



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE CIENCIAS

**Introducción a las Ciencias de la Computación**

**Práctica 2. Introducción a la programación con Karel**

PROFESOR

**Manuel Alcántara Juárez**

AYUDANTES

**Ricchy Alain Pérez Chevanier**

**Víctor Emiliano Cruz Hernández**

28 de agosto de 2024

---

# Índice

<b>1. Karel</b>	<b>3</b>
<b>2. Ejercicios</b>	<b>3</b>
2.1. Problema 1: CollectNewsPaper . . . . .	3
2.2. Problema 2: DrawDiagonals . . . . .	4
2.3. Problema 3: MidpointFindingKarel . . . . .	5
2.4. Problema 4: MultiplicationKarel . . . . .	7
<b>3. Consideraciones adicionales</b>	<b>8</b>
<b>4. Entrega</b>	<b>8</b>

## 1. Karel

Para esta práctica tendrás que resolver cuatro problemas que involucran al robot Karel:

<https://omegaup.com/karel.js/#>

Nosotros vamos a utilizar Java para el código que haremos con Karel. Podemos cambiar el lenguaje como se muestra en la imagen:

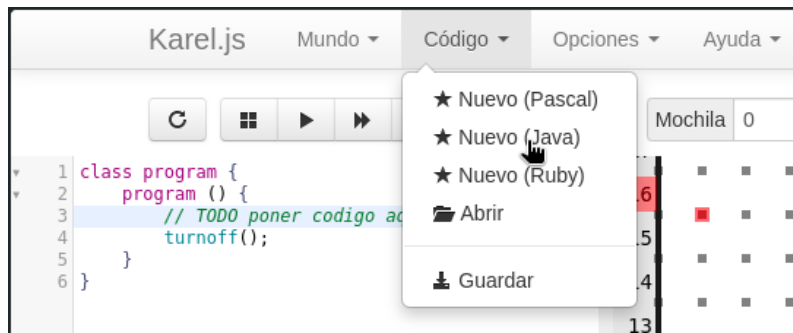


Figura 1: Cambiamos a Java

En general, la sintaxis de Karel así como las instrucciones vamos a verlas en el laboratorio. Aún así, la documentación puede encontrarse en la siguiente guía:

<https://omegaup.com/karel.js/manual/Karel.html>

## 2. Ejercicios

En esta sección se encuentran los problemas que se deben de resolver. En cada uno de ellos hay que tener especial atención en la tarea que hay que realizar, las características del mundo y de Karel que se pueden asumir e igualmente de importante **las que no**.

Además, se incluye una sección con escenarios de prueba pensados para que su solución sea lo más general posible (dentro del marco del problema). Cada una de sus soluciones debe de comportarse correctamente de acuerdo a dicho mundo y o situación donde se presente Karel.

### 2.1. Problema 1: CollectNewsPaper

Supón que Karel se encuentra instalado cómodamente en su casa, representada por el cuadrado interno en el centro de la siguiente figura:

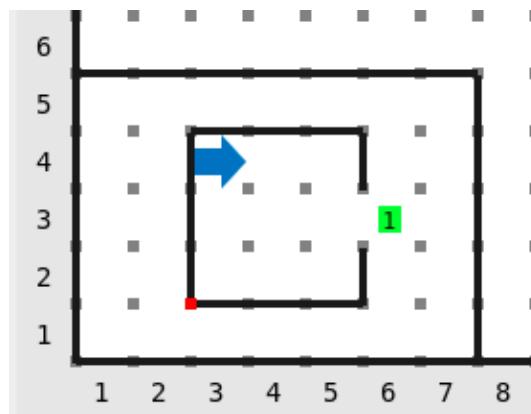


Figura 2: Mundo para Karel del problema 1.

La tarea que debes llevar a cabo es indicarle a Karel que recoja el periódico, que viene siendo el zumbador justo afuera de su casa, y, por último, debe regresar a su posición inicial dentro de su casa.

Puedes asumir lo siguiente:

- Karel comienza en la esquina superior izquierda de su casa.
- El mundo es justo como se muestra en la Figura anterior.
- La casa tiene ese tamaño precisamente
- La ubicación de la puerta es justo la que se muestra
- El zumbado esta inmediatamente afuera de la casa

Lo que tienes que hacer es escribir la secuencia de comandos necesarios para que Karel:

1. Se mueva a la ubicación donde está el periódico;
2. Lo tome; y
3. Regrese a su posición inicial.

Aunque este programa sea muy sencillo, es importante que practiques la descomposición de tu programa en pedazos más pequeños y simples. En tu solución, incluye un método privado para cada uno de los tres pasos descritos arriba.

## 2.2. Problema 2: DrawDiagonals

Karel se encuentra inicialmente en la esquina inferior izquierda, dando la cara hacia el este, dentro de un mundo que consta únicamente de las paredes exteriores del mundo, de manera que en el interior no hay contenido alguno, como se ilustra:

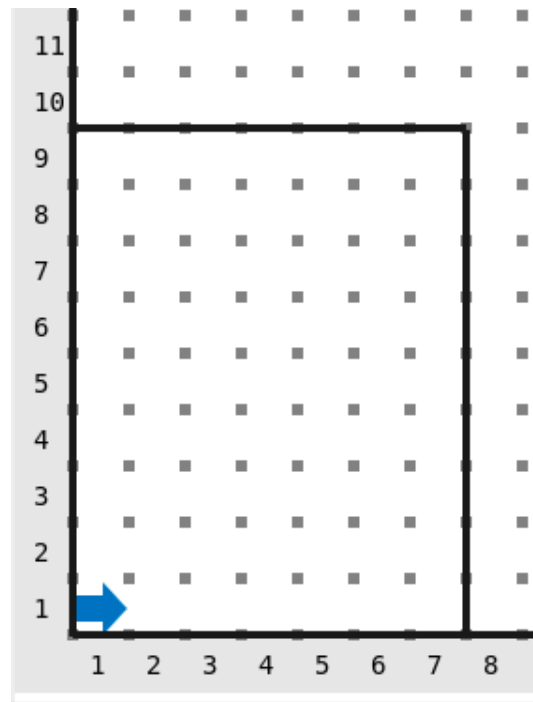


Figura 3: Mundo del problema 2. El mundo de Karel consta de un recuadro 7x9 con el interior vacío.

Tu tarea es instruir a Karel para que dibuje las diagonales del mundo en el que se encuentra, es decir, Karel debe poner un zumbador en la esquina de calle 1 a y avenida 1 a , calle 2 a y avenida 2 a , así hasta calle 7 a y avenida 7 a , con lo anterior obtenemos una diagonal. Además, Karel debe poner zumbadores en la otra diagonal,

por lo que debe haber un zumbador en la esquina de calle 1 a y avenida 7 a , calle 2 a y avenida 6 a , así hasta calle 7 y avenida 1.

Por ejemplo, para el mundo que se muestra en la imagen anterior, las diagonales deben de formarse como se muestra:

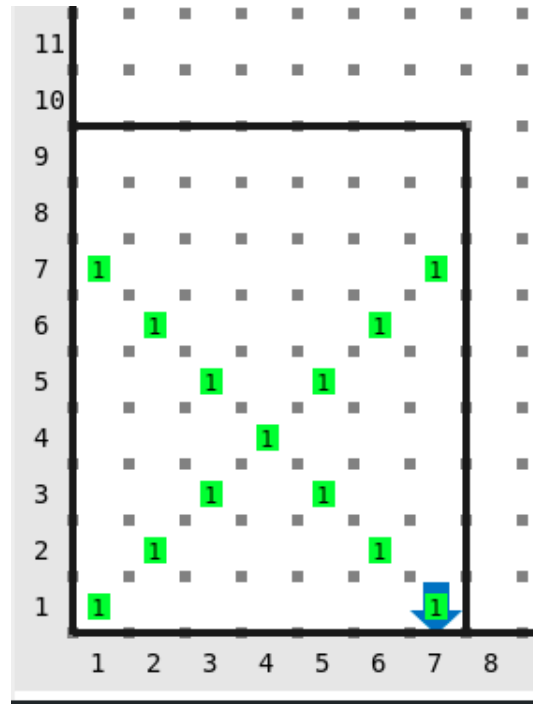


Figura 4: Diagonales correspondientes al mundo 7x9.

Puedes asumir lo siguiente:

- El número de zumbadores con los que cuenta Karel es infinito.
- Las calles son el eje y y Las avenidas el eje x. Los mundos posibles son aquellos  $(x, y)$  donde  $x \leq y$ .
- Un caso especial es cuando el mundo es un único cuadro  $(1, 1)$ , el programa debe responder correctamente a este mundo.
- Karel inicia en la esquina inferior izquierda, dando la cara hacia el este
- Al término de la ejecución Karel se encuentre en la esquina inferior derecha, dando la cara hacia el sur
- Si es que las diagonales comparten una esquina, cuida que en la intersección se tenga un sólo zumbador.

En tu solución se evaluará que tan bien descompusiste el problema original en subproblemas.

### 2.3. Problema 3: MidpointFindingKarel

Karel se encuentra dentro de un mundo cuadrado. La tarea es que Karel coloque un único zumbador al centro del mundo. Por ejemplo, si Karel comienza en un mundo como:

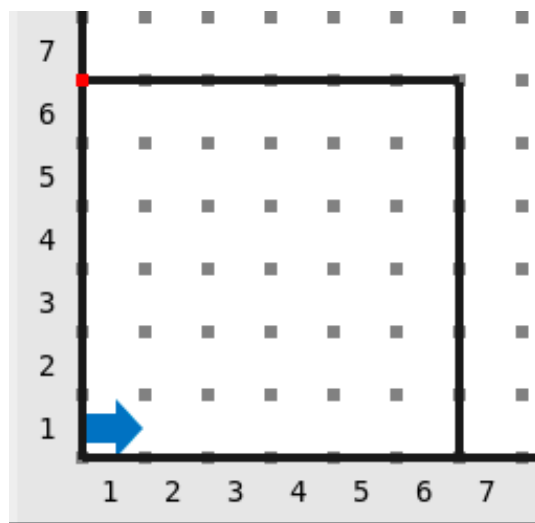


Figura 5: Mundo de 5x5 donde Karel inicia.

Al terminar, Karel debe estar en la misma esquina que el zumbador como se ve a continuación:

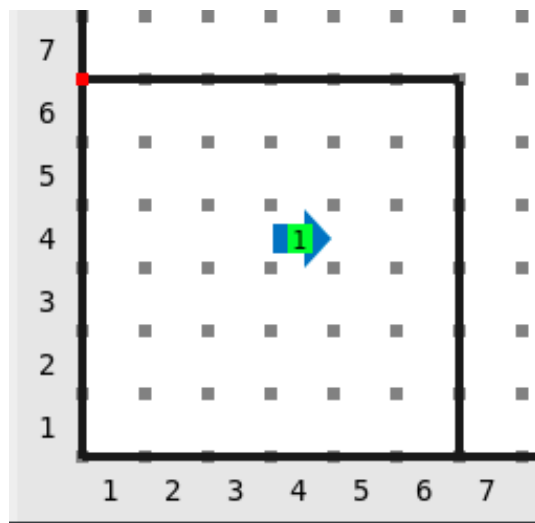


Figura 6: Karel coloca un zumbador en la posición 4,4 del mundo (6,6)

Observa que la configuración final debe contener un único zumbador en el centro del mundo. Antes de culminar, Karel tiene permitido colocar zumbadores adicionales, los cuales debe de recoger antes de ir a su posición final.

Para solucionar este problema, puedes tomar en cuenta los siguientes hechos acerca del mundo:

- Karel comienza en la esquina de calle 1 y avenida 1, dando la cara hacia el este, con un número infinito de zumbadores en su bolsa.
- El estado inicial del mundo no incluye paredes interiores o zumbadores.
- El mundo es cuadrado, es decir, el número de calles es el mismo que el número de avenidas.

El programa debe respetar lo siguiente:

- Si la longitud de los lados del mundo es impar, Karel debe poner el zumbador en la esquina que corresponde exactamente al centro el mundo.
- Si la longitud de los lados del mundo es par, Karel puede dejar el zumbador en cualquiera de las cuatro esquinas que corresponden al centro del mundo.

- Por ejemplo, para el mundo que se exhibe anteriormente, los centros del mundo son (3, 3), (3, 4), (4, 3) y (4, 4).
- No importa la dirección a la que Karel apunta al terminar la ejecución.
- Un caso particular es cuando tienen un cuadro 1x1. El programa debe de comportarse correctamente.

La recomendación para este problema es que utilices tu solución para el *Problema 2: DrawDiagonals*.

## 2.4. Problema 4: MultiplicationKarel

Supón que sobre las esquinas de la avenida 1 a se encuentran pilas de zumbadores, y sobre las esquinas de la segunda avenida se tienen zumbadores, a los más un zumbador por esquina. Karel inicia en la esquina de la calle 1 y avenida 1, dando la cara hacia el este como en el siguiente mundo:

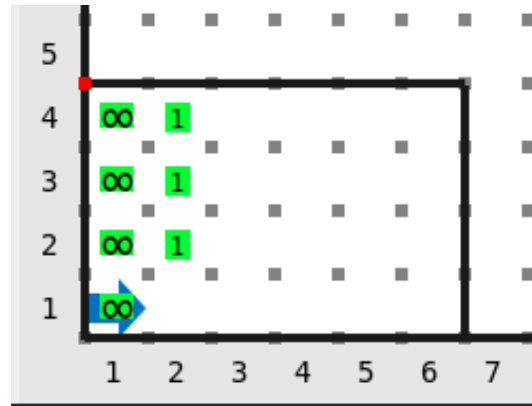


Figura 7: Mundo que pretende realizar la multiplicación 4x3

Diseña un programa que multiplique el número de pilas en la avenida 1 por el número de zumbadores en la avenida 2. Por ejemplo, para el mundo de la figura anterior se tiene que multiplicar 4 por 3.

Al término de la ejecución, el resultado se representará con el número de zumbadores en la esquina inferior derecha, como se puede apreciar en la siguiente figura:

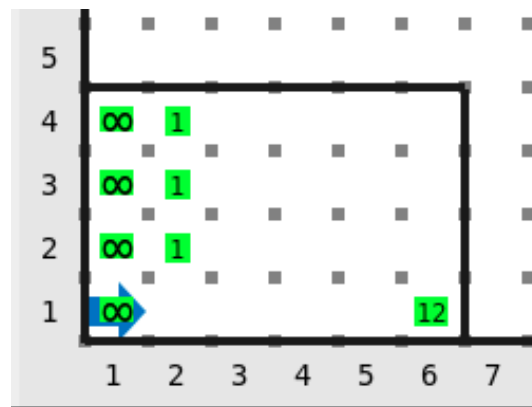


Figura 8: En el mundo se han puesto 12 zumbadores en la esquina de calle 1 y avenida 6.

Para tu programa puedes considerar:

- El número de zumbadores en las pilas ubicadas en la avenida 1 a es infinito.
- Karel inicia sin zumbador alguno en su bolsa.
- Una vez terminado tu programa la posición de los zumbadores y de Karel no importa.
- En la esquina inferior derecha se tenga el número de zumbadores equivalente al resultado de la multiplicación

---

Algunos casos especiales que el programa debe de considerar:

- Se inicia sin zumbadores en la avenida 1. Es decir, se intenta emular el producto  $0xn$
- Se inicia sin zumbadores en la avenida 2. Es decir, se intenta emular el producto  $nx0$

### 3. Consideraciones adicionales

Todos los problemas acerca de Karel que tu resuevas (a excepción de *CollectNewspaperKarel*) deben de funcionar satisfactoriamente en una variedad de mundos diferentes. Antes de enviar tus soluciones, asegúrate de probar tus programas en tantos mundos diferentes como te sea posible.

Cuando pruebes un programa, es importante que escojas casos prueba cuya finalidad es examinar aspectos diferentes del comportamiento de tu programa. Te recomendamos que realices casos prueba que examinen situaciones extremas, es decir, entradas o mundos iniciales que son legales pero que pueden resultar en un comportamiento especial.

Cuando escribas problemas en Karel, trata tanto como sea posible de usar la técnica de diseño top-bottom vista en clase. Descompón la tarea en pedazos pequeños hasta que cada subtarea sea algo que tu puedas llevar a cabo usando los comandos básicos y enunciados de control de Karel.

Un porcentaje de tu calificación se basará en que también presentas tu código. Principalmente, pon atención a los comentarios que hagas antes y dentro de los métodos que definas, de manera que sea claro el propósito y finalidad del código que estás escribiendo, asimismo concéntrate en escribir tu código con la sangría que corresponde,

### 4. Entrega

Dentro de la raíz de su repositorio deben crear un directorio por cada uno de los problemas llamada **Problema[de problema]**, por ejemplo, para el problema 1, se debe crear el directorio **Problema1**. Dentro de cada directorio deben de adjuntar el código final correspondiente al problema. Hay que asegurarse que se adjunta correctamente este código pues si tiene problemas de compilación, se tomará como inválida su propuesta. El código se puede descargar directamente de la página:

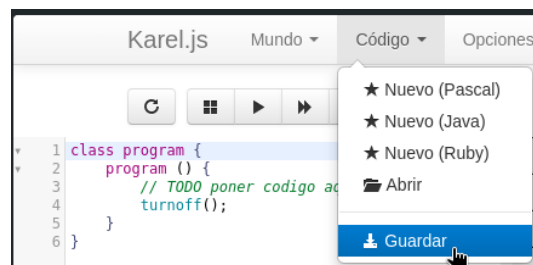


Figura 9: Botón para descargar el código de Karel

Además de incluir el código dentro de su directorio correspondiente, en el archivo **README.md** de su repositorio deben de incluir en un apartado cuál es la estrategia que siguieron para cada uno de los problemas. La claridad de su explicación es tan importante como el código que manden.