



UNIVERSIDAD
NACIONAL DE
HURLINGHAM

Instituto de Tecnología e Ingeniería / Programación Estructurada

PROCEDIMIENTOS

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.*
- *Algunos de 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. También 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.*
- *Sí un ejercicio indica BIBLIOTECA significa que podrá ser utilizado en el parcial sin definirlo. Es útil mantener registro de dichos procedimientos en su carpeta.*

PROCEDIMIENTOS Y ESTRATEGIA DE SOLUCIÓN

1. Iniciales

Escribir un programa que ponga sus iniciales en el tablero usando bolitas color Verde para los nombres y color Rojo para el apellido, dejando el cabezal en la celda inicial. Para escribir sus iniciales, se debe definir cómo se verán las letras en el tablero.

Reflexionamos: ¿Cuál es el propósito y las precondiciones del programa? ¿Cómo lo puede ayudar la definición de procedimientos? ¿Qué características tienen que tener los procedimientos para favorecer su adecuada combinación? ¿Cuál es el propósito y las precondiciones de los procedimientos realizados?

2. Por arriba

Realizar cada uno de los siguientes puntos, en el orden dado (siguiendo una metodología top-down).

a) Escribir un procedimiento **DibujarRectánguloRojoYNegroDe5x3()** cuyo contrato es el siguiente:

```
procedure DibujarRectánguloRojoYNegroDe5x3()
```

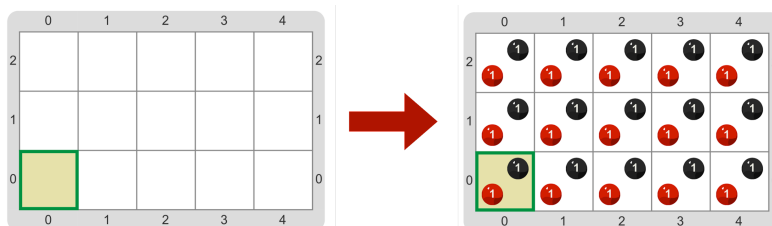
PROPÓSITO

Poner un rectángulo sólido de 5 celdas de ancho y 3 celdas de alto, en la que cada celda tenga una bolita de color Rojo y una de color Negro. El cabezal comienza y termina en el vértice inferior izquierda del mismo.

PRECONDICIONES

Hay al menos 4 celdas al Este y 2 celdas al Norte de la celda actual.

La metodología a seguir que se debe aplicar es la siguiente. En primer lugar, pensar una estrategia; se sugiere pensar una estrategia que involucre poner líneas de 5 celdas de ancho. Luego se debe definir primero el contrato de un procedimiento llamado **DibujarLíneaRojaYNegraDeTamaño5()** que exprese la subtask sugerida. Para completar el ejercicio, se debe utilizar este procedimiento auxiliar en la codificación del procedimiento pedido, **DibujarRectánguloRojoYNegroDe5x3()**. El código del procedimiento auxiliar no es parte de este inciso, sino del siguiente.



El rectángulo es sólido. Es decir, la transformación esperada es similar a la que se muestra en la imagen.

b) Escribir el procedimiento **DibujarLíneaRojaYNegraDeTamaño5()** que quedó pendiente del ítem anterior.

Para ello, se debe seguir la misma metodología que en el caso anterior: en primer lugar pensar una estrategia, luego definir los contratos de los procedimientos que expresan las subtasks, y por último codificar el procedimiento pedido usando dichas subtasks (sin su código). Se sugiere que la estrategia involucre a subtasks llamadas **PonerUnaNegraYUnaRoja()** y **Mover4VecesAlOeste()**.

c) Escribir los procedimientos **PonerUnaNegraYUnaRoja()** y **Mover4VecesAlOeste()** que quedaron pendientes del ejercicio anterior. En este caso, los procedimientos se pueden expresar fácilmente utilizando comandos primitivos, por lo que no es necesario comenzar pensando en posibles subtasks.

3. Por abajo

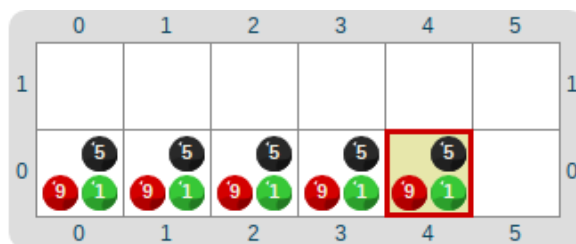
En el ejercicio anterior se trabajó con una metodología conocida como top-down: se comienza desde el problema, y se lo va descomponiendo en tareas cada vez más pequeñas, hasta llegar al uso de comandos primitivos. En este ejercicio vamos a proponer la estrategia inversa, bottom-up: primero se escriben los procedimientos elementales, y luego se van armando procedimientos cada vez más complejos, hasta construir uno que resuelva el problema original.

Escribir los siguientes procedimientos en el orden dado. Recuerde escribir los contratos correspondientes.

- BIBLIOTECA. PonerUnaDeCada()** que ponga una bolita de cada color en la celda actual.
- Mover3VecesAlOeste()** que mueva el cabezal tres celdas hacia el Oeste.
- DibujarLíneaMulticolorDeLargo4()** que ponga una línea de cuatro celdas hacia el Este en la que cada celda tenga una bolita de cada color. El cabezal debe quedar en la celda inicial. Para ello, debe reutilizar los procedimientos **PonerUnaDeCada()** y **Mover3VecesAlOeste()** (realizados en ejercicios anteriores).
- DibujarCuadradoMulticolorDeLado4()** que ponga un cuadrado sólido de 4×4 celdas en la que cada celda tenga una bolita de cada color. El cabezal debe quedar en la celda inicial. Para ello, debe reutilizar el procedimiento **DibujarLíneaMulticolorDeLargo4()**.
- Reflexionar sobre las diferencias entre ambas metodologías. ¿Cuál de las dos metodologías es más sencilla si no tenemos ninguna pista de cómo resolver el problema? ¿Qué se requiere para poder utilizar una estrategia bottom-up? ¿Cuál metodología es más utilizada y por qué?

4. Guarda de azulejos

Construir un procedimiento **PonerGuardaDe5Azulejos()**, que arme una "guarda" de 5 azulejos (como las que decoran las paredes). Cada azulejo está conformado por 1 bolita verde, 5 negras y 9 rojas.



Recordar seguir una metodología adecuada para la construcción del código: COMENZAR por escribir el contrato completo del procedimiento, luego pensar las subtareas necesarias y darles un nombre adecuado, escribir el contrato de las subtareas, LUEGO el código del procedimiento pedido en términos de las subtareas, y FINALMENTE, realizar el código de las subtareas siguiendo la misma metodología (que denominamos metodología de construcción de programas top-down).

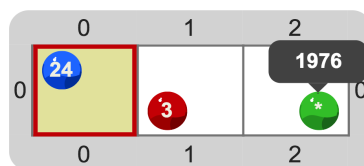
5. Día de la Memoria

Utilizando bolitas pueden representarse diversos elementos; un ejemplo de esto es la posibilidad de representar una fecha. Una fecha que los argentinos deberíamos recordar, para no repetirla jamás, es el 24 de marzo de 1976, hoy constituido como Día de la Memoria por la Verdad y la Justicia en Argentina.

Definir el procedimiento **RegistrarElDíaDeLaMemoria()** que:

- en la celda actual, ponga 24 bolitas Azules, que representan el día,
- en la celda lindante al Este de la actual, ponga 3 bolitas Rojas, que representan el mes, y

- en la celda lindante al Este de la anterior, ponga 1976 bolitas verdes, representando el año.



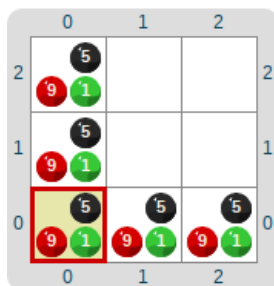
Recordar que la solución completa incluye la documentación: propósito, precondition y parámetros, los cuales deben ser escritos ANTES de escribir el código.

La solución se puede realizar íntegramente utilizando los conceptos vistos en la materia hasta el momento, y no se requieren herramientas adicionales. No debe utilizar conceptos aún no vistos en la teoría para solucionar el problema.

USANDO Y REUTILIZANDO

6. Guarda de azulejos en L

Hacer un procedimiento **PonerGuardaEnL()**, que arme una guarda en L como muestra la figura, dejando el cabezal en la posición inicial.



¿Pensaste en reutilizar el procedimiento definido antes, o empezaste a escribirlo de nuevo? Dado que ahora se cuenta con la subtaska definida en el ejercicio anterior, la metodología top-down se puede enriquecer con la reutilización de procedimientos ya realizados; esto puede hacer que ciertas divisiones en subtaskas, que permiten esa reutilización, sean más sencillas que otras.

7. Las pirámides de Gobstones

EN PAPEL. Escribir un programa para cumplir con los propósitos de cada uno de los ítems que se indican más adelante. Para ello, deben utilizarse los procedimientos indicados a continuación.

procedure DibujarBase()

Prop.

Dibuja una base de pirámide de 5 celdas de lado

Prec.

La celda actual debe estar vacía

Debe haber cuatro celdas vacías al Este del cabezal

procedure DibujarMedio()

Prop.

Dibuja un sector del medio de pirámide de 3 celdas de lado

Prec.

La celda actual debe estar vacía

Debe haber dos celdas vacías al Este del cabezal

procedure DibujarPunta()

Prop.

Dibuja una punta de pirámide

Prec.

La celda actual debe estar vacía

¡Recordar! Para cada ítem que sigue debe comenzarse redactando el contrato correspondiente, y de ser necesario, se deben definir subtarefas para construir su solución, expresándose mediante procedimientos (y en ese caso se deben escribir primero los contratos de los mismos y utilizarlos ANTES de dar su código).

- a) Una pirámide.
- b) Una pirámide invertida.
- c) Una pirámide estirada a lo alto (con dos segmentos de cada uno).
- d) Una pirámide gigante (con 11 bloques de base y 6 bloques de altura, donde cada segmento es dos celdas más chico que el que tiene debajo). Para esto deben reutilizarse los procedimientos implementados en los puntos a. y b. PISTA: una pirámide gigante se puede conseguir con 4 pirámides comunes, si se las ubica con cuidado.

MODELANDO DOMINIOS**8. El bosque, parte 1**

En este ejercicio, se usará el tablero para representar un bosque. Cada celda representa a una parcela. Cada bolita verde representa un árbol. Cada bolita roja representa una semilla. Una bolita negra representa una bomba. Una bolita azul representa una unidad de nutrientes.

Escribir los siguientes procedimientos de representación, que hacen lo que su nombre indica. Todos trabajan siempre sobre la celda actual.

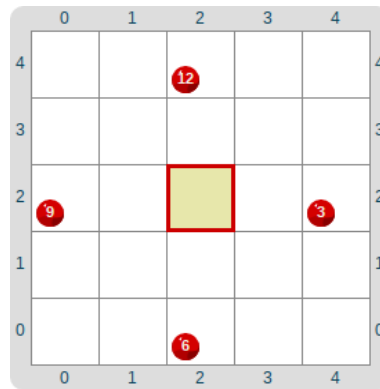
- **PonerUnaSemilla()** ▪ **PonerUnÁrbol()** ▪ **PonerUnaBomba()** ▪ **PonerUnNutriente()**
- **SacarUnaSemilla()** ▪ **SacarUnÁrbol()** ▪ **SacarUnaBomba()** ▪ **SacarUnNutriente()**

9. Reloj Analógico Simplificado

Dibujar un reloj analógico de agujas en un tablero cuadriculado puede ser un desafío. Una simplificación posible sería representar solamente algunos de los números que aparecen en el mismo: el 12 arriba, el 3 a la derecha, el 9 a la izquierda y el 6 abajo.

Construir un procedimiento **DibujarRelojAnalógicoSimplificado()**, que ponga los números del reloj tal como se indicó, alrededor del casillero actual. El tamaño del reloj será de 2 celdas de "radio" (suponiendo que miramos al reloj como un círculo).

Utilizar el comando **DibujarRelojAnalógicoSimplificado()** en un tablero inicial vacío de 5x5 con la celda inicial en el centro del mismo, es el siguiente:



¡Recordar! Escribir el contrato en primer lugar, y en caso de utilizar división en subtareas, en seguir la metodología top-down para las mismas (primero su nombre y su contrato, usarlas, y recién luego definir las con la misma metodología).

10. La llegada de Nova

EN PAPEL. En el equipo de programación empezó a trabajar un programador que no tiene formación profesional, y al que todos apodaron Nova . En el código que escribió Nova se encontró un procedimiento que no aplica la técnica de división en subtareas, y tampoco tiene contrato. Para poder integrar el código con el resto, en este equipo se requiere que los programas sigan buenas prácticas de programación. Por eso, se pide corregir este código para que cumpla con las mismas.

```
procedure ConstruirEscaleraAzulDe4Escalones() {
  Poner(Azul) Mover(Este) Poner(Azul) Mover(Norte)
  Poner(Azul) Mover(Este) Poner(Azul) Mover(Norte)
  Poner(Azul) Mover(Este) Poner(Azul) Mover(Norte)
  Poner(Azul) Mover(Este) Poner(Azul)
}
```

Comenzar por redactar el contrato y la precondition correspondientes.

- Luego aplicar la técnica de división en subtareas expresadas con procedimientos. Dado que no se trata de hacer el código de nuevo, se propone identificar partes de código que se repitan, darles un buen nombre y extraer esa parte de la estrategia en procedimientos auxiliares. Por supuesto, cada procedimiento auxiliar debe tener un nombre adecuado, y debe documentarse su propósito y precondition.
- ¿Puede apreciar las ventajas de contar con el contrato y de expresar la estrategia como parte del código en forma de procedimientos? Hay más de una forma de dividir en subtareas. Comparar la solución propuesta con la de otros compañeros, y discutir las ventajas y desventajas de la dada con las que resulten diferentes.