Algoritmos y Estructuras de Datos

Práctica 1

Instituto de Educación

UNIVERSIDAD NACIONAL DE HURLINGHAM

1. Programas y contratos

1) **Reemplazando bolitas:** Escribir un programa que reemplace una bolita de color roja con otra de color verde en la celda actual. Pruebe el programa en la computadora, modificando el tablero inicial de forma que el programa funcione satisfactoriamente.

2) **Moviendo bolitas:** Escribir un programa que mueva una bolita de color negro de la celda actual a la celda vecina al este, dejando el cabezal en la celda lindante al este.

3) **Poniendo en vecinas:** Escribir un programa que ponga una bolita de color azul en la celda vecina al norte de la actual.

4) **Analizando propósitos:** Dado el siguiente código

```
1 program {
2   Poner(Verde)
3   Sacar(Verde)
4   Poner(Azul)
5   Poner(Rojo)
6 }
```

Diversos estudiantes realizaron propuestas para redactar su propósito, y también un profesor realizó explicaciones sobre cada una de estas propuesta. Asociar cada propuesta de propósito para el mismo (indicadas con las letras A, B, etc.) con la explicación que resulta correcta para dicha propuesta (indicadas con los números 1, 2, etc.)

A) Poner una bolita azul y luego una roja en la celda actual.	(1) No es un propósito, sino una descripción del funcionamiento. Para ser un propósito no debe preocuparse de los estados intermedios, solamente de la transformación final.
B) Agregar una bolita azul y una roja.	(2) Es un propósito incompleto ya que no establece los colores de las bolitas que se agregan. El propósito debe establecer con precisión la transformación esperada.
C) Pone una bolita verde y luego la saca, para a continuación poner una bolita azul y una roja.	(3) Es una enunciación correcta del propósito. El orden en que se agregan las bolitas es irrelevante, siempre que la celda actual finalice con una más de cada uno de los colores indicados.

D) Agregar una bolita roja y una bolita azul en la celda actual.	(4) Es un propósito incompleto, ya que no establece dónde se agregan las bolitas en cuestión. El propósito debe establecer con precisión la transformación esperada.
E) Agregar dos bolitas en la celda actual.	(5) Es una forma incorrecta de indicar la transformación esperada. Utiliza un lenguaje que sugiere un pensamiento operacional (o sea, centrado en las acciones individuales antes que en la transformación esperada.)

5) **Cuadrados verdes**. Escribir los siguientes programas:

   a) Uno que ponga un cuadrado de tamaño 3 con bolitas de color verde, con centro en la celda inicial (dejando el cabezal en dicha celda al finalizar).

      - ¿Qué ocurre si el tablero ya tenía bolitas verdes? ¿Es esto un problema, o el efecto obtenido es acorde al propósito?
      - ¿Qué ocurriría si desde la celda inicial no hay espacio para colocar las bolitas necesarias para que se cumpla el propósito?

   b) Uno que saque un cuadrado de tamaño 3 con bolitas de color verde (saca una bolita de cada celda), siendo la celda inicial el centro del cuadrado, dejando el cabezal en dicha celda al finalizar.

      - ¿Qué sucede si no hay al menos una bolita verde en cada una de las celdas necesarias? ¿Y si el tablero no tiene el tamaño adecuado?

6) **EN PAPEL**. Discutir en clase cuáles serían las precondiciones para programas cuyo propósito sea:

   a) Poner 1000 bolitas color Azul en la celda actual.

   b) Poner una bolita color Rojo en la celda lindante al Este de la celda actual y sacar una bolita Azul de la celda lindante al Oeste de la celda actual.

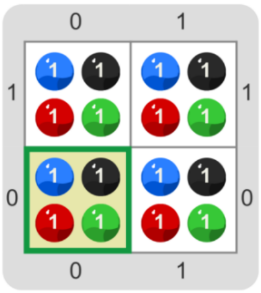
   c) Poner un rectángulo de bolitas Negras cuyo tamaño sea 3 filas y 5 columnas, centrado en la celda actual.

7) **Sacando un cuadrado**. Escriba un programa que saque del tablero un cuadrado multicolor de dos celdas de lado, donde la celda actual representa el vértice inferior izquierdo del mismo. Recuerde escribir primero el contrato del programa, y luego el código. Considere las siguientes preguntas como guía para escribir su programa:

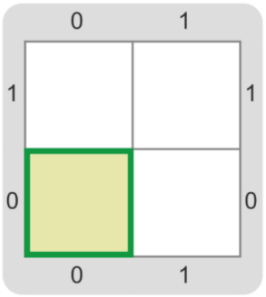
   a) ¿Que hace el programa? (Determina el propósito del programa)

   b) ¿Cuándo funciona tal cual se espera? (Determina la precondición del programa)

c) ¿Cómo lo hace? (Determina el código del programa)



Tablero inicial



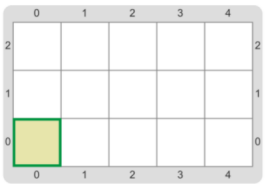
Tablero final

8) **Top-down**. Realizar cada uno de los siguientes puntos, en el orden dado (metodología top-down).

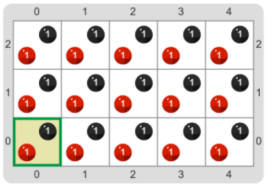
a) Escribir un procedimiento **DibujarRectánguloRojoYNegroDe5x3()** cuyo contrato es el siguiente:

> 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 subtarea 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.



Tablero inicial



Tablero final

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 subtareas, y por

último codificar el procedimiento pedido usando dichas subtareas (sin su código). Se sugiere que la estrategia involucre a subtareas 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 subtareas.

9) **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()**
- **SacarUnaSemilla()**
- **PonerUnÁrbol()**
- **SacarUnÁrbol()**
- **PonerUnaBomba()**
- **SacarUnaBomba()**
- **PonerUnNutriente()**
- **SacarUnNutriente()**

2. Repeticiones y parámetros

1) **Dibujando un rectangulo con repeticiones**. Escribir un procedimiento **DibujarRectánguloRojoYNegroDe5x3()** que dibuje un rectángulo sólido de 5 celdas de largo por 3 de alto, símilar al realizado en 8)–a), pero esta vez, utilice repetición para solucionar el problema.

2) **Pintando el tablero** Escribir un procedimiento **PintarElTableroDeAzul()** que, asumiendo que el tablero tiene 10 celdas de largo y 7 celdas de alto, pinte absolutamente todo el tablero con bolitas azules, dejando exactamente una bolita azul en cada celda.

   a) ¿Cuál es la precondición del procedimiento?

   b) ¿Se le ocurre otra estrategia para resolver el problema?

   Importante Recuerde que la estrategia de solución debe quedar clara a partir de la lectura del código. Use subtareas con nombres apropiados para dicho objetivo.

3) **Moviendo tres veces a donde quieras**. Escribir un procedimiento **Mover3VecesAl_(direcciónAMover)** que dada una dirección direcciónAMover mueva el cabezal tres posiciones en dicha dirección.

   a) ¿Qué hay que hacer ahora para mover el cabezal tres veces al Norte?

   b) ¿Qué beneficios trae el uso de parámetros?

c) ¿Cuántos procedimientos puedo ahorrarme haciendo un único procedimiento con un parámetros?

¡Recordar! No olvidar escribir el contrato del procedimiento ANTES de realizar el código (y que los parámetros son parte del mismo); también discutir la precondición escrita con sus compañeros para verificar que la misma es adecuada y correcta.

4) **Y 6 de lo que quieras**. Escribir un procedimiento **Poner6DeColor_(colorAPoner)** que dado un color colorAPoner ponga 6 bolitas del color dado.

5) **Poner de a muchas**

BIBLIOTECA. Escribir un procedimiento **Poner_DeColor_(cantidadAPoner, colorAPoner)** que dado un número cantidadAPoner y un color colorAPoner, ponga tantas bolitas como se indica del color dado de la celda actual.

6) **Moviendo tantas veces como quieras a donde quieras**

BIBLIOTECA. Escribir **Mover_VecesAl_(cantidadAMover, direcciónAMover)**, un procedimiento que dado un número cantidadAMover y una dirección direcciónAMover mueva el cabezal tantas veces como la dada en dicha dirección.

7) **Sacar de a muchas**

BIBLIOTECA. Escribir un procedimiento **Sacar_DeColor_(cantidadASacar, colorASacar)** que dado un número cantidadASacar y un color colorASacar, saque tantas bolitas como se indica del color dado de la celda actual.

8) **Moviendo y poniendo**. Escribir un procedimiento **Poner_Al_(colorAPoner, direcciónDondePoner)** que dado un color colorAPoner y una dirección direcciónDondePoner, 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 usaran 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?

9) **Reemplazando colores**. Escribir **ReemplazarUnaDe_Por_(colorAReemplazar, colorPorElCualReemplazar)**, un procedimiento que dado un primer color colorAReemplazar y un segundo color colorPorElCualReemplazar, reemplaza una bolita del primer color por una del segundo color (En la celda actual).

10) **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 partes anteriores de este dominio, si aún no lo hizo, se recomienda volver y realizar el mismo previo a solucionar el ejercicio actual.

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

11) **¡A la batalla!, parte 1**

    Suponiendo que se está programando un juego donde en las celdas del tablero se representan soldados (los aliados con una bolita de color Negro y los enemigos con una bolita de color Rojo por cada soldado), escribir los siguientes procedimientos:

    a) **EnviarAliadosParaDuplicarEnemigos()**, que agrega soldados aliados en la celda actual en cantidad suficiente para que haya el doble de aliados que de soldados enemigos.

    b) **PelearLaBatalla()**, que simula una batalla, suponiendo que hay suficiente cantidad de soldados aliados como para ganar la batalla. Durante una batalla, 2 soldados enemigos pelean contra 3 soldados aliados y todos mueren. Por ejemplo, si hay 6 enemigos y 10 aliados, mueren los 6 enemigos y 9 de los aliados; si hay 10 enemigos y 21 aliados, mueren los 10 enemigos y 15 soldados aliados.

       PISTA: ¿Qué cuenta hay que hacer para saber cuántos soldados aliados morirán?

12) **Sacando todas las de un color**

    BIBLIOTECA. Escribir un procedimiento **SacarTodasLasDeColor_(colorASacar)**, que quite de la celda actual todas las bolitas del color indicado por el parámetro.

    PISTA: Considerar utilizar el procedimiento **Sacar_DeColor_**, definido anteriormente. ¿Qué argumentos se le deberían pasar?

13) **¿Y si vaciamos la celda?**

    BIBLIOTECA. Escribir un procedimiento VaciarCelda() que quite de la celda actual todas las bolitas de todos los colores, dejando la celda vacía.

3. Alternativa condicional

1) **Sí se puede, sí se puede...**

   Escribir los siguientes procedimientos, recordando no mezclar niveles de abstracción del problema, para lo cual puede ser necesario definir otros procedimientos y/o funciones.

   a) **SacarUnaFicha_SiSePuede(colorDeLaFicha)** que, dado el colorDeLaFicha que debe sacarse, saque una ficha siempre y cuando la misma esté en la celda. Si no hubiera fichas del color dado, el procedimiento no hace nada. Si hubiera varias fichas, solo debe sacar una.

      OBSERVACIÓN: cada ficha se representa con una bolita del color correspondiente.

   b) **DesempatarParaElLocal_Contra_(colorDelLocal,colorDelVisitante)** que, dados los colores de dos jugadores cuyos puntos se representan mediante la cantidad de bolitas del color del jugador, otorgue un punto al jugador con color olorDelLocal solamente en el caso en que la celda actual contiene la misma cantidad de bolitas de ambos colores.

   c) **ExpandirBacteriaDeLaColonia()**, que siempre que en la celda actual haya un cultivo de bacterias y haya suficientes nutrientes, agregue exactamente una bacteria más y consuma nutrientes, a razón de dos nutrientes por bacteria expandida; si no hay bacterias o no hay suficientes nutrientes, no hace nada. Las bacterias se representan con bolitas Verdes y los nutrientes con bolitas Rojas.

   d) **PonerFlecha_AlNorteSiCorresponde(colorDeLaFlecha)**, que dado un color para representar flechas, ponga una flecha al Norte si existe espacio para moverse en esa dirección. Las flechas serán representadas con una bolita del color dado.

2) **Hacer solo si...**

   La combinación de parámetros y expresiones booleanas es interesante.

   a) **BIBLIOTECA.** Escribir un procedimiento **Poner_Si_(color, condición)** que dado un color y un valor de verdad llamado condición, ponga en la celda actual una bolita del color dado si el valor de verdad de la condición es verdadero, y no lo ponga si no.

      EJEMPLO: **Poner_Si_(Rojo, nroBolitas(Rojo) == 2)** solamente pone una bolita roja cuando hay exactamente dos rojas en la celda actual.

   b) **BIBLIOTECA.** Escribir el procedimiento **Sacar_Si_(color, condición)** que actúa de forma similar al anterior, pero ahora sacando bolitas si la condición se cumple.

c) **BIBLIOTECA.** Escribir el procedimiento **Mover_Si_(dirección, condición)** que actúa de forma similar a los anteriores, pero ahora moviendo solo sí se cumple la condición dada.

d) ¿Que beneficios trae tener los procedimientos **Sacar_Si_** y **Poner_Si_** contra utilizar if en cada caso?