

Conceptos fundamentales de Java 3-5:

Selección aleatoria y constructores

Actividades prácticas

Objetivos:

- Crear comportamientos asignados de forma aleatoria
- Definir operadores de comparación
- Crear sentencias de control if-else
- Crear una instancia de una clase
- Reconocer y describir la notación de puntos

Vocabulario:

Identifique el término correspondiente a cada una de las siguientes definiciones.

Notacion de puntos	Técnica que permite a una clase usar un método de otra clase u objeto. El punto entre el nombre de clase/objeto y el nombre del método indica que el método proviene de una clase u objeto diferente.
Constructor	Tipo de método especial que se ejecuta automáticamente cuando se crea una nueva instancia de la clase.
Palabra clave new	Palabra clave que indica que se está creando un nuevo objeto.
Operadores de comparacion	Símbolos que comparan dos valores aleatorios en un método.

Inténtelo/resuélvalo:

1. Escriba la firma del método que permite a una clase devolver un número aleatorio. Escriba su respuesta a continuación:

```
public static int getRandomNumber(int limit)
```

2. Indique tres razones por las que el comportamiento aleatorio será un componente importante del juego que está creando.
 1. Para mas versatilidad
 2. para eliminar la previsibilidad en el programa

3. Para que no sea repetitivo

3. En el método getRandomNumber(), ¿qué espera el parámetro? Escriba su respuesta a continuación:

Lo que esta esperando es un numero para lograr un rango.

4. ¿Cómo puede comprobar si un método es estático? Escriba su respuesta a continuación:

Si en la linea de «Public» a continuacion le sigue «Static»

5. Describa lo que hace el siguiente método. Escriba su respuesta a continuación. Greenfoot.getRandomNumber(15)

Dar un numero random de el 0 al 14

6. Cree una tabla que enumere los operadores de comparación y lo que hace cada uno.

Símbolo	Descripción
<	Menor que
>	Mayor que
<=	Menor o igual que
> =	Mayor o igual que
==	Igual
!=	Distinto de

7. Exprese la siguiente sentencia en código fuente: Queremos programar una flor para que gire un número aleatorio de grados, hasta 15 grados, el 28% del tiempo.

```
public void act()
{
    if(Greenfoot.getRandomNumber(100) < 28)
    {
        turn(Greenfoot.getRandomNumber(15));
    }
}
```

8. Exprese la siguiente sentencia en código fuente: Queremos que un abejorro gire 35 grados si se devuelve un

```
public void act()
{
    if(Greenfoot.getRandomNumber(10) < 5)
    {
        turn(Greenfoot.getRandomNumber(35));
    }
    else
    {
        turn(Greenfoot.getRandomNumber(35));
    }
}
```

número aleatorio menor que 5, y si se devuelve un número mayor que 5, queremos que el abejorro gire 2 grados.

9. En el siguiente constructor, ¿qué tamaño tiene el

```
public BeeWorld()  
{  
    super(550, 300, 1);  
}
```

El mundo tiene 550x300 de dimensiones y sus casillas son de 1x1.

10. En el constructor del mundo del escenario, escriba un código que cree automáticamente dos nuevas instancias de una clase Actor cuando se inicialice el mundo.

```
public class fondo extends World  
{  
    /**  
     * Constructor for objects of class fo  
     */  
    public fondo()  
    {  
        // Create a new world with 600x400  
        super(550, 300, 1);  
        addObject (new Bee(), 150, 100);  
        addObject (new pepe(), 300, 200);  
    }  
}
```

11. **Nota: la codificación desde este punto se realiza en el mismo escenario. Utilice el escenario base JF_3_5_Practice_Start como punto de partida.**

Vamos a crear un juego llamado Grab the Barrel.

Usted es el piloto de un avión y lo han enviado a recoger barriles que se han caído de un buque de carga. Al volar sobre un barril, lo recogerá automáticamente. El problema es que el mar está revuelto y los barriles se hunden en el agua, por lo que tendrá que recogerlos cuando estén en la superficie. Para empeorar aún más las cosas, la zona es un área de pruebas de cohetes de la NASA y no pueden dejar de disparar cohetes hacia la misma zona en la que está.

Usted debe programar este juego.

El avión siempre se estará moviendo, pero podemos controlar sus giros a la izquierda y a la derecha. Al volar sobre un barril, lo recogerá y buscará el siguiente. Al recoger un barril, ganará puntos.

El barril aparecerá al azar en la pantalla, pero solo estará una cantidad de tiempo establecida y, a continuación, se hundirá y volverá a aparecer en otra parte de la pantalla.

Los cohetes aparecerán en la parte superior de la pantalla y, a continuación, se moverán hacia abajo aleatoriamente hasta que desaparezcan del mapa.

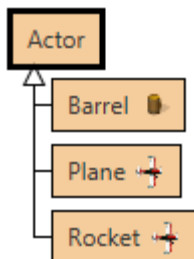
No debe chocar con el cohete. Se estima que podrá sobrevivir a 3 choques directos con cohetes y, a continuación, el juego terminará.

Abra el escenario del curso "JF_3_5_Practice_Start".

12. Cree una subclase de la clase World llamada BarrelWorld con las dimensiones 800 x 600. Utilice la imagen wet-blue.jpg como imagen de escenario.

```
public BarrelWorld()  
{  
    super(800, 600, 1);  
}
```

13. Cree una subclase de Actor llamada Plane y utilice la imagen airplane1.
14. Cree una subclase de Actor llamada Barrel y utilice la imagen Barrel.
15. Cree otra subclase de Actor llamada Rocket y utilice la imagen rocket1.



16. Introduzca el código para agregar automáticamente un barril en la parte superior izquierda.

```

public void resetBarrel()
{
    timer++;
    if (timer > maxtime)
    {
        timer = 0;
        setLocation(Greenfoot.getRandomNumber(650)+70, Greenfoot.getRandomNumber(450)+200);
    }
}

```

17. Introduzca el código para agregar un cohete en la parte superior de la pantalla. Este código generará de forma aleatoria la coordenada x y definirá la coordenada y en 10.

```

addObject (new Barrel(250),(Greenfoot.getRandomNumber(100)) ,(Greenfoot.getRandomNumber(100)));

```

18. Introduzca el código para agregar automáticamente la instancia de avión en el centro de la pantalla.

```

addObject (new Plane(), 400, 300);

```

19. Introduzca el código a la clase Plane para que se mueva constantemente hacia adelante a una velocidad de 2 y gire a la izquierda cuando se pulse la tecla de flecha izquierda o a la derecha cuando se pulse la tecla de flecha derecha. Girará 2 grados.

```

move(2);
if (Greenfoot.isKeyDown("left"))
{
    turn(-2);
}
if (Greenfoot.isKeyDown("right"))
{
    turn(2);
}

```