Penjelasan Kode Program: Double Circular Linked List di PHP

Berikut ini adalah penjelasan baris demi baris dari kode program implementasi Double Circular Linked List menggunakan PHP.

class Node {  
Penjelasan: Mendefinisikan kelas Node yang menjadi elemen dasar dalam linked list.

public $data;  
Penjelasan: Menyimpan nilai data dari node.

public $next;  
Penjelasan: Pointer ke node selanjutnya.

public $prev;  
Penjelasan: Pointer ke node sebelumnya.

function \_\_construct($data) {  
Penjelasan: Konstruktor untuk menginisialisasi data node saat dibuat.

$this->data = $data;  
Penjelasan: Menyimpan nilai data ke dalam node.

$this->next = null;  
Penjelasan: Inisialisasi pointer next ke null.

$this->prev = null;  
Penjelasan: Inisialisasi pointer prev ke null.

}  
Penjelasan: Tutup fungsi konstruktor.

}  
Penjelasan: Tutup kelas Node.

class DoubleCircularLinkedList {  
Penjelasan: Mendefinisikan kelas untuk linked list circular dua arah.

public $head = null;  
Penjelasan: Pointer ke node pertama.

public $tail = null;  
Penjelasan: Pointer ke node terakhir.

function insertFront($data) {  
Penjelasan: Fungsi untuk menambahkan node di depan list.

$newNode = new Node($data);  
Penjelasan: Membuat node baru dengan data yang diberikan.

if ($this->head == null) {  
Penjelasan: Jika list kosong.

$this->head = $this->tail = $newNode;  
Penjelasan: Head dan tail menunjuk ke node yang sama.

$newNode->next = $newNode->prev = $newNode;  
Penjelasan: Node menunjuk ke dirinya sendiri (circular).

} else {  
Penjelasan: Jika list tidak kosong.

$newNode->next = $this->head;  
Penjelasan: Node baru menunjuk ke head sebelumnya.

$newNode->prev = $this->tail;  
Penjelasan: Node baru menunjuk ke tail.

$this->head->prev = $newNode;  
Penjelasan: Head sebelumnya menunjuk balik ke node baru.

$this->tail->next = $newNode;  
Penjelasan: Tail menunjuk ke node baru.

$this->head = $newNode;  
Penjelasan: Update head ke node baru.

}  
Penjelasan: Tutup kondisi else.

}  
Penjelasan: Tutup fungsi insertFront.

function insertBack($data) {  
Penjelasan: Fungsi untuk menambahkan node di akhir list.

$newNode = new Node($data);  
Penjelasan: Membuat node baru dengan data yang diberikan.

if ($this->tail == null) {  
Penjelasan: Jika list kosong.

$this->head = $this->tail = $newNode;  
Penjelasan: Inisialisasi head dan tail.

$newNode->next = $newNode->prev = $newNode;  
Penjelasan: Node menunjuk ke dirinya sendiri.

} else {  
Penjelasan: Jika list tidak kosong.

$newNode->prev = $this->tail;  
Penjelasan: Node baru menunjuk ke tail lama.

$newNode->next = $this->head;  
Penjelasan: Node baru menunjuk ke head.

$this->tail->next = $newNode;  
Penjelasan: Tail lama menunjuk ke node baru.

$this->head->prev = $newNode;  
Penjelasan: Head menunjuk balik ke node baru.

$this->tail = $newNode;  
Penjelasan: Update tail ke node baru.

}  
Penjelasan: Tutup kondisi else.

}  
Penjelasan: Tutup fungsi insertBack.

function deleteFront() {  
Penjelasan: Fungsi untuk menghapus node dari depan.

if ($this->head == null) return;  
Penjelasan: Jika list kosong, keluar fungsi.

if ($this->head == $this->tail) {  
Penjelasan: Jika hanya satu node dalam list.

$this->head = $this->tail = null;  
Penjelasan: Kosongkan list.

} else {  
Penjelasan: Jika lebih dari satu node.

$this->head = $this->head->next;  
Penjelasan: Pindah head ke node berikutnya.

$this->head->prev = $this->tail;  
Penjelasan: Update pointer prev pada head baru.

$this->tail->next = $this->head;  
Penjelasan: Update pointer next pada tail.

}  
Penjelasan: Tutup kondisi else.

}  
Penjelasan: Tutup fungsi deleteFront.

function deleteBack() {  
Penjelasan: Fungsi untuk menghapus node dari belakang.

if ($this->tail == null) return;  
Penjelasan: Jika list kosong, keluar fungsi.

if ($this->head == $this->tail) {  
Penjelasan: Jika hanya satu node.

$this->head = $this->tail = null;  
Penjelasan: Kosongkan list.

} else {  
Penjelasan: Jika lebih dari satu node.

$this->tail = $this->tail->prev;  
Penjelasan: Pindah tail ke node sebelumnya.

$this->tail->next = $this->head;  
Penjelasan: Update pointer next pada tail baru.

$this->head->prev = $this->tail;  
Penjelasan: Update pointer prev pada head.

}  
Penjelasan: Tutup kondisi else.

}  
Penjelasan: Tutup fungsi deleteBack.

function display() {  
Penjelasan: Menampilkan semua node dalam list.

if ($this->head == null) {  
Penjelasan: Jika list kosong.

echo "List kosong\n";  
Penjelasan: Tampilkan pesan kosong.

return;  
Penjelasan: Keluar dari fungsi.

}  
Penjelasan: Tutup kondisi if.

$current = $this->head;  
Penjelasan: Mulai dari head.

do {  
Penjelasan: Gunakan do-while untuk menelusuri circular list.

echo $current->data . " ";  
Penjelasan: Cetak data dari node saat ini.

$current = $current->next;  
Penjelasan: Pindah ke node berikutnya.

} while ($current != $this->head);  
Penjelasan: Ulangi hingga kembali ke head.

echo "\n";  
Penjelasan: Ganti baris setelah selesai.

}  
Penjelasan: Tutup fungsi display.

function clear() {  
Penjelasan: Fungsi untuk menghapus seluruh list.

$this->head = $this->tail = null;  
Penjelasan: Set head dan tail menjadi null.

}  
Penjelasan: Tutup fungsi clear.

}  
Penjelasan: Tutup kelas DoubleCircularLinkedList.