

Introducción a la programación

Clase 6

Profesor – Martín Gómez Vega

Da Vinci
PRIMERA ESCUELA DE ARTE MULTIMEDIAL

Agenda

1
...

Generación de datos aleatorios en programación

2
...

Uso de datos aleatorios en el lenguaje de programación Java

3
...

Casos especiales

4
...

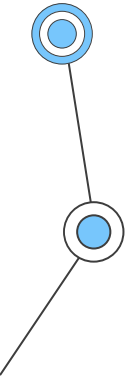
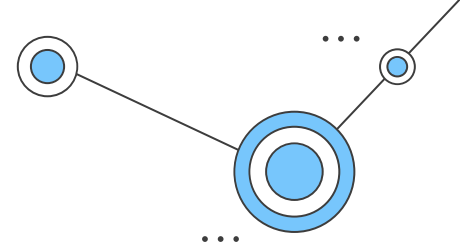
Generación de datos aleatorios en Structorizer

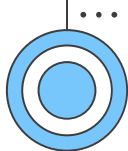
5
...

Selección múltiple en Structorizer

6
...

Selección múltiple - switch





Generación de datos aleatorios en programación

Todos los lenguajes permiten generar números al azar o aleatorios, que se llaman RANDOM (significa azar en inglés). Para poder generarlos en Java, se utiliza la siguiente función:

`Math.random()`

El método `random()` de la clase `Math` devuelve un número al azar positivo de tipo `double` mayor o igual que 0.0 y menor que 1.0

No podemos cambiar esto, pero si se le puede manejar desde código para poder tener el número que necesitamos.

Siempre lo genera como un número real, de la siguiente forma:

Para pedir un número real:

```
variable = Math.random();
```

Este número lo genera en el intervalo $[0,1)$, es decir, siempre toma el 0 nunca el 1.

Los posibles resultados pueden ser:

- 0,00123
- 0,51432
- 0,99999

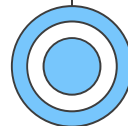
Para obtener un número real entre 0 y 10:

```
variable = Math.random() * 10;
```

Genera un número entre $[0,10)$ toma el 0 pero nunca el 10.



...



...

Agenda

1
...

Generación de datos aleatorios en programación

2
...

Uso de datos aleatorios en el lenguaje de programación Java

3
...

Casos especiales

4
...

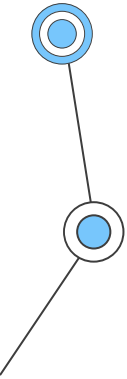
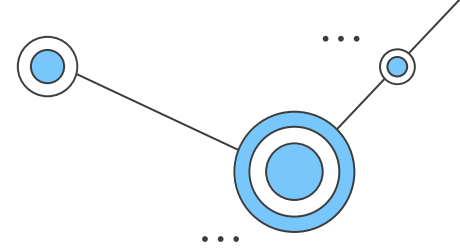
Generación de datos aleatorios en Structorizer

5
...

Selección múltiple en Structorizer

6
...

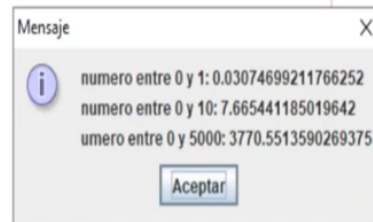
Selección múltiple - switch



Uso de datos aleatorios en el lenguaje de programación Java

En código Java se implementa de la siguiente forma.

```
14 public class Nuevorandom {
15
16     /**
17      * @param args the command line arguments
18      */
19     public static void main(String[] args) {
20         double a,b,c;
21         a=Math.random();
22         b=Math.random()*10;
23         c=Math.random()*5000;
24         JOptionPane.showMessageDialog(null, "numero entre 0 y 1: "+a
25         +"\nnumero entre 0 y 10: "+b+"\nnumero entre 0 y 5000: "+c);
26
27     }
28
29 }
```



Para obtener un número random entero:

Int n;

n = (int) (Math.random() * 10);

Agenda

1
...

Generación de datos aleatorios en programación

2
...

Uso de datos aleatorios en el lenguaje de programación Java

3
...

Casos especiales

4
...

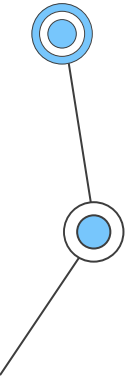
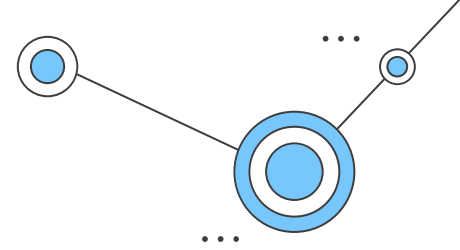
Generación de datos aleatorios en Structorizer

5
...

Selección múltiple en Structorizer

6
...

Selección múltiple - switch



Casos especiales

Ejemplo – Número de dado

Esto genera un número hasta 5 y se le suma 1 para correr los valores de lugar, y de esta forma evitamos el 0.

```
int dado;  
dado=(int)(Math.random()* 6 +1);
```

Ejemplo – Números negativos

No genera números negativos pero para obtenerlos los multiplicamos por el valor -1.

```
int n;  
n=(int)(Math.random()* 6 +1);
```

```
9  *  
0  * @author elda  
1  */  
2  public class Nuevorandom {  
3  
4  /**  
5  * @param args the command line arguments  
6  */  
7  public static void main(String[] args) {  
8      int dado,carta,neg, rango;  
9      dado=(int)(Math.random()*6+1); // valor entre 1 y 6  
10     carta=(int)(Math.random()*12+1); // valor entre 1 y 12  
11     neg=(int)(Math.random()*100*-1); // valor negativo hasta 100  
12     rango=(int)(Math.random()*11+10); // valor rango 10-20  
13     JOptionPane.showMessageDialog(null, "valor dado: "+dado  
14     +"\nvalor carta: "+carta+"\nvalor negativo: "+neg+ "\nrango 10-20: "+ rango);  
15 }  
16  
17 }
```



Agenda

1
...

Generación de datos aleatorios en programación

2
...

Uso de datos aleatorios en el lenguaje de programación Java

3
...

Casos especiales

4
...

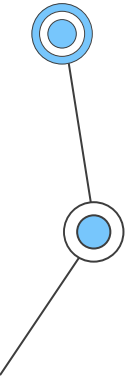
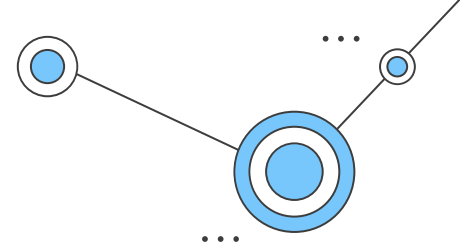
Generación de datos aleatorios en Structorizer

5
...

Selección múltiple en Structorizer

6
...

Selección múltiple - switch



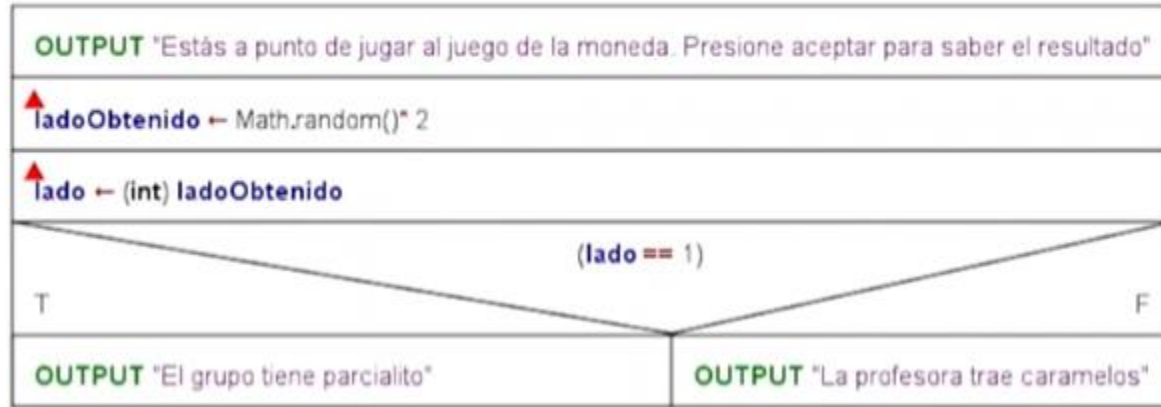
Generación de datos aleatorios en Structorizer

Juego de la moneda

Se tira una moneda al aire y se pide ingresar el número:

- 0 – cara: la profesora trae caramelos
- 1 – ceca: tienen parcialito

JuegoDeLaMoneda



Agenda

1
...

Generación de datos aleatorios en programación

2
...

Uso de datos aleatorios en el lenguaje de programación Java

3
...

Casos especiales

4
...

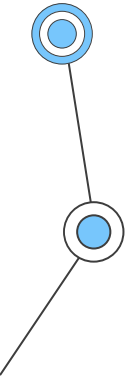
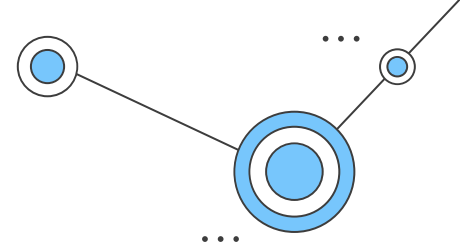
Generación de datos aleatorios en Structorizer

5
...

Selección múltiple en Structorizer

6
...

Selección múltiple - switch



Selección múltiple en Structorizer

Permite trabajar con múltiples alternativas, solo trabaja con dos tipos de datos String o int, no trabaja con double, y se usa para casos puntuales.

No trabajo con signos < o >, con ninguno.

Estructura general

Selector				
Valor 1	Valor 2	Valor n	/
Acción a	Acción b	Acción z	/

estructura general					
valor 1	valor 2	valor 3	valor n	selector
accion a	accion b	accion c	accion z	/
					mostrar "error"

Agenda

1
...

Generación de datos aleatorios en programación

2
...

Uso de datos aleatorios en el lenguaje de programación Java

3
...

Casos especiales

4
...

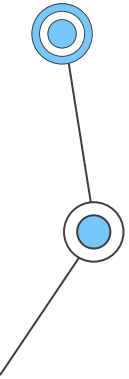
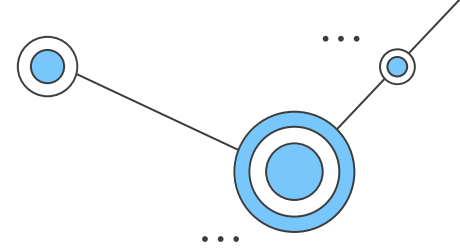
Generación de datos aleatorios en Structorizer

5
...

Selección múltiple en Structorizer

6
...

Selección múltiple - switch

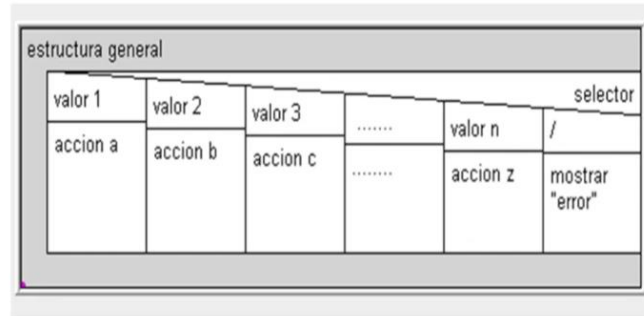


Selección múltiple – switch

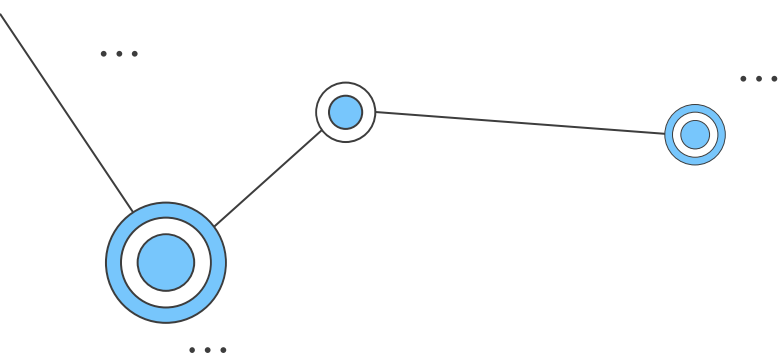
Esta estructura se utiliza para tomar decisiones basadas en el valor de una expresión y ejecutar un bloque de código específico correspondiente a ese valor.

Se utiliza principalmente para simplificar el código en situaciones en las que se tiene que comparar una expresión con varios valores posibles. Es una alternativa más legible y eficiente que utilizar una serie de declaraciones "if-else" anidadas.

```
switch (variable){  
  case valor1: acción a;  
    break;  
  case valor2: acción b;  
    break;  
  case valor3: acción c;  
    break;  
  .....  
  case valorn: acción z;  
    break;  
  default: (mensaje de error);  
    break;  
}
```

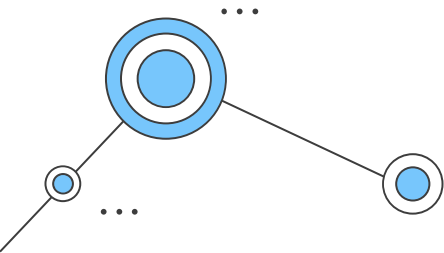


es opcional



Práctica

Crear un sistema que permita al usuario adivinar el número de un dado. (con tres intentos)



¿Dudas?