



## Tarea No Presencial de la Sesión 01

### A. Proyecto personSession1Task

- Abre el proyecto personSession1 realizado en clase y crea uno nuevo (Guardar Como) con el nombre **personSession1Task**.
- Realiza las siguientes modificaciones en la clase **Person** del proyecto **personSession1Task**:
  1. Añade las siguientes propiedades o atributos privados:  
`String surname`  
`boolean gender`
- Añade los métodos de modificación y acceso (**get** y **set**) necesarios para dichas propiedades.
- Una vez introducidas las modificaciones, compila la clase y ejecuta los métodos necesarios para comprobar que los resultados son los esperados.

### B. Proyecto houseSession1Task

- Abre el proyecto **house**, y crea una copia del mismo con nombre **houseSession1Task** (opción Guardar Como). *Otro modo de crear la copia es desde el explorador creando una copia de la carpeta completa del proyecto y cambiando su nombre.*
- Crea un objeto de la clase **Picture** e invoca al método **draw** (*se habrá dibujado una casa y un sol*).
- Se pretende modificar el código para que cuando se invoque el método **draw**, además de la casa y el sol, se pinte una persona a la izquierda de la casa. Para hacer esto, deberás añadir el código necesario a la clase **Picture**. Accede al código de la clase **Picture** con la opción abrir editor; centra tu atención en las declaraciones de campos o atributos que están en la parte superior de esta clase. Encontrarás este código

```
private Square wall;  
private Square window;  
private Triangle roof;  
private Circle sun;
```
- Aquí es donde se debe añadir una línea con la declaración de un objeto de la clase **Person** (**¡Ojo!**, ten en cuenta que esta clase **Person** no tiene nada que ver con la clase **Person** que has hecho en clase en el proyecto personSession1. Se trata de la clase **Person** que aparece en este proyecto **houseTareaSession1**).
- Analiza el código del método **draw**, te darás cuenta de que se crea un objeto, por ejemplo, **wall**, y a continuación se modifican sus propiedades con ciertos valores (estado del objeto),

y finalmente se hace visible. Fíjate en cómo se llama a los métodos (nombre del objeto punto nombre del método).

- Añade ahora en el método **draw** el código necesario para crear a una persona con un estado concreto. Crea la persona, muévela a la posición en pantalla a la izquierda de la casa y hazla visible. Prueba primero a crearla y hacerla visible, posteriormente haz que se coloque a la izquierda de la casa antes de hacerla visible.
- Comprueba que funciona correctamente. Si se crea un objeto de la clase **Picture** y se llama al método **draw** dibujará una casa, un sol y una persona situada a la izquierda.

### Desafío 1

- Añade otra operación al dibujo: la puesta de sol. Crea en la clase **Picture** un nuevo método denominado **sunset**. Deberás para ello colocarte al final del código, después del método **setColor**. Asegúrate de que lo incluyes ANTES de la última llave de la clase, que indica finalización de la clase.
- Para llevar a cabo esta operación deberán colocarse las instrucciones que permiten descender al sol en el método **sunset**. Recuerda que el círculo tiene un método **slowMoveVertical** y puedes usarlo para lograr que el sol descienda.
- De este modo al invocar al método **draw()** se verá el sol en lo alto de la imagen pero si se invoca al método **sunset** (un método independiente) hará que el sol descienda hasta desaparecer.

### Desafío 2

- Añade a la puesta de sol anterior la aparición de una **luna de color azul**, de mucho menor tamaño que el sol. Haz que esta luna aparezca desde abajo y suba a una posición en el cielo después de que el sol desaparezca.

### Desafío 3

Accede a la página <https://codingbat.com/java>. Entra en el apartado

“Warmup-1”.

- Trata de resolver dos de los problemas indicados a continuación, dependiendo de tu número de identificador UO. Resuelve los 2 problemas cuyo número resulte de realizar las siguientes operaciones con el valor numérico de tu identificador personal de Uniovi UO:

1.  $(\text{Número de UO} \% 18) + 1$  (siendo % el resto de la división entre 18)
2.  $(\text{Número de UO} + 9) \% 18 + 1$

Ejemplo: para el alumno UO278249, deberá hacer los ejercicios

$(278249 \% 18) + 1 = 6$  (problema número 6. makes10)

$(278249 + 9) \% 18 + 1 = 15$  (problema número 15. Close10)

- |                  |                |              |
|------------------|----------------|--------------|
| 1. sleepIn       | 6. makes10     | 11. in1020   |
| 2. monkeyTrouble | 7. nearHundred | 12. hasTeen  |
| 3. sumDouble     | 8. posNeg      | 13. loneTeen |
| 4. diff21        | 9. or35        | 14. intMax   |
| 5. parrotTrouble | 10. icyHot     | 15. close10  |

16. in3050

17. max1020

18. lastDigit

1. Verifica tu solución ejecutándola y comprobando que pasa todas las pruebas. En caso de no ser capaz solucionarlo, estudia la solución que se presenta al pulsar “*Show solution*”.
2. Modifica tu código para que no pase la primera prueba (las otras podrán o no fallar).
3. Modifica tu código para que no pase la segunda prueba (las otras podrán o no fallar).
4. Recorta (con la herramienta recortes de windows) una imagen del código y la ejecución que resulta al pulsar el botón *Go*, en los tres casos anteriores y pégala en un fichero de texto denominado “*Desafio3.doc*”.
5. Intenta, como ampliación, resolver alguno de los otros problemas propuestos en la página. Para estos no es necesario que entregues nada.

### Entrega de Tarea No Presencial de la Sesión 01

Incluye en un fichero comprimido denominado “*ApellidosYNombre-SxLx.zip*” los proyectos: *person.Session1Task*, *house.Session1Task* y *Desafio3.pdf*. Ejemplo: AlonsoAlvarezPablo-S1L4.zip (Siendo S1 sesión 1 y L4 el laboratorio 4, en tu caso el que te corresponda).

Entrega este fichero comprimido en el campus virtual en el apartado “Actividades para entregar de prácticas de laboratorio” **antes de las 9:00 del día anterior a tu próxima sesión de prácticas.**