

## Práctica 7

### Introducción a Registros

Introducción a la Programación  
1<sup>er</sup> Semestre de 2014

#### Ejercicio 1

Declarar el registro `Coord` que representa una coordenada del tablero con los campos `fila` y `columna`, donde la coordenada del origen es `Coord(fila  $\leftarrow$  0, columna  $\leftarrow$  0)`. Implementar las siguientes operaciones:

- (a) `esMayorCoord` que, dadas dos coordenadas, determina si la primer coordenada aparece después de la segunda en un recorrido que va primero hacia el norte y luego hacia el este. Es decir, `esMayorCoord(c1, c2)` denota `True` cuando `columna(c1) > columna(c2)` o cuando `columna(c1) == columna(c2)` pero `fila(c1) > fila(c2)`.
- (b) `coordOrigen` que denote la coordenada del origen, i.e., `Coord(fila  $\leftarrow$  0, columna  $\leftarrow$  0)`.
- (c) `reflejo` que, dada una coordenada, denote la coordenada que se obtiene de intercambiar las filas y la columna. Es decir, `reflejo(c)` es la coordenada `Coord(fila  $\leftarrow$  columna(c), columna  $\leftarrow$  fila(c))`.
- (d) `sobreLaDiagonal` que, dada una coordenada, indique si la misma se encuentra sobre la diagonal que va del origen hasta la ultima fila. En otras palabras, denota `True` cuando la fila de la coordenada es superior a la columna.
- (e) `IrACoord` que, dada una coordenada, posicione el cabezal en la celda correspondiente del tablero.
- (f) `coordActual` que retorne la coordenada en la que se encuentra el cabezal.
- (g) `ultimaCoord` que denote la ultima coordenada del tablero en un recorrido hacia el noreste.
- (h) `siguienteCoord` que, dada una coordenada, denote la siguiente coordenada en un recorrido que va primero hacia el norte y luego hacia el este.
- (i) `siguienteCoordSobreLaDiagonal` que, dada una coordenada, denote la siguiente coordenada que se encuentre sobre la diagonal. Por ejemplo, en un tablero de  $8 \times 8$ , `siguienteCoordSobreLaDiagonal(Coord(fila  $\leftarrow$  7, columna  $\leftarrow$  3))` denota `Coord(fila  $\leftarrow$  5, columna  $\leftarrow$  4)`. ¿Cuál es la precondition de la función?
- (j) `ultimaCoordSobreLaDiagonal` que denota la última coordenada sobre la diagonal. Por ejemplo, en un tablero de  $8 \times 8$ , denota `Coord(fila  $\leftarrow$  7, columna  $\leftarrow$  6)`. Como precondition, debe existir al menos una celda sobre la diagonal o, en otras palabras, el tablero debe tener al menos dos filas.

- (k) `triangularInferior` que denota `True` cuando las celdas sobre la diagonal están vacías (i.e., no tienen bolitas). Sugerencia: estructurar la solución usando un recorrido de coordenadas sobre la diagonal.

## Ejercicio 2

Declarar el registro `Celda` con campos `azul`, `negro`, `rojo` y `verde` que represente una celda del tablero contando la cantidad de bolitas de cada color que hay en la celda. ¿Es válido el valor `Celda(azul ← 1, negro ← 3, rojo ← 7, verde ← -3)`? Escribir el invariante de representación, e implementar las siguientes operaciones

- (a) `celdaActual` que denote la registro correspondiente a la celda actual.
- (b) `celdaEnCoord` que, dada una coordenada (modelada con un registro `Coord`), denote el registro correspondiente a la celda en dicha coordenada.
- (c) `VolcarCelda` que, dado un registro celda, deje tantas bolitas de cada color como indica el registro.
- (d) `IntercambiarCeldas` que, dadas dos coordenadas, intercambie las bolitas de las celdas en dichas coordenadas.
- (e) `ReflejarCelda` que, dada una coordenada, intercambie el contenido de la celda actual con aquella que se encuentra en su coordenada reflejo.
- (f) `ReflejarTablero` que refleje todas las celdas de un tablero cuadrado (i.e., misma cantidad de filas que columnas). Para ello, conviene hacer un recorrido del tablero en que se reflejen únicamente las celdas correspondientes a posiciones sobre la diagonal.
- (g) `simetrico` que indique si cada celda sobre la diagonal contienen la misma cantidad de bolitas que la celda en su coordenada reflejo.

## Ejercicio 3

Declarar el registro `Fecha` que representa una fecha, con día, mes y año, usando un registro que contenga esa información. Incluir en la documentación el invariante de representación que indique qué valores son válidos. Realizar los siguientes procedimientos que devuelven valores para manipular fechas:

- (a) `esMenorFecha`, que recibe dos fechas y devuelve un booleano indicando si la primera es anterior a la segunda.
- (b) `anhosEntre`, que recibe dos fechas y devuelve la diferencia en años. Sólo contar los años completos, es decir entre el 2 de febrero de 2008 y el 15 de enero de 2009 hay 0 años.
- (c) `bisiesto`, que dada una fecha, determina si el año correspondiente es bisiesto<sup>1</sup>.
- (d) `mañana`, que dada una fecha denote la fecha siguiente. Tener en cuenta los cambios de mes y año, y los años bisiestos.

---

<sup>1</sup>Un año es bisiesto cuando o bien es divisible por 400 o bien es divisible por cuatro y no por 100.

- (e) **esFinDeMes**, que dada una fecha denota si la misma es el último día del mes.
- (f) **diferenciaEnDias**, que recibe dos fechas y devuelve la cantidad de días que pasaron de la primera fecha a la segunda. Notar que si la primera fecha es posterior a la segunda, el resultado es negativo. Recordar los años bisiestos.
- (g) **diaDeLaSemana**, que dada una fecha, retorne el número del día de la semana, teniendo en cuenta que el día número 1 es domingo, el día número 2 es lunes, etc. Ayuda: observe que el 1/1/2011 fue un domingo (día número 1). Luego, basta contar la cantidad de días desde esa fecha hasta la pasada como parámetro, tomar el resto de dividir por 7 y sumarle 1.
- (h) [Opcional] **PonerMesCalendario** que, dado un número de mes y año, ponga en un rectángulo de  $6 \times 7$  el dibujo de un calendario. Cada celda tiene tantas bolitas azules como el día que representa. La primera columna del rectángulo corresponde al día domingo, la segunda al lunes, etc. En la primera fila se dibujan las fechas de la primera semana, en la segunda la de la segunda, etc. Aquellas posiciones del rectángulo que no representen fechas deben dejarse vacías.
- (i) [Opcional] **PonerAñoCalendario** que, dado un número de año, ponga 12 rectángulos de  $6 \times 7$  que representan los distintos meses calendarios. Organizarlo en una grilla de  $4 \times 3$  rectángulos (notar que el tablero tiene que tener al menos  $24 \times 21$  celdas).

#### Ejercicio 4

Declarar el registro **Persona**, que contenga el número de DNI, el domicilio (un número), un booleano indicando si la persona es de sexo masculino, y la fecha de nacimiento (usando el registro **Fecha** declarado anteriormente). Implementar las siguientes operaciones:

- (a) **convivientes** que, dadas dos personas, indique si comparten domicilio.
- (b) **nacimiento** que, dada una persona **madre**, un booleano indicando el sexo y una fecha de nacimiento, cree una nueva persona que conviva con la **madre**.
- (c) **edadEnFecha**, recibe una persona y una fecha y devuelve la edad que la persona va a tener en esa fecha.
- (d) **igualEdadEnFecha**, recibe dos personas y una fecha y retorna si las dos personas tienen la misma edad en la fecha indicada.
- (e) **menorPersona**, recibe dos personas y retorne si la primera es menor en edad a la segunda.
- (f) **cambioDomicilio**, recibe una persona y un nuevo domicilio y denota la persona con el domicilio cambiado.
- (g) **cambioSexo**, recibe una persona y le cambia su sexo en el documento.
- (h) **cumpleaños**, recibe una persona y un entero, y denota la fecha en la que la persona cumple tantos años como indica el parámetro. Las personas que nacieron el 29 de febrero lo festejan el 28 en los años no bisiestos.

### Ejercicio 5

Declarar el registro **Alumno**, que contenga los datos básicos del alumno (modelado con el registro **Persona**), su número de legajo, la cantidad de materias que rindió, la cantidad que aprobó, y la fecha de ingreso (modelado con el registro **Fecha**). Implementar las siguientes operaciones:

- (a) **alumnoIngresante**, recibe una persona y un número de legajo y crea un alumno que no curso ninguna materia.
- (b) **materiasDesaprobadas**, que retorna el número de materias desaprobadas por un alumno.
- (c) **cambioLegajo**, que dado un alumno y un número de legajo retorna un nuevo alumno que resulta de aquel pasado como argumento modificando su legajo.
- (d) **cambioDomicilioAlumno**, recibe un alumno y una nuevo domicilio y retorna un nuevo alumno resultante de aquel pasado como argumento pero con el nuevo domicilio.
- (e) **alumnoPrecoz**, recibe un alumno e indica si empezó sus estudios antes de cumplir los 15 años.

### Ejercicio 6

Identificar el invariante de representación del tipo **Alumno** del ejercicio anterior. Considerar que el registro persona es correcto.