Autoevaluación UT4

1. Una pastelería nos ha pedido realizar un programa que haga presupuestos de tartas. El programa preguntará primero de qué sabor quiere el usuario la tarta: manzana, fresa o chocolate. La tarta de manzana vale 18 euros y la de fresa 16. En caso de seleccionar la tarta de chocolate, el programa debe preguntar además si el chocolate es negro o blanco; la primera opción vale 14 euros y la segunda 15. Por último se pregunta si se añade nata y si se personaliza con un nombre; la nata suma 2.50 y la escritura del nombre 2.75.
2. Implementa el juego piedra, papel y tijera. Primero, el usuario 1 introduce su jugada y luego el usuario 2. Si alguno de los usuarios introduce una opción incorrecta, el programa deberá mostrar un mensaje de error Realiza un programa que sume los 100 números siguientes a un número entero y positivo introducido por teclado. Se debe comprobar que el dato introducido es correcto (que es un número positivo)
3. Escribe un programa que permita ir introduciendo una serie indeterminada de números mientras su suma no supere el valor 10000. Cuando esto último ocurra, se debe mostrar el total acumulado, el contador de los números introducidos y la media
4. Escribe un programa que lea un número n e imprima una pirámide de números con n filas como en la siguiente figura:

```
1
121
12321
1234321
```

5. Escribe un programa que pida dos números por teclado y que luego mezcle en dos números diferentes los dígitos pares y los impares. Se van comprobando los dígitos de la siguiente manera: primer dígito del primer número, primer dígito del segundo número, segundo dígito del primer número, segundo dígito del segundo número, tercer dígito del primer número… Para facilitar el ejercicio, podemos suponer que el usuario introducirá dos números de la misma longitud y que siempre habrá al menos un dígito par y uno impar. Usa long en lugar de int donde sea necesario para admitir números largos.

6. Realiza un programa que pinte un reloj de arena relleno hecho de asteriscos. El programa debe pedir la altura. Se debe comprobar que la altura sea un número impar mayor o igual a 3, en caso contrario se debe mostrar un mensaje de error. Ejemplo: Por favor, introduzca la altura del reloj de arena: 5

```
*****
 ***
  *
 ***
*****
```

7. Escribe un programa que permita partir un número introducido por teclado en dos partes. Las posiciones se cuentan de izquierda a derecha empezando por el 1. Suponemos que el usuario introduce correctamente los datos, es decir, el número introducido tiene dos dígitos como mínimo y la posición en la que se parte el número está entre 2 y la longitud del número. No se permite en este ejercicio el uso de funciones de manejo de String (por ej. para extraer subcadenas dentro de una cadena).

8. Realiza un programa que calcule el máximo, el mínimo y la media de una serie de números enteros positivos introducidos por teclado. El programa terminará cuando el usuario introduzca un número primo. Este último número no se tendrá en cuenta en los cálculos. El programa debe indicar también cuántos números ha introducido el usuario (sin contar el primo que sirve para salir).

9. Realiza un programa que sea capaz de desplazar todos los dígitos de un número de derecha a izquierda una posición. El dígito de más a la izquierda, pasaría a dar la vuelta y se colocaría a la derecha. Si el número tiene un solo dígito, se queda igual. Por ejemplo, si el número es 1234 el resultado debe ser 2341.

10. Según cierta cultura oriental, los dígitos de la suerte son el 3, el 7, el 8 y el 9. Los dígitos de la mala suerte son el resto: el 0, el 1, el 2, el 4, el 5 y el 6. Un número es afortunado si contiene más dígitos de la suerte que de la mala suerte. Realiza un programa que diga si un número introducido por el usuario es afortunado o no.

11. Escribe un programa que pinte por pantalla un rectángulo hueco de 6 caracteres de ancho por 3 de alto y, a continuación, un menú que permita agrandarlo, reducirlo o cambiar su orientación. Cada vez que el rectángulo se agranda se incrementa en 1 tanto su anchura como su altura. Cuando se reduce se decrementa en 1 su anchura y altura. Por último, cuando se cambia la orientación, los valores de anchura y altura se intercambian. El valor mínimo de la altura o la anchura es 2. Ejemplo:

```
*******
*     *
*******
```

1. Agrandarlo
2. Achicarlo
3. Cambiar la orientación
4. Salir

12. Realiza un simulador de máquina tragaperras que responda a los siguientes requisitos:

    a) El ordenador mostrará una tirada que consiste en mostrar 3 figuras. Hay 5 figuras posibles: corazón, diamante, herradura, campana y limón.
    b) Si las tres figuras son diferentes se debe mostrar el mensaje “Lo siento, ha perdido”.
    c) Si hay dos figuras iguales y una diferente se debe mostrar el mensaje “Bien, ha recuperado su moneda”.
    d) Si las tres figuras son iguales se debe mostrar “Enhorabuena, ha ganado 10 monedas”.
    e) El jugador tendrá un saldo inicial de n monedas y la partida terminará cuando el jugador pierda todo su dinero.

13. Implementa un programa que pida al usuario un valor entero A utilizando un nextInt() y luego muestre por pantalla el mensaje “Valor introducido: …”. Se deberá tratar la excepción InputMismatchException que lanza nextInt() cuando no se introduce un entero válido. En tal caso se mostrará el mensaje “Valor introducido incorrecto”.

14. Implementa un programa que pida dos valores int A y B utilizando un nextInt() (de Scanner), calcule A/B y muestre el resultado por pantalla. Se deberán tratar de forma independiente las dos posibles excepciones, InputMismatchException y ArithmeticException, mostrando en cada caso un mensaje de error diferente en cada caso.

15. Realiza un programa en el que try y catch se encuentra en el main pero el error de la captura de datos se encuentra en un método. Queremos ver que el error se deriva al main.

16. Escribe un programa que pida 10 números con parte decimal. Si se insertan letras pedirá un nuevo número. Visualizar el número más alto introducido Cada vez que se produzca el error Exception, ver el mensaje correspondiente. Además debemos indicar cuantas veces se ha producido el error.

17. Implementa un programa con tres funciones: ◦ void imprimePositivo(int p): Imprime el valor p. Lanza una ‘Exception’ si p < 0. void imprimeNegativo(int n): Imprime el valor n. Lanza una ‘Exception’ si p >= 0. La función main para realizar pruebas. Puedes llamar a ambas funciones varias veces con distintos valores, hacer un bucle para pedir valores por teclado y pasarlos a las funciones, etc. Maneja las posibles excepciones.

18. Implementa una clase Gato con los atributos nombre y edad, un constructor con parámetros, los getters y setters, además de un método imprimir() para mostrar los datos de un gato. El nombre de un gato debe tener al menos 3 caracteres y la edad no puede ser negativa. Por ello, tanto en el constructor como en los setters, deberás comprobar que los valores sean válidos y lanzar una ‘Exception’ si no lo son. Luego, haz una clase principal con main para hacer pruebas: instancia varios objetos Gato y utiliza sus setters, probando distintos valores (algunos válidos y otros incorrectos). Maneja las excepciones.

19. La serie siguiente permite calcular el valor de la función trigonométrica seno de un ángulo x.

$$\sum_{i=0}^{\infty} \frac{(-1)^i}{(2i+1)!} x^{2i+1} = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \cdots = \sin x$$

Desarrolle un proyecto en lenguaje Java con **eclipse** (o un IDE similar) que incluya una clase Main con método main y una clase denominada Seno que responda a las siguientes especificaciones:

a) Los objetos de la clase Seno contarán con dos atributos: “**angulo**” de tipo double y “**numero**” de tipo entero.
b) La clase tendrá un constructor sin argumentos que asignará el valor cero a ambos atributos.

c) La clase tendrá un método **factorial** que permitirá calcular el factorial de un número.

d) La clase tendrá un método **calcularSeno** que recibirá dos argumentos: el ángulo y el número de términos. El método calculará y devolverá el valor del seno del ángulo de acuerdo con la serie para el número de términos especificado.

e) Realice la depuración del proyecto anterior para el escenario siguiente: ángulo = 3.14159 y número de términos = 25.

f) ¿En qué líneas ha ubicado el/los puntos de interrupción? (Indíquelo con un comentario sobre el código desarrollado).

g) Complete la tabla siguiente a partir de los valores de la variable i indicados:

Variable i	Valor de la serie
3	
12	
18	
24	

20. Repita el ejercicio anterior con esta otra serie:

$$\sum_{i=0}^{\infty} \frac{(-1)^i}{(2i)!} x^{2i} = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \cdots = \cos x$$