



- Tarea UT03 - Utilización de Objetos

Enunciado

En esta unidad has visto las características fundamentales de la programación orientada a objetos y estudiado los conceptos más importantes de este modelo de programación. Has aprendido a crear y manipular objetos, qué son los métodos y cómo se definen los parámetros. También has podido conocer cómo se estructura la Biblioteca de Clases de Java, viendo algunas clases importantes, como por ejemplo las que nos permiten realizar la programación de la consola.

Ejercicio 1

1. Crea un proyecto Java denominado Prog03_Ejerc1.
2. Dentro del proyecto, crea un paquete denominado com.prog03.figuras.
3. Dentro de dicho paquete, crea una clase denominada **Triangulo** que:
 - Declare atributos para la base y la altura de un triángulo.
 - Declare atributos para los ángulos: anguloIzq y anguloDer.
 - Declare un constructor vacío que inicialice los atributos a 0.
 - Declara un constructor que inicialice base, altura, anguloIzq y anguloDer.
 - Declare los siguientes métodos:
 - Métodos para actualizar y obtener el valor de cada atributo.
 - float getArea(): devuelve el área del triángulo.
 - float getAnguloSup(): devuelve el ángulo superior.
 - String toString(): devuelve una cadena conteniendo su área (con dos números decimales), su altura y el ángulo superior.
 - boolean isEquilatero(): devuelve un booleano indicando si el triángulo es equilátero.

4. Crea otro paquete con el nombre com.prog03.Principal.
5. Dentro de dicho paquete crea una clase denominada principal que contenga el método main.
6. En el método main, instancia al menos dos objetos de la clase **Triangulo** y comprueba su funcionamiento.

Ejercicio 2

- Construye un nuevo proyecto Java denominado Prog03_Ejerc2. En el proyecto debe aparecer un paquete, que no puede ser el paquete por defecto, ponle el nombre que creas oportuno. Dentro de dicho paquete:
- Crea una clase denominada Libro. Esta clase no debe contener método main.
- Declara en el fichero de la clase un tipo enumerado, denominado enumGenero, para los distintos Géneros de libros: (FICCIÓN, CIENCIA_FICCIÓN, FANTASÍA, HISTORIA, ROMÁNTICA, NOVELA_NEGRA)
- La clase debe contener atributos para el título, el autor (como String), el género (del tipo enumerado declarado) y la fecha de publicación en formato DATE.
- Implementa un constructor que inicialice el título y el género a los valores recibidos por parámetro y los demás atributos a 0 o cadena vacía en caso de que sea un String "". Observa su cabecera en el siguiente código:

Unset

```
Libro(String titulo, enumGenero genero) {
```

- Declara otro constructor que inicialice todos los atributos de la clase. Su cabecera podría ser la siguiente:

Unset

```
Libro(String titulo, String autor, enumGenero genero, Date  
fechaPublicacion) {
```

- Implementa los métodos que permitan acceder y modificar cada uno de los atributos de la clase. Los nombres de dichos métodos serán: getXXX () para obtener el valor del atributo XXX y setXXX (v) para actualizar el atributo XXX con el valor v. Puedes observar la cabecera de algún método en el siguiente código:

Unset

```
public String getTitulo() {  
  
}  
public void setTitulo(String titulo) {  
  
}
```

- Implementa un método que devuelva **true** si el libro es considerado un clásico (consideramos un clásico a cualquier libro anterior al 2000) y **false** en caso contrario. Observa su cabecera en el siguiente código:

Unset

```
public boolean isClasico(){
```

No es necesario utilizar estructuras condicionales pues aún no las hemos trabajado. Se puede implementar este método utilizando operadores de comparación.

- Implementa un método que devuelva una cadena con la fecha en formato largo, por ejemplo, Título: Dune, Autor: Frank Herbert, Género: CIENCIA_FICCIÓN, Fecha de publicación: 01/08/1965. Observa su cabecera:

Unset

```
public String toString (){
```

Ya tenemos nuestra clase **Libro** implementada. Ahora vamos a probar su funcionalidad desde otra clase, denominada Principal, que contendrá el método **main**. Esta clase la debes crear en el mismo paquete que la clase Libro. Dentro de dicha clase:

- Instancia un objeto de la clase **Libro** denominado **objLibro1** con el primer constructor.
- Actualiza los atributos autor y fechaPublicacion para dicho objeto.
- Muestra el Libro por pantalla en formato largo.
- Muestra un mensaje por pantalla indicando si el libro es un clásico (puedes utilizar el operador de comparación ternario. Punto 5.3 de la Unidad 2).
- Instancia otro objeto de la clase **Libro** denominado **objLibro2** con el segundo constructor.
- Muestra un mensaje por pantalla indicando si el libro es un clásico.

- Muestra el Libro por pantalla en formato largo.

Observa un ejemplo de ejecución:

Unset

Primer libro, inicializado con el primer constructor

Título: Nacidos de la Bruma: El Imperio Final, Autor:
Brandon Sanderson, Género: FANTASÍA, Fecha de publicación:
17/08/2006

No es un clásico

Segundo libro, inicializado con el segundo constructor

Título: Dune, Autor: Frank Herbert, Género: CIENCIA_FICCIÓN,
Fecha de publicación: 01/08/1965

Es un clásico

BUILD SUCCESSFUL (total time: 0 seconds)

Criterios de puntuación. Total 10 puntos.

- Ejercicio 1: 6 puntos.
- Ejercicio 2: 4 puntos.
- Total: 10 puntos.

Recursos necesarios para realizar la Tarea.

- Ordenador personal.
- JDK y JRE 21 o superior.
- NetBeans IDE 19 o superior.

Consejos y recomendaciones.

Para una mejor organización de la tarea, crea una carpeta para cada ejercicio llamada EjercicioXX, donde la XX es el número del ejercicio. Sitúa dentro de cada carpeta los ficheros de cada proyecto y cuando todos los ejercicios estén completos comprime esas carpetas en un sólo archivo para subirlo a la plataforma.

Añade al código fuente los comentarios que consideres oportuno.

Indicaciones de entrega

Cada ejercicio estará contenido en un fichero cuyo nombre sea similar a **PROG03_EjerXX**, donde la XX se sustituirá por el número del ejercicio. Se deben entregar cada uno de los proyectos Netbeans creados. Para empaquetar un proyecto en Netbeans, utiliza la opción **File - Export Project** de Netbeans: generarás un fichero .zip con el contenido completo del proyecto.

Una vez realizada la tarea elaborarás un único fichero comprimido (zip o rar) que contenga los dos ficheros correspondientes a los proyectos. El envío se realizará a través de la plataforma de la forma establecida para ello, y el archivo se nombrará siguiendo las siguientes pautas:

Asegúrate de que los archivos fuente (.java), contengan, como comentarios Java, tu nombre y apellidos y el número de ejercicio.

Los archivos fuente deben compilar y ejecutarse siguiendo las instrucciones de su enunciado. De no hacerlo, no se considerarán válidos y serán evaluados con un 0.

En el supuesto que tengas/quieras adjuntar un documento de texto con explicaciones adicionales, o con la solución a alguno de los puntos pedidos, debes de seguir las siguientes consideraciones que se explican en el siguiente [documento](#).

Crea una carpeta para la entrega y adjunta los archivos creados, y renombra la carpeta siguiendo las siguientes pautas:

apellido1_apellido2_nombre_PROG_Tarea03

Asegúrate que el nombre no contenga la letra ñ, tildes ni caracteres especiales extraños. Así por ejemplo la alumna Begoña Sánchez Mañas debería nombrar esta tarea como...

sanchez_manas_begona_PROG_Tarea03

IMPORTANTE

Si la carpeta es muy pesada para el límite máximo de subida de la plataforma Moodle, súbela a tu unidad del Drive. Utiliza la cuenta de Educación, y en la entrega en Moodle adjunta la URL de tu entrega.

Para que la entrega se considere realizada en tiempo y forma, sube al Moodle un documento con la URL que contenga la solución de tu tarea. Deberá contener los archivos y documentos solicitados. Comparte la carpeta mediante su enlace correspondiente al Drive.

No olvides ajustar los privilegios de la carpeta en Drive para que pueda acceder a tu trabajo.

Todos los ejercicios y/o /documentos demandados que no cumplan exactamente con las instrucciones y su formato de entrega no se considerarán válidos y serán evaluados con un 0.