



PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE
ESCUELA DE INGENIERÍA
DEPARTAMENTO DE CIENCIA DE LA COMPUTACIÓN

IIC2233 - Programación Avanzada
1^{er} semestre 2017

Actividad 03

Jueves 16-Marzo-2017

Estructuras de Datos

Introducción

Chile está realizando un estudio para simular las cadenas de distribución, en situaciones de emergencia. El grupo de profesores de programación, con el afán de ayudar al país (y de pasada ganarse unos pesos) deciden intentar desarrollar una aplicación para dar solución al problema. En esta actividad, debes preparar todas las estructuras necesarias para poder llevar a cabo una simulación. **NO DEBEN HACER LA SIMULACIÓN.**

Debes implementar los siguientes tres elementos que existen en esta cadena de distribución, especificados como sigue:

Camiones: Son los encargados de llevar los productos a los distintos puntos afectados. Se te pide que cada camión tenga, como mínimo:

- `agregar_producto(producto)`: Método que recibe un producto, y lo almacena en el camión.
- `__str__`: Este método debe retornar como mínimo, una especificación con los tipos de productos que tiene en su interior, y la cantidad de cada uno de ellos.
- `capacidad_maxima`: Atributo que indica la cantidad máxima de kilogramos que puede llevar en su interior.
- `urgencia`: Atributo, es un número del 1 al 10 que indica el nivel de urgencia que tiene dicho camión por llegar a su destino (esto será relevante en el centro de distribución).

Centros de distribución: Se encargan de recibir los camiones y llenarlos lo máximo posible. Se pide que tenga, como mínimo:

- `fila`: Atributo que almacena los camiones a medida que van llegando. La forma de llegada y donde se ubica cada camión se explica más abajo.
- `bodega`: Atributo que almacena los distintos productos según su tipo.

- `recibir_camion(camion)`: Método que recibe un camión, y lo incluye en la fila, en la posición que corresponda según la urgencia.
- `rellenar_camion()`: Método que se encarga de llenar el primer camión de la fila. Al momento de ser llenado, se debe tener en cuenta que lo ideal es llevar la mayor cantidad de peso posible. Si el envío es exitoso, se debe mostrar en pantalla el camión que fue enviado, su nivel de urgencia, y cuantos productos de cada uno fueron enviados.
- `enviar_camion()`: Método que revisa el primer camión de la fila. Si ya fue llenado, lo saca, de no ser así no hace nada.
- `mostrar_productos_por(tipo)`: Recibe el tipo de producto que se quiere ver, y muestra todos los productos de ese tipo que hay en stock, y la cantidad de cada uno.
- `recibir_donacion(prod1, prod2, prod3, ...)`: Método que recibe una cantidad arbitraria de productos, y los va agregando a la bodega de manera ordenada. Se especifica más abajo como se debe organizar la bodega.

Producto: También se tienen los productos, cada uno de éstos debe tener, como mínimo:

- **nombre:** String que indica el nombre del producto
- **tipo:** String que indica de qué tipo es el producto (Alimento, Construcción, Hogar, etc...)
- **peso:** Número que indica cuanto pesa el producto

Funcionamiento fila Cuando un camión llega, este entra en la fila de los camiones, y se ubica en esta según el nivel de emergencia que tenga. A mayor nivel de urgencia, mayor prioridad (ej: si un camión tiene nivel de emergencia 8, debe avanzar en la fila hasta que se encuentre con un camión que tiene un nivel de emergencia de 8 o más).

Funcionamiento bodega En la bodega, los productos deben estar almacenados según su tipo, es decir, los artículos del tipo “Ropa” están separados de los del tipo “Alimento”. Una vez clasificados por tipo, dentro de estos se deben juntar todos los productos iguales (2 productos se dicen iguales, si tienen la misma categoría, y el mismo nombre; el peso no es relevante) Si en un momento dado, en una donación llega un producto de un tipo de producto que no existía anteriormente, se debe crear un nuevo estante con los productos de dicha categoría. Por ejemplo, si existen productos del tipo “Construcción” y “Alimento”, y en una donación llega un producto del tipo “Hogar”, entonces debe crear un nuevo espacio para los productos de ese tipo). **Usted no sabe a priori todos los tipos de productos que existen.**

Junto con el enunciado, se les entregarán 2 archivos `.txt`. En el archivo `camiones.txt`, estará la especificación de los camiones, en donde cada línea del archivo contiene las especificaciones de cada camión de la forma *peso máximo carga, nivel de urgencia*. Ej: Una línea del archivo de la forma 2500, 8 significa que hay un camión que puede llevar máximo 2500 kilogramos, y tiene un nivel de urgencia de 8.

El archivo `productos.txt`, corresponderá a los productos que se tienen disponibles en el centro de distribución, en donde cada línea del archivo representa un producto de la forma *tipo, nombre, peso*. Ej: Una línea del archivo de la forma “Construcción, Zinc, 30” significa que se tiene un zinc, que es del tipo construcción, que pesa 30 kilogramos.

Para mostrar que sus estructuras funcionan de manera correcta, su programa debe poblar el sistema con todos los datos del archivo (poner todos los productos en el stock y hacer que los camiones ingresen

en la fila), teniendo en cuenta que los camiones del archivo van llegando al centro de distribución según su aparición en el `.txt` (vale decir, el camión de la primera línea es el primero en llegar, el de la segunda línea es el segundo en llegar, etc...). Una vez poblado el sistema, deben rellenar y enviar 5 camiones.

Requerimientos

- Clase camión, con los atributos necesarios, y los métodos:
 - `__str__`: Muestra el listado de tipos de productos que lleva el camión, y cuántos lleva de cada uno.
 - `agregar_producto(producto)`: Recibe un producto, y lo agrega al camión de ser posible (no sobrepasar peso máximo).
- Clase CentroDistribucion, con los atributos necesarios y los métodos:
 - `recibir_donacion(producto1, producto2, ...)`: Recibe una cantidad no definida de productos, y los agrega al stock manteniendo el orden de los tipos de producto
 - `recibir_camion(camion)`: Recibe un camión, y lo añade a la fila en la posición que corresponda según el nivel de urgencia
 - `enviar_camion()`: Remueve de la fila el primer camión de la fila
 - `rellenar_camion()`: Rellena el primer camión de la fila de manera que lleve el máximo de peso posible.
 - `productos_por_tipo(tipo)`: Recibe un tipo de producto, y muestra todos los productos que son de ese tipo

Notas

- **OJO:** Una mala modelación de las estructuras de datos necesarias para almacenar los distintos datos será considerado incorrecto (a todos nos gustan las listas, pero no son siempre la mejor opción).
- Para rellenar los camiones, no es necesario usar backtracking, puedes implementar un algoritmo codicioso

To - DO

- Camión (1.50 pts)
 - `agregar_producto` (1.00 pto)
 - `__str__` (0.50 pts)
- Centro Distribución (4.00 pts)
 - `recibir_donacion` (1.00 pto)
 - `recibir_camion` (1.00 pto)
 - `enviar_camion` (0.50 pts)
 - `productos_por_tipo` (0.50 pts)
 - `rellenar_camion` (1.00 pto)
- Poblar el sistema (0.50 pts)

Entrega

- **Lugar:** GIT - Carpeta: Actividades/AC03
- **Hora:** 16:55