

Lenguaje de Programación: Ciclos y Condicionales (instrucción iterate)

José Luis Alonzo Velázquez

Universidad de Guanajuato

Sesión 5

La instrucción iterate

Esta instrucción es empleada cuando una misma instrucción(o bloque) requiere ser repetida.

La sintaxis de iterate es:

```
iterate(<número positivo>)  
  {  
    <instrucción>  
    <instrucción>  
    :  
    <instrucción>  
  }
```

Nota:

Cuando sólo queremos repetir una sola instrucción, es posible omitir las llaves “{” y “}”. Sin embargo recomiendo que se utilicen al principio, aunque sólo sea una instrucción.

Ejemplo: Vuelta a la derecha

```
define turnright()  
  {  
    iterate(3)  
      {  
        turnleft();  
      }  
  }
```

Cosechando

Como Karel siempre ha sido muy atento con su familia, decidió que era tiempo de contribuir con los ingresos de la familia. Por lo tanto consiguió un trabajo para cosechar hileras de 20 a 100 esquinas (ya que son las mejores pagadas). Hacer un programa que coseche 20 o 100 esquinas.

```
class program
{
  define cosecha()
  {
    iterate(20)
    {
      pickbeeper();
      move();
    }
  }
  program()
  {
    cosecha();
    turnoff();
  }
}
```

Anidando ciclos

En general, es posible anidar tantos ciclos como sean necesarios. Esto es muy semejante a lo que ocurría cuando definíamos una instrucción que usaba otra instrucción definida antes.

Ejemplo: Sembrando

El Dr. Karelov desea colocar 10 beepers en 5 esquinas a lo largo de una calle. Por ello, Karel debe ser programado para hacer el trabajo por él.

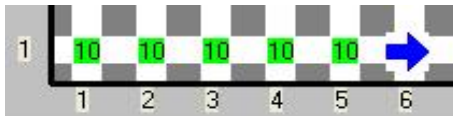


Figura: Colocando 10 beepers en 5 esquinas consecutivas.

Programa

```
program()  
  {  
    iterate(5)  
      {  
        iterate(10)  
          {  
            putbeeper();  
          }  
        move();  
      }  
  }
```

Cosechando

El Dr. Karelov está complacido por el trabajo de Karel. Por ello, le ha pedido que recoja todos los tornillos que hay en el suelo. Hay 5 tornillos por esquina a lo largo de un cuadrado de 6 avenidas y 4 calles.

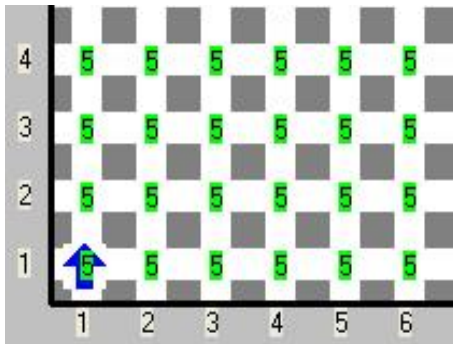


Figura: Tornillos en el cuarto.

```
program()  
{  
  iterate(6)  
  {  
    iterate(4)  
    {  
      iterate(5)  
      pickbeeper();  
      move();  
    }  
    halfturn();  
    iterate(4)  
    move();  
    leftmove();(*gira a la izquierda y da un paso*)  
    turnleft();  
  }  
  turnoff();  
}
```

Errores en ciclos

Durante el uso de ciclos, es común cometer errores entre una iteración y otra. Es decir, los errores se producen justo después de terminar una iteración o al iniciar una.

Ejemplo: Cosechando la calle

En cierta ocasión el Dr. Karellov envió a Karel por unos beepers a lo largo de una calle que llegaba hasta una pared. Has un programa para que Karel recoja todos los beepers hasta topar con pared.

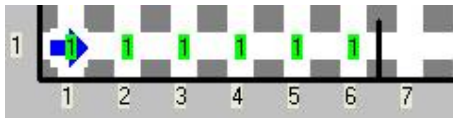


Figura: Recogiendo los beepers hasta la pared.

El típico código sería

```
program()  
  {  
    iterate(6)  
      {  
        pickbeeper();  
        move();  
      }  
  }
```

Resultado

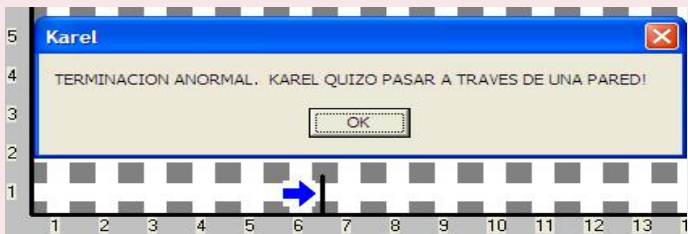


Figura: Mensaje de error al chocar con una pared.

Un programa que si funciona bien es:

```
program()  
  {  
    iterate(5)  
      {  
        pickbeeper();  
        move();  
      }  
    pickbeeper();  
  }
```




Edgar Alfredo Duéñez Guzmán & Edgar Said Hernández
Sánchez & Marte Alejandro Ramírez Ortega *Los Dilemas de
Karel*. CIMAT, Mayo 2006.