

|   |
|---|
| <p style="text-align: center;"><b>EXAMEN 1ª EVALUACION. Tipo A</b><br/><b>1. Introducción a los Ordenadores, Internet y Java.</b><br/><b>2. Conceptos Básicos de Programación.</b><br/><b>3. Estructuras de control condicionales y bucles.</b></p> |
|---|

Nombre: \_\_\_\_\_

1. Realiza un pequeño programa, usando *Geany* denominado *Triángulo1.java* que cumpla los siguientes requisitos:

- Debemos tener en cuenta que la suma de los tres ángulos que forman un triángulo son siempre 180 grados
- Usando la generación de números enteros aleatorios siguiente (que produce números comprendidos entre 1 y 180) genera dos variables de tipo entera denominadas *angulo1* y *angulo2*, que corresponde a dos posibles ángulos de un triángulo:

```
new java.util.Random().nextInt(180) + 1
```

- Calcula el ángulo restante y almacénalo en una variable suponiendo su tipo y llámala *angulo3*.
- Puede ocurrir que la suma de los ángulos 1 y ángulos 2 (obtenidos aleatoriamente) superen o igualen los 180 grados, por lo tanto no se puede formar un triángulo, en ese caso mostraremos en consola un mensaje indicando esta situación, que puede ser en el caso que hayamos obtenido aleatoriamente los valores 95 y 85:

Con los ángulos 95 y 85 no se puede formar un triángulo

- En caso contrario mostramos el mensaje con el valor de los tres ángulos, los dos obtenidos aleatoriamente y el calculado por el programa, ejemplo de salida:

Ángulo 1: 100, ángulo 2: 70 y ángulo 3: 10

- En el caso anterior los valores 100 y 70 son obtenidos aleatoriamente y el valor de 10 es calculado por el programa.
- Usa un *printf* para formatear la salida.
- El programa debe ser ejecutable.

2. Usando *Geany* crea un programa denominado *Triangulo2.java* que es una versión del anterior, en este programa vamos a usar la clase *Scanner* para solicitar los valores de los ángulos 1 y 2, en vez de que el programa los obtenga de forma aleatoria. Y todas las variables a utilizar son de tipo *float*.

El uso del *Scanner* se basa en:

- Importar el *Scanner* con la sentencia:

```
import java.util.Scanner;
```

- Generar la referencia del *Scanner*:

```
Scanner sc = new Scanner(System.in);
```

- Leer los datos de los ángulos que para valores *float* sería:

```
sc.nextFloat()
```

- El programa solicita el ángulo 1 y posteriormente el ángulo 2
- Si la suma de ambos es superior a o igual a 180, se deben volver a solicitar los datos, hasta que se pueda formar un triángulo.
- En el caso de que todo sea correcto, el programa continuará y calculará el tercer ángulo y mostraremos los tres ángulos por pantalla, usando dos decimales:

Ángulo 1: 100,90, ángulo 2: 68,11 y ángulo 3: 10.99

- Usamos un *printf* para este menester.
- El programa debe ser ejecutable.

3. Usando *Geany* crea una clase llamada *Numeros.java* teniendo en cuenta lo siguiente:

- Crearemos los siguientes métodos estáticos:
  1. Un método llamado *devolverMedia* que recibe tres números enteros y devuelve el valor medio como *float*
  2. Un método llamado *devolverMayorValor* que recibe tres números enteros y devuelve el valor mas grande de los tres.
  3. Un método llamado *devolverMenorValor* que recibe tres números enteros y devuelve el valor más pequeño de los tres.

- Los números enteros se pueden generar de forma aleatoria (puedes usar el ejemplo de la pregunta 1) o utilizar un *Scanner* como en la cuestión 2.

- Posteriormente se desplegará un menú con las siguientes opciones:

1. Calcular el valor medio.
2. Obtener el mayor valor.
3. Obtener el menor valor.
4. Salir

INTRODUCE OPCIÓN:

Solo se admiten, esos valores, cualquier otro valor va a hacer que se repita el menú. El menú se va a estar mostrando hasta que se pulse 4, que mostrará en consola *FIN DE PROGRAMA* y el programa termina.

- Usando la sentencia *switch* se recogerá la opción mostrada, llamará al método correspondiente de la clase anterior y se mostrará por consola el resultado. Ejemplo con los valores 1, 3 y 5:

(Pulsando 1) Valor medio: 4.50

(Pulsando 2) Mayor valor: 5

(Pulsando 3) Menor valor: 1

(Pulsando 4) FIN DE PROGRAMA (y finaliza el programa)

- Crea un comentario de bloque al inicio de la clase que explique que es lo que hace el programa.
  - Crea un comentario de línea encima de cada método que explique que es lo que hace. Si te hace falta más de una línea, usa por cada una de ellas un comentario de línea, nunca un comentario de bloque.
  - El programa es ejecutable.
4. Usando *Geany* crea un programa para calcular el tiempo de caída libre y la velocidad con que un objeto llega al suelo

Las fórmulas de caída libre son:

$$t = \sqrt{2h/g}$$

$$v = \sqrt{2hg}$$

Dónde:

- $t$  es el tiempo que tarda el cuerpo en caer.
- $h$  es la altura que cae.
- $g$  es la constante de la gravedad y vale *9.8*.
- $v$  es la velocidad con la que llega al suelo.
- Solicita por *Scanner* el valor de la altura con al que cae un cuerpo, dicha variable será entera.
- Calcula el tiempo usando una variable de tipo entero, en este caso de tipo *short* y usando el correspondiente *casting*
- Calcula la velocidad usando una variable de tipo entero, en este caso de tipo *long* y usando el correspondiente *casting*
- Muestra los resultados en consola usando un *printf*, ejemplo:  
 Un cuerpo que cae de desde 22 metros tarda 2 segundos en caer y llega al suelo con una velocidad de 20 m/s
- Crea un comentario de bloque al inicio de la clase que explique que es lo que hace el programa.
- El programa es ejecutable.

|  |
|--|
| <b>CALIFICACIÓN DE RESULTADOS DE APRENDIZAJE Y CRITERIOS DE EVALUACIÓN</b> |
|--|

**(RA1)**

Reconoce la estructura de un programa informático, identificando y relacionando los elementos propios del lenguaje de programación utilizado.

- a) Se han identificado los bloques que componen la estructura de un programa informático. (9)
- b) Se han creado proyectos de desarrollo de aplicaciones. (9)
- c) Se han utilizado entornos integrados de desarrollo. (9)
- d) Se han identificado los distintos tipos de variables y la utilidad específica de cada uno. (9)
- e) Se ha modificado el código de un programa para crear y utilizar variables. (9)
- f) Se han creado y utilizado constantes y literales. (9)
- g) Se han clasificado, reconocido y utilizado en expresiones los operadores del lenguaje. (9)
- h) Se ha comprobado el funcionamiento de las conversiones de tipo explícitas e implícitas. (9)
- i) Se han introducido comentarios en el código. (8)

**(RA2)**

- e) Se han escrito llamadas a métodos estáticos. (12)
- f) Se han utilizado parámetros en la llamada a métodos. (12)
- i) Se ha utilizado el entorno integrado de desarrollo en la creación y compilación de programas simples. (12)

**(RA3)**

- a) Se ha escrito y probado código que haga uso de estructuras de selección. (15)
- b) Se han utilizado estructuras de repetición. (15)
- c) Se han reconocido las posibilidades de las sentencias de salto. (14)
- e) Se han creado programas ejecutables utilizando diferentes estructuras de control. (15)
- f) Se han probado y depurado los programas. (14)

|                                    |
|------------------------------------|
| <b>CALIFICACIONES PARCIALES RA</b> |
|------------------------------------|

**RA1** (80/80)

**RA2** (36/110)

**RA3** (73/130)

|                          |
|--------------------------|
| <b>ENTREGA DE EXAMEN</b> |
|--------------------------|

Se entrega el documento de papel con tu nombre, y un fichero comprimido con los códigos fuentes de los tres ejercicios con el formato APELLIDOS\_NOMBRE.zip o APELLIDOS\_NOMBRE.rar, subiendo éste a la plataforma mas una copia de seguridad que se entrega al profesor.