

Practica 1 - Programación de Dispositivos Móviles

De: Rodrigo Alonso Ruelas Lope

Fecha: 24 de agosto del 2024

1. Leer datos por teclado `readln()`

Este es un comando para poder ingresar datos por medio de la consola, esto gracias al comando `readLine()`, donde se ingresarán datos del tipo `String`.

En la práctica, se utiliza de la siguiente manera:

```
val Nombre = readLine()
```

Más adelante veremos su aplicación en los problemas propuestos avanzados.

2. Leer valores aleatorios con `random()`

Con este comando podremos generar valores aleatorios, con la ayuda del comando `random()`, donde se podrá escoger números aleatorios del tipo `Int` o `Double`, con o sin un rango específico.

Primero, debemos importar una librería específica para poder usar el comando antes mencionado:

- `import kotlin.random.Random`

Con esta importación, ahora podremos usar el comando `random()`. Como se mencionó anteriormente, podemos seleccionar un rango específico para seleccionar un valor aleatorio.

- `val numAle = Random.nextInt(1, 10)`
- `val numAle = Random.nextDouble(5.0, 10.0)`

Si no queremos darle un rango, simplemente ignoramos los paréntesis del final, por ejemplo.

- `val numAle = Random.nextInt()`
- `val numAle = Random.nextDouble()`



3. Problemas Propuestos

- a) Problema 1. Determinar el salario de un empleado mediante una evaluación de desempeño.

Código Implementación.

```
1 // Pregunta 1: Determinar salario de un empleado mediante una evaluación de desempeño
2 // De: Rodrigo Alonso Ruelas Lope
3 // Fecha de Creacion: 24/08/2024 - 9:09 am
4 // Ultima modificacion: 24/08/2024 - 9:49 am
5
6 fun main(args: Array<String>) {
7     var Puntuacion_E : Int = 8
8     var Salario_E : Double = 10000.00
9
10    println(" -- Resultados de la evaluacion -- ")
11
12    println("Nivel de rendimiento")
13    Hallar_Nivel(Puntuacion_E)
14    println("Dinero recibido")
15    print(Hallar_Dinero(Puntuacion_E, Salario_E))
16 }
17
18 fun Hallar_Nivel(punt : Int) : Unit {
19     when (punt) {
20         in 0 ≤ .. ≤ 3 -> println("Inaceptable")
21         in 4 ≤ .. ≤ 6 -> println("Aceptable")
22         in 7 ≤ .. ≤ 10 -> println("Meritorio")
23         else -> println("Puntuacion no valida")
24     }
25 }
26
27 fun Hallar_Dinero(punt : Int, sal : Double) : Unit {
28     val Dinero : Double = sal * (punt/10.0)
29     println(Dinero)
30 }
```

Resultados.

```
-- Resultados de la evaluacion --
Nivel de rendimiento
Meritorio
Dinero recibido
8000.0
```

b) Problema 2. Juego de Piedra-Papel-Tijera.
Código Implementación.


```
1 // Pregunta 2: Juego de Piedra-Papel-Tijera.
2 // De: Rodrigo Alonso Ruelas Lope
3 // Fecha de Creacion: 24/08/2024 - 10:30 am
4 // Ultima modificacion: 24/08/2024 - 11:02 am
5
6 import kotlin.random.Random
7
8 fun obtenerOpcion(opcion: Int): String {
9     return when (opcion) {
10         1 -> "Piedra"
11         2 -> "Papel"
12         3 -> "Tijera"
13         else -> ""
14     }
15 }
16
17 fun determinarGanador(jugador: Int, computadora: Int): Int {
18     return when {
19         jugador == computadora -> 0 // Empate
20         (jugador == 1 && computadora == 3) || // Piedra vence a Tijera
21         (jugador == 2 && computadora == 1) || // Papel vence a Piedra
22         (jugador == 3 && computadora == 2) -> 1 // Tijera vence a Papel
23         else -> -1
24     }
25 }
26
27 fun main() {
28     println("Bienvenido al juego de Piedra, Papel o Tijera!")
29     println("Elige una opción:\n1. Piedra\n2. Papel\n3. Tijera")
30
31     val jugador = readLine()?.toIntOrNull()
32
33     if (jugador == null || jugador !in 1..3) {
34         println("Opción inválida, por favor selecciona 1, 2 o 3.")
35         return
36     }
37
38     val computadora = Random.nextInt(from: 1, until: 4)
39
40     println("Tú elegiste: ${obtenerOpcion(jugador)}")
41     println("La computadora eligió: ${obtenerOpcion(computadora)}")
42
43     val resultado = determinarGanador(jugador, computadora)
44
45     when (resultado) {
46         0 -> println("¡Es un empate!")
47         1 -> println("¡Ganaste!")
48         -1 -> println("Perdiste, la computadora gana!")
49     }
50 }
51
```

Resultados.

```
Bienvenido al juego de Piedra, Papel o Tijera!  
Elige una opción:  
1. Piedra  
2. Papel  
3. Tijera  
1  
Tú elegiste: Piedra  
La computadora eligió: Tijera  
¡Ganaste!
```

c) Problema 3. Implementación de una calculadora básica.

Código Implementación.

```
1 // Pregunta 3. Implementación de una calculadora básica.  
2 // De: Rodrigo Alonso Ruelas Lope  
3 // Fecha de Creacion: 24/08/2024 - 04:00 pm  
4 // Ultima modificacion: 24/08/2024 - 04:50 pm  
5  
6  fun main() {  
7     println("¡Bienvenido a la calculadora en Kotlin!")  
8     while (true) {  
9         val operacion : Int = seleccionarOperacion()  
10  
11         if (operacion == 5) {  
12             println("Saliendo de la calculadora. ¡Adiós!")  
13             break  
14         }  
15  
16         val num1 = obtenerNumero( mensaje: "Ingresa el primer número:")  
17         val num2 = obtenerNumero( mensaje: "Ingresa el segundo número:")  
18  
19         val resultado = realizarOperacion(operacion, num1, num2)  
20         println("El resultado de la operación es: $resultado\n")  
21     }  
22 }  
23  
24 fun seleccionarOperacion() : Int {  
25     while (true) {  
26         println("Selecciona una operación:")  
27         println("1. Suma")  
28         println("2. Resta")  
29         println("3. Multiplicación")  
30         println("4. División")  
31         println("5. Salir")
```

```

33         val operacion = readLine()?.toIntOrNull()
34
35         if (operacion != null && operacion in 1..5) {
36             return operacion
37         } else {
38             println("Opción inválida. Por favor selecciona una operación válida.")
39         }
40     }
41 }
42
43
44 fun obtenerNumero(mensaje: String): Double {
45     println(mensaje)
46     val numero = readLine()?.toDoubleOrNull()
47
48     if (numero == null) {
49         println("Entrada inválida. Por favor ingresa un número válido.")
50         return obtenerNumero(mensaje)
51     }
52
53     return numero
54 }
55
56 fun realizarOperacion(operacion: Int, num1: Double, num2: Double): Double {
57     return when (operacion) {
58         1 -> suma(num1, num2)
59         2 -> resta(num1, num2)
60         3 -> multiplicacion(num1, num2)
61         4 -> {
62             if (num2 == 0.0) {
63                 println("Error: No se puede dividir por cero.")

```

```

64                 0.0
65             } else {
66                 division(num1, num2)
67             }
68         }
69         else -> 0.0
70     }
71 }
72
73 fun suma(a: Double, b: Double): Double {
74     return a + b
75 }
76
77 fun resta(a: Double, b: Double): Double {
78     return a - b
79 }
80
81 fun multiplicacion(a: Double, b: Double): Double {
82     return a * b
83 }
84
85 fun division(a: Double, b: Double): Double {
86     return a / b
87 }

```

Resultados.

```
¡Bienvenido a la calculadora en Kotlin!
Selecciona una operación:
1. Suma
2. Resta
3. Multiplicación
4. División
5. Salir
3
Ingresa el primer número:
5.3
Ingresa el segundo número:
6
El resultado de la operación es: 31.799999999999997

Selecciona una operación:
1. Suma
2. Resta
3. Multiplicación
4. División
5. Salir
5
Saliendo de la calculadora. ¡Adiós!
```

d) Problema 4. Juego de adivinar un número.

Código Implementación.

```
1 // Pregunta 4: Juego de adivinar un número.
2 // De: Rodrigo Alonso Ruelas Lope
3 // Fecha de Creacion: 24/08/2024 - 4:55 pm
4 // Ultima modificacion: 24/08/2024 - 5:20 pm
5
6 import kotlin.random.Random
7
8 fun main() {
9     val numeroSecreto = generarNumeroSecreto(inicio = 1, fin = 30)
10    val intentosMaximos = 5
11
12    println("¡Bienvenido al juego de adivinar el número!")
13    println("He escogido un número entre 1 y 30. Tienes $intentosMaximos intentos para adivinarlo.")
14
15    val adivinadoCorrectamente = jugar(intentosMaximos, numeroSecreto)
16
17    if (adivinadoCorrectamente) {
18        println("¡Felicidades! ¡Has adivinado el número!")
19    } else {
20        println("Lo siento, se te acabaron los intentos. El número era $numeroSecreto.")
21    }
22 }
23
24 fun generarNumeroSecreto(inicio: Int, fin: Int): Int {
25     return Random.nextInt(inicio, until = fin + 1)
26 }
27
28 fun jugar(intentosMaximos: Int, numeroSecreto: Int): Boolean {
29     var intentosRestantes = intentosMaximos
30
31     while (intentosRestantes > 0) {
32         val suposicion = obtenerSuposicion()
```

```

34         if (suposicion == null) {
35             println("Por favor, introduce un número válido.")
36             continue
37         }
38
39         if (evaluarSuposicion(suposicion, numeroSecreto)) {
40             return true
41         }
42
43         intentosRestantes--
44         if (intentosRestantes > 0) {
45             println("Te quedan $intentosRestantes intentos.")
46         }
47     }
48     return false
49 }
50
51 fun obtenerSuposicion(): Int? {
52     println("\nIntroduce tu suposición:")
53     return readLine()?.toIntOrNull()
54 }
55
56 fun evaluarSuposicion(suposicion: Int, numeroSecreto: Int): Boolean {
57     return when {
58         suposicion < numeroSecreto -> {
59             println("¡Demasiado bajo!")
60             false
61         }
62         suposicion > numeroSecreto -> {
63             println("¡Demasiado alto!")
64             false
65         }
66     }

```

Resultado.

```

¡Bienvenido al juego de adivinar el número!
He escogido un número entre 1 y 30. Tienes 5 intentos para adivinarlo.

Introduce tu suposición:
8
¡Demasiado alto!
Te quedan 4 intentos.

Introduce tu suposición:
5
¡Demasiado alto!
Te quedan 3 intentos.

Introduce tu suposición:
6
¡Demasiado alto!
Te quedan 2 intentos.

Introduce tu suposición:
4
¡Felicidades! ¡Has adivinado el número!

```