**Simulación de una empresa de Transporte de automóviles en C++**

Quitar métodos de pruebas como fin()



Álvaro Camacho Martín 48083824R

Óscar Montesano Martínez 53852484P

TAD´s implementados en el proyecto

* Pila:

Especificación PILA[VEHICULO]

Operaciones:

{Generadoras}

Pila():pila ; genera una pila vacía de longitud 0

Apilar(Vehiculo): ; Añade un vehículo a la cima de la pila

{Modificadoras}

Parcial Desapilar(): ; Saca el ultimo vehículo añadido a la pila

{Observadoras}

Parcial Cima(): Vehiculo; Muestra el último vehículo añadido a la pila

Es\_vacia(): Bool; Devuelve True si la pila está vacía

* Cola:

Especificación COLA[VEHICULO]

Operaciones:

{Generadoras}

Cola():cola ; genera una cola vacía (con ambos nodos a NULL y con longitud 0)

Encolar(Vehiculo): ; Añade un vehículo al final de la cola

{Modificadoras}

Parcial Desencolar(): Vehículo; Saca el ultimo vehículo de la cola (y lo devuelve)

{Observadoras}

Parcial Inicio(): Vehículo; Muestra el primer vehículo de la cola

Get\_longitud(): Entero; Devuelve la longitud de la cola, 0 si es vacía

Es\_vacia(): Bool; Devuelve True si la cola es vacía

Mostrar\_cola(): Vehiculos; Va mostrando el primer vehículo de la cola uno detrás de otro.

b.2) Para implementar ambos Tipos Abstractos de datos hemos usado memoria dinámica. Cada TAD consta de dos clases, la suya y el nodo. Un nodo es una celda que contiene un elemento de la estructura de datos además de un puntero que apunta al anterior nodo. En el caso de Colas, hay además un puntero que apunta al inicio, por donde salen los elementos y otro que apunta al final, por donde entran. La pila solo tiene uno apuntando al nodo que en ese momento sea la cima.

Las dificultades que nos hemos encontrado han principalmente sido trabajar con punteros, ya que nunca lo habíamos hecho, y aunque en java habíamos trabajado con clases en POO, al haber dos ficheros por clase y tantos métodos nos ha costado bastante más corregir los errores.

b.3) Como hemos mencionado anteriormente, contamos con una clase Nodo por cada TAD. Esta toma como ”friend class” a su TAD para que este pueda usarla sin problemas.

Para las colas, tomamos dos punteros a NodoCola, primero y ultimo. Si la cola es vacia apuntan a NULL. Cada vez que queremos insertar un elemento creamos un nuevo NodoCola; Si la cola es vacia tanto primero como ultimo apuntarán a el. Sino, apuntamos el siguiente de ultimo al nuevo nodo y este lo convertimos en ultimo. Finalmente incrementamos la longitud de la cola en uno.

Para las pilas,

b.3.1)

b.3.2) El programa comienza con un menú de 8 opciones. Para comenzar la simulación crearemos un numero de coches, ya sea aleatorio o determinado por el usuario, con ambas opciones brindadas por el programa. Una vez haya coches en las colas (Almacenes) podremos observarlos presionando 3. Si queremos llevar coches a alguna cola de zona podremos hacerlo, de igual manera, aleatoriamente o aportando el numero.

Para eliminar los coches de la fabrica basta con pulsar 4.

b.4) Bibliografía: Apuntes de la asignatura.