio base de 4,50€ por entrada**, los precios se ven **modificados según el día y hora** de la sesión que se quiera reservar.

Las sesiones **comienzan a las 17:00h** y se realizan **cada 2 horas y media** (2:30h) siendo las siguientes a las **19:30h** y **22:00h**. Después de la sesión de las 22:00h **ya no hay ninguna sesión más**.

Todas las sesiones **parten del mismo precio base (4,50€)** pero la sesión de las **22:00 cuesta 5,50€ (1€ más)** más por ser la más demandada por nuestros espectadores. Además, todos los **lunes** y todos los **jueves** son nuestros ***"Días del espectador"*** cuya principal ventaja es que **el precio a pagar se reduce un 50%**.

Para poder ofrecer estos precios tan competitivos el único requisito es que **la compra de entradas debe realizarse al menos con una semana de antelación**.

Nos solicitan que desarrollemos un programa en Java (Ejercicio3) que permita calcular el importe que debe pagar un espectador según la fecha y hora para la que quiera sacar su entrada.

Para ello, el programa deberá realizar las siguientes acciones:

Lectura desde el teclado de una fecha determinada.

Solicita la introducción de una fecha **en una cadena de texto**. Se recomienda el formato **05/12/2022** (dd/mm/yyyy).

Se intentará crear un objeto de la clase `LocalDate` para esa fecha introducida. La clase `LocalDate` tiene métodos para **convertir cadenas con formato de fecha a objetos de fecha**, consulta su API en la documentación en el enlace que encuentras en el **apartado 8.1** del tema.

Si la creación de la fecha **no ha sido posible** (fecha **errónea**) o **no tiene una antelación suficiente** (solo se aceptan compras con un **mínimo de una semana de antelación**) se indicará mediante un mensaje y se volverá a solicitar al usuario la **introducción de una nueva fecha**.

Se determinará a **qué día de la semana** pertenece esa fecha para saber si la reserva es para alguno de los ***"Días del espectador"***. **Nota:** Para esto será muy útil, de nuevo, consultar la **API de** `LocalDate`**.**

Una vez conocida la fecha, se solicitará al usuario la **introducción de la hora** para la que quiere comprar entrada:

Primero se solicitará **una hora (entre 0 y 23)** y mientras no se introduzca una hora válida (porque no esté en el rango o porque ni siquiera sea un número entero) se volverá a solicitar hasta que se disponga de una hora válida.

A continuación, se solicitará el **minuto (puede ser cualquiera entre 0 y 59)** y se realizará la misma validación que en el caso de la hora.

Creación de los objetos `LocalTime` **de referencia:**

Una vez definida la fecha y hora para la que querría comprar el usuario, **se creará un objeto** de la clase `LocalTime` con la hora de la primera sesión, es decir, las 17:00.

Continuaremos con la creación de un segundo objeto de la clase `LocalTime` con la hora y minuto válidos que ha introducido el usuario mediante el teclado.

> Recuerda que la clase `LocalTime` **no dispone de constructores públicos**, de manera que tendrás que usar los **métodos "fábrica" o pseudoconstructores** que proporcione para crear objetos instancia de esta clase.

A continuación, realizaremos cálculos con esos dos objetos de tipo `LocalTime` para obtener cuál es la siguiente sesión a la que el usuario podrá entrar según la hora que haya indicado. Tendremos los siguientes casos:

Si la hora introducida por el teclado es **anterior a la hora de inicio de la primera sesión**:
Se calculará cuántos **minutos quedarían hasta el comienzo de la primera** sesión.
Se calculará el **precio de la primera sesión** (según la **fecha** introducida).

Si la hora introducida por el teclado es **posterior al comienzo de una sesión**:
Se calculará cuántos **minutos quedarían hasta el comienzo de la siguiente** sesión.
Se calculará el **precio de la siguiente sesión** (según la **fecha y hora** introducidas).

Si la hora introducida por el teclado es **posterior al comienzo de la última sesión**:
Se calculará cuántos **minutos haría que empezó esa sesión respecto a la hora introducida**.
Se indicará que ya **no hay más proyecciones** después.

Mostrar por pantalla los resultados obtenidos según el procesamiento realizado: Lo que obtengas como resultado de las comprobaciones y cálculos anteriores será lo que tendrás que mostrar como mensaje de salida. Puedes almacenarlo en una cadena de caracteres para mostrarlo en el bloque final de salida de resultados.

Orientaciones para el alumnado

IMPORTANTE: Para llevar a cabo los cálculos y comprobaciones del paso 4 solo debes usar esos dos objetos de tipo `LocalTime` (hora de la primera sesión y hora introducida por teclado). A partir de ellos y aprovechando los métodos proporcionados de esta clase `LocalTime` (*isBefore, isAfter, until, plusHours*, etc.). También te será de utilidad el enumerado `ChronoUnit` para indicar que la unidad deseada en los cálculos es "minutos" (`ChronoUnit.MINUTES`). La idea de este ejercicio es que **practiques con el uso de esos métodos, NO que hagas cálculos "a mano" utilizando los valores de horas y minutos independientemente**.

Ejemplos de ejecución

A continuación, te proporcionamos algunos ejemplos de cómo podría quedar la ejecución del programa dependiendo de las posibles entradas que se puedan proporcionar.

Ejemplo 1: Pruebas con fechas y horas válidas e inválidas

Mostrar retroalimentación

```
Ejercicio 3. ¡Vamos al cine!
----------------------------
Introduce la fecha para la que quieres comprar la entrada
---------------------------------------------------------
- Fecha (ej. 06/04/2023): 9-11-2022
¡ERROR! La fecha introducida NO es válida (utiliza el formato sugerido)

- Fecha (ej. 06/04/2023): 9/11/2022
¡ERROR! No se permiten reservas con menos de una semana de antelación

- Fecha (ej. 06/04/2023): 15/02/2023

Introduce la hora para la que quieres comprar la entrada
--------------------------------------------------------
- Hora (00-23): 28
¡ERROR! Hora fuera del rango 0-23
- Hora (00-23): 15
- Minuto (00-59): 92
```

```
¡ERROR! Minuto fuera del rango 0-59
- Minuto (00-59): 55

A la hora indicada faltarían 65 minutos para el inicio de la primera sesión
El precio de una entrada para esta sesión es de 4,50€

El programa ha finalizado!!
```

Como puedes observar, si se introducen números fuera del rango permitido se vuelve a solicitar la información. Por otro lado, si se introduce algo diferente a un número entero, el programa no debería "romperse" debido a una `InputMismatchException`, sino que la excepción debería ser capturada y convenientemente gestionada para a continuación volver a pedir el dato.

Nota: puedes usar System.err.println para mostrar esos mensajes de error.

Ejemplo 2: Día del Espectador y entrada para sesión intermedia (por ejemplo, a partir de las 18:45h)

Mostrar retroalimentación

```
Ejercicio 3. ¡Vamos al cine!
----------------------------
Introduce la fecha para la que quieres comprar la entrada
-----------------------------------------------------------
- Fecha (ej. 06/04/2023): 23/01/2023

Introduce la hora para la que quieres comprar la entrada
-----------------------------------------------------------
- Hora (00-23): 18
- Minuto (00-59): 45

A la hora indicada faltarían 45 minutos para el inicio de la sesión de las 19:30
El precio de una entrada para esta sesión es de 2,25€
Nota: se ha aplicado un descuento del 50.0% por ser Día del Espectador

El programa ha finalizado!!
```

Ejemplo 3: Día normal y entrada para la última sesión (por ejemplo, a partir de las 21:45h)

Mostrar retroalimentación

```
Ejercicio 3. ¡Vamos al cine!
----------------------------
Introduce la fecha para la que quieres comprar la entrada
-----------------------------------------------------------
- Fecha (ej. 06/04/2023): 27/12/2022

Introduce la hora para la que quieres comprar la entrada
-----------------------------------------------------------
- Hora (00-23): 21
- Minuto (00-59): 45

A la hora indicada faltarían 15 minutos para el inicio de la sesión de las 22:00
El precio de una entrada para esta sesión es de 5,50€
```

```
El programa ha finalizado!!
```

Ejemplo 4: Día del espectador y entrada para la última sesión (por ejemplo, a partir de las 20:33h)

Mostrar retroalimentación

```
Ejercicio 3. ¡Vamos al cine!
----------------------------
Introduce la fecha para la que quieres comprar la entrada
---------------------------------------------------------
- Fecha (ej. 06/04/2023): 29/12/2022

Introduce la hora para la que quieres comprar la entrada
--------------------------------------------------------
- Hora (00-23): 20
- Minuto (00-59): 33

A la hora indicada faltarían 87 minutos para el inicio de la sesión de las 22:00
El precio de una entrada para esta sesión es de 2,75€
Nota: se ha aplicado un descuento del 50.0% por ser Día del Espectador

El programa ha finalizado!!
```

Ejemplo 5: Intento de reserva de entrada después del comienzo de la última sesión (por ejemplo, a partir de las 22:30h)

Mostrar retroalimentación

```
Ejercicio 3. ¡Vamos al cine!
----------------------------
Introduce la fecha para la que quieres comprar la entrada
---------------------------------------------------------
- Fecha (ej. 06/04/2023): 25/04/2023

Introduce la hora para la que quieres comprar la entrada
--------------------------------------------------------
- Hora (00-23): 22
- Minuto (00-59): 30

La última sesión de este día comienza 30 minutos ANTES de la hora indicada
No hay más sesiones después de esta.

El programa ha finalizado!!
```

Ejemplo 6: Intento de reserva para la misma hora de comienzo de una sesión (por ejemplo, la sesión de las 22:00h)

Mostrar retroalimentación

```
Ejercicio 3. ¡Vamos al cine!
---------------------------
Introduce la fecha para la que quieres comprar la entrada
---------------------------------------------------------
- Fecha (ej. 06/04/2023): 13/01/2023

Introduce la hora para la que quieres comprar la entrada
---------------------------------------------------------
- Hora (00-23): 22
- Minuto (00-59): 00

A la hora indicada faltarían 0 minutos para el inicio de la sesión de las 22:00
El precio de una entrada para esta sesión es de 5,50€

El programa ha finalizado!!
```

Ejemplo 7: Intento de reserva para una hora posterior al comienzo de la última sesión en Día del Espectador (por ejemplo, las 22:45h)

Mostrar retroalimentación

```
Ejercicio 3. ¡Vamos al cine!
---------------------------
Introduce la fecha para la que quieres comprar la entrada
---------------------------------------------------------
- Fecha (ej. 06/04/2023): 23/03/2023

Introduce la hora para la que quieres comprar la entrada
---------------------------------------------------------
- Hora (00-23): 22
- Minuto (00-59): 45

La última sesión de este día comienza 45 minutos ANTES de la hora indicada
No hay más sesiones después de esta.

El programa ha finalizado!!
```

2.- Información de interés.

Recursos necesarios y recomendaciones

Debes usar OBLIGATORIAMENTE el proyecto base NetBeans que te proporcionamos aquí. Ese proyecto incluye una **biblioteca específica para esta tarea** (`libtarea3`) que contiene las clases `CartonBingo` y `Bombo`. Esas clases no forman parte de la API de Java.

mohamed Hassan (Licencia Pixabay)

Es vital que utilices este proyecto porque si no lo haces no dispondrás de esas clases y no podrás implementar correctamente tus programas al no disponer de esas clases ni de la configuración necesaria de tus bibliotecas.

Además, este proyecto contiene tres programas (clases `Ejercicio1`, `Ejercicio2` y `Ejercicio3`) correspondientes a cada uno a los ejercicios que debes implementar. Cada uno tendrá su propio `main` con el esqueleto del programa y algunas indicaciones de las pautas que debes seguir y así orientarte un poco el trabajo que debes realizar. ¡Aprovéchalos!

Descarga y utiliza este proyecto. ¡NO crees tu propio proyecto en Netbeans! Si lo haces, tu tarea será considerada como NO APTA pues no tendrá las características que necesitamos que tenga.

Descargar proyecto Netbeans para la Tarea Online 03 (zip - 432 KB)

Una vez descargado el proyecto, renómbralo con el estilo de nombre que solemos darle al paquete de entrega de la tarea:

Apellido1_Apellido2_Nombre_PROG_Tarea03

Y ya podrás ponerte a trabajar sobre él. ¡Ánimo!

Aunque en el proyecto base que se te proporciona ya contiene la **documentación JavaDoc** de las clases `CartonBingo` y `Bombo`. Te las proporcionamos también aparte por si quieres disponer de ellas para consultarlas independientemente del IDE Netbeans:

Descargar JavaDoc de las clases `CartonBingo` y `Bombo` para la Tarea Online 03 (zip - 412.18 KB)

Una vez descargado y descomprimido el paquete debes abrir con un navegador web el archivo `index.html` para empezar a navegar por el javadoc de esta API y consultar la documentación de cada clase.

Indicaciones de entrega

Una vez realizada la tarea, el envío se realizará a través de la plataforma. Comprime la carpeta del proyecto NetBeans en un fichero `.zip` y nómbralo siguiendo las siguientes pautas:

Apellido1_Apellido2_Nombre_PROG_Tarea03

3.- Evaluación de la tarea.

Criterios de evaluación implicados

En esta unidad se va a evaluar el resultado de aprendizaje **RA2. "Escribe y prueba programas sencillos, reconociendo y aplicando los fundamentos de la programación orientada a objetos."** y algunos criterios de evaluación correspondientes al resultado de aprendizaje **RA5 "Realiza operaciones de entrada y salida de información, utilizando procedimientos específicos del lenguaje y librerías de clases"**. Ello se hará mediante los siguientes Criterios de Evaluación correspondientes a cada resultado de aprendizaje citado:

Criterios de Evaluación correspondientes al **RA2**:

- RA2.a) Se han identificado los fundamentos de la programación orientada a objetos.
- RA2.b) Se han escrito programas simples.
- RA2.c) Se han instanciado objetos a partir de clases predefinidas.
- RA2.d) Se han utilizado métodos y propiedades de los objetos.
- RA2.e) Se han escrito llamadas a métodos estáticos.
- RA2.f) Se han utilizado parámetros en la llamada a métodos.
- RA2.g) Se han incorporado y utilizado librerías de objetos.
- RA2.h) Se han utilizado constructores.
- RA2.i) Se ha utilizado el entorno integrado de desarrollo en la creación y compilación de programas simples

Peggy_Marco (Pixabay Licens

Criterios de Evaluación correspondientes al **RA5**:

- RA5.a) Se ha utilizado la consola para realizar operaciones de entrada y salida de información.
- RA5.b) Se han aplicado formatos en la visualización de la información.
- RA5.c) Se han reconocido las posibilidades de entrada/salida del lenguaje y las librerías asociadas.

De manera general en esta tarea podremos comprobar que

- **Se declaran** correctamente **variables de referencia**.
- Se crean objetos **LocalDate**.
- Se crean objetos **CartonBingo** y **Bombo** válidos y/o inválidos.
- **Se obtiene información** de objetos CartonBingo y Bombo mediante sus métodos adecuados.
- Se aplican métodos de los distintos objetos: **marcarNumero, sacarBola, cantarLinea, verificarBingo, etc.**.
- Se obtiene información de los objetos mediante el uso del método **toString**.
- Se muestra la **información correcta por pantalla**, utilizando el **formato apropiado**.
- Se obtiene información de clase mediante el uso correcto de **métodos estáticos de la misma**.
- Se crean objetos **CartonBingo** gestionando de forma apropiada de **excepciones** en la invocación a **constructores**.
- **Se crean objetos** LocalTime. Uso de **métodos "*fábrica*"**. Solo es necesario crear dos objetos. **Ninguno más**.
- **Se comprueba la ubicación de una hora** dentro de varios rangos horarios: uso de isBefore, *isAfter, plusHours, plusMinutes*, etc. **usando únicamente los dos objetos LocalTime**
- Se obtiene el **día de la semana de una fecha concreta**. Uso de getDayOfWeek. Uso del enumerado DayOfWeek
- Se obtiene el **tiempo transcurrido entre dos horas**. Uso de until. Uso del enumerado ChronoUnit

¿Cómo valoramos y puntuamos tu tarea?

A continuación mostraremos los elementos que podrán penalizar en la evaluación de la tarea. Tenlos muy en cuenta al resolver los ejercicios:

1. **Los programas deben compilar**. Cualquier funcionalidad pedida en el enunciado debe poderse probar y ha de funcionar correctamente de acuerdo a las especificaciones del enunciado.**Si el programa ni siquiera compila, el ejercicio podría considerarse nulo y su puntuación podría llegar a ser 0.**
2. Se utilizan **estructuras de salto incondicional** para el **control del flujo**, o bien herramientas como return, exit(0) o cualquier otro mecanismo que no sea el **fin de la ejecución natural del programa**.
3. No se respeta la **estructura de bloques** propuesta en la plantilla: **entrada de datos**, **procesamiento** y **salida de resultados** (ni siquiera una versión adaptada a las necesidades del problema)
4. **Algunos identificadores no cumplen con el convenio sobre "asignación de nombres a identificadores"** establecido para el lenguaje Java para constantes, variables, métodos, clases, etc.
5. **Se declaran identificadores que no tienen nombres significativos o descriptivos** que representen de alguna manera la información que están almacenando para que el código quede lo más claro, legible y autodocumentado posible.
6. No se observa una **corrección ortográfica y gramatical**, así como la **coherencia en las expresiones lingüísticas**, tanto en los **comentarios en el código** como en los **textos de los mensajes** que aparezcan en pantalla para pedir información de entrada al usuario o para mostrar resultados de salida. Deben evitarse **mensajes de entrada de datos y/o salida de resultados** inapropiados, descontextualizados, insuficientes o incorrectos.
7. **El código no está correctamente indentado. Es fundamental para poder observar e intuir rápidamente, y de forma visual, la estructura de los programas. Recuerda que NetBeans puede hacer esto por ti (selecciona el código a "*formatear*" o "*embellecer*" y pulsa Alt+Mayús.+F).**

En esta otra tabla puedes observar la puntuación máxima asignada a cada ejercicio de la tarea así como los principales elementos que se tendrán en cuenta a la hora de corregir cada uno de ellos:

Rúbrica de la tarea	
Punto Control 1: Se han identificado los fundamentos de la programación orientada a objetos, y utilizado en la creación de programas simples. • RA2.a) Se han identificado los fundamentos de la programación orientada a objetos. • RA2.b) Se han escrito programas simples.	12,00%
Punto Control 2: Se han instanciado objetos a partir de clases predefinidas, la creación de estos objetos se hace con sentido para la resolución del problema planteado en cada caso, sin la creación excesiva de objetos y controlando las necesidades que solicita cada problema particular • RA2.c) Se han instanciado objetos a partir de clases predefinidas.	4,00%
Punto de Control 3: Se han utilizado métodos y propiedades de los objetos en el desarrollo de la soluciones de los ejercicios propuestos, no solo su definición si no también su utilización coherente. • RA2.d) Se han utilizado métodos y propiedades de los objetos.	4,00%
Punto de Control 4: Se han escrito llamadas a métodos estáticos, entendiendo su uso como métodos de clase y no de objeto, y facilitando de esta forma la resolución de cada problema concreto.	4,00%

• RA2.e) Se han escrito llamadas a métodos estáticos.	
Punto de Control 5: Se han utilizado parámetros en la llamada a métodos. La llamada a los método tanto de librería de clases de Java, como implementadas por el alumno/a incorporarán los parámetros correctamente definidos y verificados antes de su uso, eligiendo en cada caso la llamada con los parámetros que más se ajusten a las necesidades de cada problema. • RA2.f) Se han utilizado parámetros en la llamada a métodos.	4,00%
Punto de Control 6: Se han importado librerías de la API de Java descritas en la unidad que facilitan resolución de los problemas mediante la utilización de objetos y métodos conociendo su alcance y uso correcto, eligiendo el método o métodos de las librerías más apropiados. • RA2.g) Se han incorporado y utilizado librerías de objetos.	4,00%
Punto de Control 7: En la resolución de los problemas propuestos se han utilizado constructores más apropiados en cada caso particular, bien creados por el alumno/a, o bien haciendo uso de los ya definidos en las librerías de API de Java que se ha visto en la unidad. • RA2.h) Se han utilizado constructores.	4,00%
Punto de Control 8: Para la resolución de la tarea se utiliza un entorno integrado de desarrollo (IDE) en la creación y compilación de programas simples, en este caso NetBeans, presentando un proyecto agrupando la resolución de la tarea en archivos independientes dentro del proyecto. • RA2.i) Se ha utilizado el entorno integrado de desarrollo en la creación y compilación de programas simples	4,00%
Punto de Control 9: La entrada salida de datos se realiza mediante la consola del IDE utilizando las clases necesarias y salvando los posibles errores que pudiesen generar de su uso en los diferentes problemas planteados. • RA5.a) Se ha utilizado la consola para realizar operaciones de entrada y salida de información. • RA5.c) Se han reconocido las posibilidades de entrada/salida del lenguaje y las librerías asociadas.	6,67%
Punto de Control 10: Los resultados y la salida de datos se presentan acorde con los formatos de visualización de la información requeridos en cada problema.	3,33%

• RA5.b) Se han aplicado formatos en la visualización de la información.	
Punto de Control 11: Los ejercicios desarrollados funcionan correctamente de acuerdo a las especificaciones y limitaciones requeridas en la tarea • RA2.a) Se han identificado los fundamentos de la programación orientada a objetos. • RA2.b) Se han escrito programas simples. • RA2.c) Se han instanciado objetos a partir de clases predefinidas. • RA2.d) Se han utilizado métodos y propiedades de los objetos. • RA2.e) Se han escrito llamadas a métodos estáticos. • RA2.f) Se han utilizado parámetros en la llamada a métodos. • RA2.g) Se han incorporado y utilizado librerías de objetos. • RA2.h) Se han utilizado constructores. • RA2.i) Se ha utilizado el entorno integrado de desarrollo en la creación y compilación de programas simples • RA5.a) Se ha utilizado la consola para realizar operaciones de entrada y salida de información. • RA5.b) Se han aplicado formatos en la visualización de la información. • RA5.c) Se han reconocido las posibilidades de entrada/salida del lenguaje y las librerías asociadas.	50,00%