



Gobstones

Introducción a la Programación - Práctica 3

Procedimientos con parámetros Repetición

CONSEJOS:

- Leer el enunciado en su totalidad y pensar en la forma de resolverlo **ANTES** de empezar a escribir código.
- Si un ejercicio no sale, se puede dejar para después y continuar con los ejercicios que siguen.
- Los ejercicios están pensados para ser hechos después de haber mirado la teórica correspondiente, y esta después de las actividades de indagación.
- Los ejercicios están tomados de las guías prácticas utilizadas en la materia de Introducción a la Programación de la Universidad Nacional de Quilmes por Pablo Ernesto "Fidel" Martínez López y su equipo, Introducción a la Programación de la Universidad Nacional de Hurlingham de Alan Rodas Bonjour y su equipo, de ejercicios y actividades realizadas por Federico Aloí y Miguel Miloro, a su vez basada en las guías Ejercicios de Introducción a la Programación del CIU General Belgrano, elaboradas por Carlos Lombardi y Alfredo Sanzo, y Fundamentos de la Programación del Proyecto Mumuki. Agradecemos a todos los que nos ayudaron con su inspiración.
- Realizar **EN PAPEL** los ejercicios que así lo indiquen.
- Si un ejercicio indica **BIBLIOTECA** significa que será útil para la realización de futuros ejercicios tanto en esta guía como en las siguientes, pudiendo ser utilizado sin tener que volver a definirlo. Es útil mantener registro de dichos procedimientos en su carpeta.

EJERCICIOS:

1. Escribir un procedimiento **Mover3VecesAl_(direcciónAMover)** que dada una dirección **direcciónAMover** mueva el cabezal tres posiciones en dicha dirección. Puede usarse como base el procedimiento **Mover3VecesAlOeste** (realizado en la primera práctica). No olvidar escribir el contrato del procedimiento ANTES de realizar el código (y que los parámetros son parte del mismo); también discutir la precondition escrita con sus compañeros para verificar que la misma es adecuada y correcta.
2. Escribir un procedimiento **Poner_Al_(colorAPoner, direcciónHaciaDondePoner)** que dado un color **colorAPoner** y una dirección **direcciónHaciaDondePoner**, ponga una bolita del color dado en la celda vecina en la dirección dada, dejando el cabezal en dicha celda.
 - a. ¿Cuántos casos distintos habría que considerar si no se usarán parámetros en este caso?
 - b. ¿Cómo debería invocar al procedimiento para que ponga una bolita Azul en la celda al Norte?
 - c. ¿Y sí quisiera una Azul y una Roja?
3. Generalizar el ejercicio del reloj analógico de agujas de la práctica anterior para que se pueda pasar el radio como parámetro. O sea, escribir un procedimiento:

DibujarRelojAnalógicoSimplificadoDeRadio_(radio)

que ponga los números del reloj como en el programa original, pero donde el **radio** que recibido por parámetro indica el tamaño: mientras más grande es el radio, más alejados están los números del centro.

Por ejemplo, el ejercicio original puede obtenerse con el comando

DibujarRelojAnalógicoSimplificadoDeRadio_(2)

4. ¿Se acuerdan de Nova? Es el nuevo compañero del equipo, que no tiene formación profesional, y su código está lleno de malas prácticas de programación. Esta vez, en su código se encontró un procedimiento que tiene el código hecho, pero no el contrato, y no está indentado.
 - a. En primer lugar, se pide corregir los errores de Nova. No olvidar indicar en qué posición queda el cabezal, ni de establecer para qué son los parámetros. El código de Nova es el siguiente

```
procedure Poner_ADistancia3Al_(color, dirección) {  
  Mover3VecesAl_(dirección) Poner(color) }
```

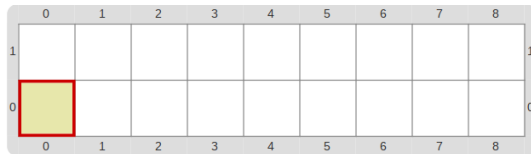
- b. En segundo lugar, se pide contestar la siguiente pregunta que realizó Nova:
¿Cuál es la relación entre el parámetro `dirección` de este procedimiento con el parámetro `direcciónAMover` de `Mover3VecesAl_`?
5. Construir un procedimiento `EscribirFecha___(día, mes, año)`, que permita representar cualquier fecha dados el día, mes y año. La representación debe ser la misma utilizada en el ejercicio 2 de la práctica 2 donde se registró el día de la Memoria. Recordar que debe comenzarse por escribir el contrato; en este caso puede resultar útil escribir también una observación con la representación a utilizar.
6. Construir un programa que escriba un listado vertical con las siguientes fechas:
- inicio de la Reforma Universitaria;
 - reglamentación del voto femenino en Argentina;
 - puesta en funcionamiento de la Tecnicatura Universitaria en Programación Informática de la UNQ;
 - fecha en la que ocurrieron los hechos conmemorados en el Día Internacional de los Trabajadores;
 - creación del Ministerio de Ciencia y Tecnología argentino;
 - primera celebración del Día de la Mujer;
 - disolución del Ministerio de Ciencia y Tecnología argentino.
- ¿Es necesario pensar procedimientos para escribir fechas o sirve algo de lo realizado con anterioridad?
7. Al continuar revisando código, encontramos otro procedimiento de Nova que carece de contrato.

- a. Escribir el contrato faltante

```
procedure Pintar3PuntosAzules() {  
    Poner(Azul)  
    Poner_ADistancia3Al_(Azul, Este)  
    Poner_ADistancia3Al_(Azul, Oeste)  
    Mover3VecesAl_(Oeste) Mover3VecesAl_(Oeste)  
}
```

- b. Ayudar a Nova a contestar la siguiente pregunta: ¿Hay alguna relación entre los parámetros de `Poner_ADistancia3Al_` y `Mover3VecesAl_`?
- c. Dado que Nova no pensó en código general, basándose en el procedimiento corregido recién, generalizarlo para que, dado un color `colorPunto`, dibuje los puntos de ese color. El nuevo procedimiento debe llamarse `Pintar3Puntos_`.

8. Utilizando el procedimiento **Pintar3Puntos_**, construir el procedimiento **PintarArcoIris()** que ponga el tablero de la derecha cuando el tablero inicial es el de la izquierda. ¡A no ser como Nova, y empezar escribiendo el contrato!



(a) Tablero inicial



(b) Tablero final

9. **BIBLIOTECA** La combinación de parámetros con repetición es interesante.
- Generalizar el procedimiento **Mover3VecesAl_** para que la cantidad también pueda ser indicada, haciendo **Mover_VecesAl_(cantidad, dirección)**.
 - ¿Es posible hacer este procedimiento sin usar repetición simple? ¿Por qué?
 - Construir los procedimientos **Poner_DeColor_(cantidad, color)** y **Sacar_DeColor_(cantidad, color)** que trabajen de forma similar al del ítem a. ¿No olvidaste las precondiciones, no?
10. **BIBLIOTECA** Escribir el procedimiento **IrAEsquina_Y_(primeraDirección, segundaDirección)** que dadas dos direcciones posiciona el cabezal en la esquina en dichas direcciones.
11. **EN PAPEL** Escribir un programa que escriba su nombre en el tablero utilizando la primitiva de dibujo **DibujarLíneaHacia_DeLargo_** que se supone primitiva. También puede utilizarse el procedimiento **Mover_Veces** del ejercicio anterior.
- ¿Hace falta que sigamos diciendo que debe primero escribirse el contrato y dividir el trabajo en subtareas? ¿Qué tal una subtarea para cada letra? ¿Será razonable hacer una excepción a la regla de nombrar a los procedimientos con verbos, para que el programa sea mucho más fácil de leer?

```
procedure DibujarLíneaHacia_DeLargo_(dirección, largoDeLaLínea)
```

```
/* PROPÓSITO:
```

```
  * Dibujar una línea de longitud largoDeLaLínea en dirección
    dirección.
```

```
PRECONDICIONES:
```

```
  * La celda actual está vacía y hay al menos
    **largoDeLaLínea** celdas en dirección **dirección**.
```

```
PARÁMETROS:
```

```
  * largoDeLaLínea: la longitud de la línea que se dibuja
  * dirección: la dirección hacia la que se dibuja la línea
```

```
OBSERVACIONES:
```

```
  * Notar por la descripción del propósito, que el cabezal debe
```

```

    terminar en la celda inicial
*/

```

12. **EN PAPEL** ¡Nova volvió a hacer de las suyas! Esta vez, el código que escribió no funciona y hay que corregirlo. Parece que uno de los problemas de corrección está en el alcance de los parámetros... Aprovechar también para mejorar los nombres de los procedimientos y la mezcla de niveles, y también para generalizar el tamaño.

```

procedure (colorDelCuadrado) {
  /* PROPÓSITO: Dibujar un cuadrado de 2x2 de color
    **colorDelCuadrado**

  PRECONDICIONES:
    * Hay al menos 1 celda al Norte y 1 al Este de la actual
  PARÁMETROS:
    * colorDelCuadrado: el color con el que se dibuja el cuadrado
  */

  DibujarLínea2()
  PosicionarseParaSiguienteLínea()
  DibujarLínea2()
  Mover(Sur)
}

```

```

procedure DibujarLíneaDeLongitud2() {
  /* PROPÓSITO: Dibujar una línea de longitud 2 de color
    **colorCuadrado**

  PRECONDICIONES: Hay al menos 1 celda al Este de la actual
  */

  Poner(colorCuadrado) Mover(Este)
  Poner(colorCuadrado) Mover(Oeste)
}

```

13. El bosque, parte 2

Continuaremos representando el bosque que comenzamos en la práctica anterior. Esta vez queremos ser capaces de poner o sacar múltiples elementos de una sola vez.

Importante: para realizar este ejercicio se espera haya realizado la parte 1 en la Práctica 2, si aún no lo hizo, se recomienda volver y realizar el mismo previo a solucionar el ejercicio actual.

- Poner_Semillas(cantidadDeSemillasAPoner)
- Sacar_Semillas(cantidadDeSemillasASacar)
- Poner_Árboles(cantidadDeÁrbolesAPoner)
- Sacar_Árboles(cantidadDeÁrbolesASacar)
- Poner_Nutrientes(cantidadDeNutrientesAPoner)

f. **Sacar_Nutrientes(cantidadDeNutrientesASacar)**