



Instituto Politécnico Nacional
Centro de Investigación en Computo



Alumno
RODRÍGUEZ ARRENDONDO JACOBO
jacobo.rodriquez499@gmail.com

Diplomado en lenguaje Java

Facilitador: Alan Badillo Salas

Módulo: 1

Examen: 1

Fecha de entrega: 4 de Mayo de 2022

1) Ejemplo de uso de la estructura condicional “if”

Programa que crea una lista con números aleatorios de 0 a 10 y evalúa si es mayor a 5 para tener una calificación aprobatoria o reprobatoria.

```
"C:\Program Files\Java\jdk-17.0.1\bin\java.exe" '
  IDEA 2021.3.3\bin" -Dfile.encoding=UTF-8 -classpath
Ejericio de if
Cantidad de Alumnos: 8
El alumno 1 paso con: 7.489698
El alumno 2 no paso: 5.446450
El alumno 3 paso con: 6.368624
El alumno 4 no paso: 0.789730
El alumno 5 paso con: 6.304537
El alumno 6 paso con: 7.561597
El alumno 7 no paso: 4.114927
El alumno 8 paso con: 7.199345

Process finished with exit code 0
```

Ilustración 1 Ejercicio if

2) Ejemplo de uso de la estructura del ciclo iterador “for”

Programa que crea un `ArrayList<Double>` el cual lo llena con valores al azar del 0 a 100 y haciendo uso de un ciclo `for`, posteriormente ordena la lista usando un `for` anidado para emplear el método burbuja y ordenarlo, también se hace uso de la función `ImpArrayDouble` el cual uso un `for` de elemento para imprimir cualquier `ArrayList` tipo `double`

```
"C:\Program Files\Java\jdk-17.0.1\bin\java.exe" "-ja
IDEA 2021.3.3\bin" -Dfile.encoding=UTF-8 -classpath
Ejercicio de for
Cantidad de elementos: 46
lista antes de ser ordenada
lst[1]: 0.357196
lst[2]: 0.160772
lst[3]: 0.389550
lst[4]: 0.530757
lst[5]: 0.268017
lst[6]: 0.545966
lst[7]: 0.039574
lst[8]: 0.798520
lst[9]: 0.730354
lst[10]: 0.183271
lst[11]: 0.638104
lst[12]: 0.184376
lst[13]: 0.346150
lst[14]: 0.696838
lst[15]: 0.924986
lst[16]: 0.095389
lst[17]: 0.244760
lst[18]: 0.546156
lst[19]: 0.783748
lst[20]: 0.364387
lst[21]: 0.156549
lst[22]: 0.121436
lst[23]: 0.876272
lst[24]: 0.591724
lst[25]: 0.635896
lst[26]: 0.979184
lst[27]: 0.876526
lst[28]: 0.974464
lst[29]: 0.289212
lst[30]: 0.448818
lst[31]: 0.559851
lst[32]: 0.196161
lst[33]: 0.454419
lst[34]: 0.377352
lst[35]: 0.798254
lst[36]: 0.814612
lst[37]: 0.025396
lst[38]: 0.221494
lst[39]: 0.462340
lst[40]: 0.855535
lst[41]: 0.018228
lst[42]: 0.045203
lst[43]: 0.336977
lst[44]: 0.996812
lst[45]: 0.035672
lst[46]: 0.235586
```

Ilustración 2 lista antes de ser ordenada

```

Lista después de ser ordenada
lst[1]: 0.018228
lst[2]: 0.025396
lst[3]: 0.035672
lst[4]: 0.039574
lst[5]: 0.045203
lst[6]: 0.095389
lst[7]: 0.121436  lst[28]: 0.545966
lst[8]: 0.156549  lst[29]: 0.546156
lst[9]: 0.160772  lst[30]: 0.559851
lst[10]: 0.183271 lst[31]: 0.591724
lst[11]: 0.184376 lst[32]: 0.635896
lst[12]: 0.196161 lst[33]: 0.638104
lst[13]: 0.221494 lst[34]: 0.696838
lst[14]: 0.235586 lst[35]: 0.730354
lst[15]: 0.244760 lst[36]: 0.783748
lst[16]: 0.268017 lst[37]: 0.798254
lst[17]: 0.289212 lst[38]: 0.798520
lst[18]: 0.336977 lst[39]: 0.814612
lst[19]: 0.346150 lst[40]: 0.855535
lst[20]: 0.357196 lst[41]: 0.876272
lst[21]: 0.364387 lst[42]: 0.876526
lst[22]: 0.377352 lst[43]: 0.924986
lst[23]: 0.389550 lst[44]: 0.974464
lst[24]: 0.448818 lst[45]: 0.979184
lst[25]: 0.454419 lst[46]: 0.996812
lst[26]: 0.462340
lst[27]: 0.530757 Process finished with exit code 0

```

Ilustración 3 lista después de ser ordenada

3) Ejemplo de uno de la estructura del ciclo “while”

Programa que crea una variable “n” double con un valor random de $0 \leq n \leq 10,000$ el cual es dividido por el valor 2^i , donde $i=0,1,2,\dots,n$, mientras que el valor de n sea mayor o igual a 1.

```

"C:\Program Files\Java\jdk-17.0.1\bin\java.exe
IDEA 2021.3.3\bin" -Dfile.encoding=UTF-8 -cla
Ejercicio de while
while n: 2329.049337 > 1
ciclo: 1 | n: 2329.049337 | opc: 1.000000
ciclo: 2 | n: 1164.524668 | opc: 2.000000
ciclo: 3 | n: 291.131167 | opc: 4.000000
ciclo: 4 | n: 36.391396 | opc: 8.000000
ciclo: 5 | n: 2.274462 | opc: 16.000000
ciclo: 6 | n: 0.071077 | opc: 32.000000

Process finished with exit code 0

```

Ilustración 4 ejecución del ejercicio while

4) Ejemplo de la estructura selector switch

Programa que permite seleccionar al usuario uno de los ejercicios posteriormente realizados para ejecutarlo.

```
"C:\Program Files\Java\jdk-17.0.1\bin\java.exe
IDEA 2021.3.3\bin" -Dfile.encoding=UTF-8 -cla
Seleccione una de las siguiente opciones:
1) Ejercicio if
2) Ejercicio for
3) Ejercicio while
9) salir
opción: 2
Cantidad de elementos: 80
Lista antes de ser ordenada
lst[1]: 0.759643
lst[2]: 0.682963
lst[3]: 0.753421
lst[4]: 0.210351
lst[5]: 0.476939
lst[6]: 0.858266
lst[7]: 0.840264
lst[8]: 0.270169
lst[9]: 0.541169
lst[10]: 0.360740
lst[11]: 0.058032
lst[12]: 0.750168
lst[13]: 0.545057
lst[14]: 0.102799
lst[15]: 0.823614
```

Ilustración 5 -switch 1

```
lst[80]: 0.996023
Seleccione una de las siguiente opciones:
1) Ejercicio if
2) Ejercicio for
3) Ejercicio while
9) salir
opción: 3
while n: 1518.295758 > 1
ciclo: 1 | n: 1518.295758 | opc: 1.000000
ciclo: 2 | n: 759.147879 | opc: 2.000000
ciclo: 3 | n: 189.786970 | opc: 4.000000
ciclo: 4 | n: 23.723371 | opc: 8.000000
ciclo: 5 | n: 1.482711 | opc: 16.000000
ciclo: 6 | n: 0.046335 | opc: 32.000000
```

Ilustración 6 Switch 2

```
Seleccione una de las siguiente opciones:
1) Ejercicio if
2) Ejercicio for
3) Ejercicio while
9) salir
opción: 9
Seleccione una de las siguiente opciones:
1) Ejercicio if
2) Ejercicio for
3) Ejercicio while
9) salir
opción:
Process finished with exit code 0
```

Ilustración 7 switch - 3

5) Clase contador

Una vez creado la clase contador con los métodos especificado se procedió a elaborar una implementación en la clase "Evaluación" donde se crean inicialización de objeto random, las variables i y j son variables para determinar el valor inicial del c1 y c2 respectivamente, opc es la cantidad de ciclos que se va a repetir 100 veces las operaciones, después se reutilizan las variables para determinar si son pares o impares, cada vez que opc sea un múltiplo de 100 se imprimiría el ciclo y los valores que tienen los contadores en ese ciclo, después se pregunta si i y j son pares se incrementan sus conteos, si i y j son impares se decrementan sus conteos, si i es par se incrementa el conteo de c1 y se decrementa el conteo de c2, si j es par se incrementa el conteo de c2 y se decrementa el conteo de c1 y por último un caso especial donde no se cumplen las condiciones se imprime el ciclo y los valores de cada contador.

```
"C:\Program Files\Java\jdk-17.0.1\bin\java.exe" "-j
IDEA 2021.3.3\bin" -Dfile.encoding=UTF-8 -classpath
Ejericio de clase conteo
inicio | c1 conteo: 91 | c2 conteo: 80
cilco: 1 | c1 conteo: 91 | c2 conteo: 80
cilco: 3 | c1 conteo: 90 | c2 conteo: 81
cilco: 5 | c1 conteo: 89 | c2 conteo: 82
cilco: 7 | c1 conteo: 90 | c2 conteo: 83
cilco: 9 | c1 conteo: 91 | c2 conteo: 82
cilco: 11 | c1 conteo: 90 | c2 conteo: 83
cilco: 13 | c1 conteo: 91 | c2 conteo: 84
cilco: 15 | c1 conteo: 90 | c2 conteo: 83
cilco: 17 | c1 conteo: 89 | c2 conteo: 84
cilco: 19 | c1 conteo: 88 | c2 conteo: 85
cilco: 21 | c1 conteo: 89 | c2 conteo: 86
cilco: 23 | c1 conteo: 88 | c2 conteo: 85
cilco: 25 | c1 conteo: 89 | c2 conteo: 84
cilco: 27 | c1 conteo: 88 | c2 conteo: 85
cilco: 29 | c1 conteo: 89 | c2 conteo: 86
cilco: 31 | c1 conteo: 88 | c2 conteo: 87
cilco: 33 | c1 conteo: 87 | c2 conteo: 88
cilco: 35 | c1 conteo: 86 | c2 conteo: 87
cilco: 37 | c1 conteo: 85 | c2 conteo: 86
cilco: 39 | c1 conteo: 84 | c2 conteo: 87
cilco: 41 | c1 conteo: 85 | c2 conteo: 86
cilco: 43 | c1 conteo: 86 | c2 conteo: 87
cilco: 45 | c1 conteo: 85 | c2 conteo: 88
```

Ilustración 8 ejecución del ejercicio de clase contador

```
cilco: 65 | c1 conteo: 87 | c2 conteo: 92  
cilco: 67 | c1 conteo: 88 | c2 conteo: 93  
cilco: 69 | c1 conteo: 89 | c2 conteo: 94  
cilco: 71 | c1 conteo: 88 | c2 conteo: 93  
cilco: 73 | c1 conteo: 87 | c2 conteo: 94  
cilco: 75 | c1 conteo: 86 | c2 conteo: 95  
cilco: 77 | c1 conteo: 85 | c2 conteo: 94  
cilco: 79 | c1 conteo: 86 | c2 conteo: 95  
cilco: 81 | c1 conteo: 87 | c2 conteo: 94  
cilco: 83 | c1 conteo: 88 | c2 conteo: 95  
cilco: 85 | c1 conteo: 87 | c2 conteo: 96  
cilco: 87 | c1 conteo: 86 | c2 conteo: 95  
cilco: 89 | c1 conteo: 85 | c2 conteo: 96  
cilco: 91 | c1 conteo: 86 | c2 conteo: 95  
cilco: 93 | c1 conteo: 85 | c2 conteo: 96  
cilco: 95 | c1 conteo: 86 | c2 conteo: 97  
cilco: 97 | c1 conteo: 85 | c2 conteo: 96  
cilco: 99 | c1 conteo: 86 | c2 conteo: 97
```

```
Process finished with exit code 0
```

Ilustración 9