



Universidad Nacional Autónoma de México

Facultad de Estudios Superiores Aragón

Ingeniería en Computación

Asignatura: Estructura de datos

TAREA 8: Ejercicio con el ADT ColaConPrioridadAcotada(Banco)


Profesor: Jesús Hernández Cabrera

Alumno: Juan Diego Ortiz Cruz

Grupo: 1360

Fecha: 20/09/2024

Captura Clase ColaADT:



```
1  import { ListaLigada } from "../ListaLigada";
2
3  export class colaADT<E> {
4      private dato: ListaLigada<E>;
5
6      constructor() {
7          this.dato = new ListaLigada();
8      }
9
10     public estaVacia(): boolean {
11         return this.dato.getlongitud() === 0;
12     }
13
14     public longitud(): number {
15         return this.dato.getlongitud();
16     }
17
18     public frente(): E | null {
19         return this.dato.getPrimero();
20     }
21
22     public encolar(elemento: E): void {
23         this.dato.agregarAlFinal(elemento);
24     }
25
26     public desencolar(): E | null {
27         return this.dato.eliminarElPrimero();
28     }
29
```

Captura Clase Cola con prioridad acotada:

```
1  export class ColaConPrioridadAcotada<E> {
2      private colas: colaADT<E>[];
3      private maxPrioridad: number;
4
5      constructor(maxPrioridad: number) {
6          this.maxPrioridad = maxPrioridad;
7          this.colas = new Array(maxPrioridad + 1);
8          for (let i = 0; i <= this.maxPrioridad; i++) {
9              this.colas[i] = new colaADT<E>();
10         }
11     }
12
13     public longitud(): number {
14         let longitud = 0;
15         for (let i = 0; i <= this.maxPrioridad; i++) {
16             longitud += this.colas[i].longitud();
17         }
18         return longitud;
19     }
20
21     public estaVacia(): boolean {
22         return this.longitud() === 0;
23     }
24
25     public encolar(elemento: E, prioridad: number): void {
26         if (prioridad >= 1 && prioridad <= this.maxPrioridad) {
27             this.colas[prioridad].encolar(elemento);
28         }
29     }
30
31     public desencolar(): E | null {
32         for (let i = 1; i <= this.maxPrioridad; i++) {
33             if (!this.colas[i].estaVacia()) {
34                 return this.colas[i].desencolar();
35             }
36         }
37         return null;
38     }
39
40     public toString(): string {
41         let result = "ColaConPrioridadAcotada:\n";
42         for (let i = 1; i <= this.maxPrioridad; i++) {
43             result += `Prioridad ${i}: ${this.colas[i].toString()}\n`;
44         }
45         return result;
46     }
47 }
```

Capturas de la clase ClienteBanco:

```
1  export enum PerfilCliente {
2      NoCliente = 5,
3      ClienteNuevo = 4,
4      ClienteFrecuente = 3,
5      ClientePremium = 2,
6      Celebridad = 1
7  }
8
9  export class ClienteBanco {
10     private nombre: string;
11     private perfil: PerfilCliente;
12     private saldo: number;
13
14     constructor(nombre: string, perfil: PerfilCliente, saldo: number = 0) {
15         this.nombre = nombre;
16         this.perfil = perfil;
17         this.saldo = saldo;
18     }
19
20     public getNombre(): string {
21         return this.nombre;
22     }
23
24     public getPerfil(): PerfilCliente {
25         return this.perfil;
26     }
27
28     public getSaldo(): number {
29         return this.saldo;
30     }
31
32     public retirar(monto: number): boolean {
33         if (this.saldo >= monto) {
34             this.saldo -= monto;
35             return true;
36         }
37         return false;
38     }
39
40     public toString(): string {
41         return `ClienteBanco{nombre: "${this.nombre}",
42             perfil: ${PerfilCliente[this.perfil]}, saldo: ${this.saldo}}`;
43     }
44 }
```

Capturas de la clase Main:

```
1 class SistemaBanco {
2     private cola: ColaConPrioridadAcotada<ClienteBanco>;
3
4     constructor() {
5         this.cola = new ColaConPrioridadAcotada<ClienteBanco>(5);
6     }
7
8     public agregarCliente(cliente: ClienteBanco): void {
9         this.cola.encolar(cliente, cliente.getPerfil());
10        console.log(`${cliente.getNombre()} ha sido agregado a la cola.`);
11    }
12
13    public atenderCliente(): void {
14        const cliente = this.cola.desencolar();
15        if (cliente) {
16            console.log(`Atendiendo a ${cliente.getNombre()}`);
17            if (cliente.retirar(10000)) {
18                console.log(`${cliente.getNombre()} ha retirado $10,000. Nuevo saldo: ${cliente.getSaldo()}`);
19            } else {
20                console.log(`${cliente.getNombre()} no tiene suficiente saldo para retirar $10,000.`);
21            }
22        } else {
23            console.log("No hay más clientes en la cola.");
24        }
25    }
26
27    public mostrarEstadoCola(): void {
28        console.log("Estado actual de la cola:");
29        console.log(this.cola.toString());
30    }
31
32    public static main(): void {
33        console.log("Iniciando sistema de atencion a clientes(Banco)...");
34
35        const sistema = new SistemaBanco();
36
37        // 1. Lleguen 2 clientes nuevos
38        sistema.agregarCliente(new ClienteBanco("Juan", PerfilCliente.ClienteNuevo, 15000));
39        sistema.agregarCliente(new ClienteBanco("María", PerfilCliente.ClienteNuevo, 20000));
40
41        // 2. Lleguen 3 personas que no son clientes
42        sistema.agregarCliente(new ClienteBanco("Pedro", PerfilCliente.NoCliente));
43        sistema.agregarCliente(new ClienteBanco("Ana", PerfilCliente.NoCliente));
44        sistema.agregarCliente(new ClienteBanco("Luis", PerfilCliente.NoCliente));
45
46        // 3. Llego una celebridad
47        sistema.agregarCliente(new ClienteBanco("Famoso", PerfilCliente.Celebridad, 1000000));
48
49        // 4. Imprime el estado de la cola con prioridad acotada
50        sistema.mostrarEstadoCola();
51
52        // 5. Se atiende al siguiente cliente, en donde retire $10,000 de su cuenta
53        sistema.atenderCliente();
54
55        // 6. Llegan dos clientes más, uno frecuente y un premium
56        sistema.agregarCliente(new ClienteBanco("Carlos", PerfilCliente.ClienteFrecuente, 50000));
57        sistema.agregarCliente(new ClienteBanco("Elena", PerfilCliente.ClientePremium, 100000));
58
59        // 7. Atender al siguiente cliente
60        sistema.atenderCliente();
61
62        // 8. Imprime el estado de la cola con prioridad acotada
63        sistema.mostrarEstadoCola();
64
65        // 9. Atender todos los clientes restantes
66        while (!sistema.cola.estaVacia()) {
67            sistema.atenderCliente();
68        }
69
70        // 10. Imprime el estado de la cola con prioridad acotada
71        sistema.mostrarEstadoCola();
72    }
73 }
74
75 // Ejecutar el programa
76 SistemaBanco.main();
```

Capturas Ejecucion(Console):

```
PS C:\Users\LIMA\Desktop\Proyectos\Estructura-de-datos\tarea8> tsx Main
Iniciando sistema de atencion a clientes(Banco)...
Juan ha sido agregado a la cola.
María ha sido agregado a la cola.
Pedro ha sido agregado a la cola.
Ana ha sido agregado a la cola.
Luis ha sido agregado a la cola.
Famoso ha sido agregado a la cola.
Estado actual de la cola:
ColaConPrioridadAcotada:
Prioridad 1: ColaADT: [ ClienteBanco{nombre: "Famoso", perfil: Celebridad, saldo: $100000} (Frente) (Final) ]
Prioridad 2: ColaADT: [ ]
Prioridad 3: ColaADT: [ ]
Prioridad 4: ColaADT: [ ClienteBanco{nombre: "Juan", perfil: ClienteNuevo, saldo: $15000} (Frente) <- ClienteBanco{nombre: "María", perfil: ClienteNuevo, saldo: $20000} (Final) ]
Prioridad 5: ColaADT: [ ClienteBanco{nombre: "Pedro", perfil: NoCliente, saldo: $0} (Frente) <- ClienteBanco{nombre: "Ana", perfil: NoCliente, saldo: $0} <- ClienteBanco{nombre: "Luis", perfil: NoCliente, saldo: $0} (Final) ]

Atendiendo a Famoso
Famoso ha retirado $10,000. Nuevo saldo: $90000
Carlos ha sido agregado a la cola.
Elena ha sido agregado a la cola.
Atendiendo a Elena
Elena ha retirado $10,000. Nuevo saldo: $90000
Estado actual de la cola:
ColaConPrioridadAcotada:
Prioridad 1: ColaADT: [ ]
Prioridad 2: ColaADT: [ ]
Prioridad 3: ColaADT: [ ClienteBanco{nombre: "Carlos", perfil: ClienteFrecuente, saldo: $50000} (Frente) (Final) ]
Prioridad 4: ColaADT: [ ClienteBanco{nombre: "Juan", perfil: ClienteNuevo, saldo: $15000} (Frente) <- ClienteBanco{nombre: "María", perfil: ClienteNuevo, saldo: $20000} (Final) ]
Prioridad 5: ColaADT: [ ClienteBanco{nombre: "Pedro", perfil: NoCliente, saldo: $0} (Frente) <- ClienteBanco{nombre: "Ana", perfil: NoCliente, saldo: $0} <- ClienteBanco{nombre: "Luis", perfil: NoCliente, saldo: $0} (Final) ]

Atendiendo a Carlos
Carlos ha retirado $10,000. Nuevo saldo: $40000
Atendiendo a Juan
Juan ha retirado $10,000. Nuevo saldo: $5000
Atendiendo a María
María ha retirado $10,000. Nuevo saldo: $10000
Atendiendo a Pedro
Pedro no tiene suficiente saldo para retirar $10,000.
Atendiendo a Ana
Ana no tiene suficiente saldo para retirar $10,000.
Atendiendo a Luis
Luis no tiene suficiente saldo para retirar $10,000.
Estado actual de la cola:
ColaConPrioridadAcotada:
Prioridad 1: ColaADT: [ ]
Prioridad 2: ColaADT: [ ]
Prioridad 3: ColaADT: [ ]
Prioridad 4: ColaADT: [ ]
```