자료구조 과제

2020011919 곽윤정

|  |
| --- |
| 연습문제 01. |
| 더미 헤드를 사용하는 장점  효율성이 좋음.  포인터 변수 tail이 불필요하다  더미헤드 사용시 단점  저장된 순서를 유지하지 않는다. |
| 연습문제 02. |
| **public** **boolean** contains(E x) {  **return** indexOf(x) >= 0;  } |
| 연습문제 03. |
| 1. get, set, indexOf메소드  2. **public** Integer get(**int** index) {  **if**(index >= 0 && index <= numItems -1)  **return** item[index+1];  **else**  **return** **null**;  }    **public** **void** set(**int** index, Integer x) {  **if**(index >= 0 && index <= numItems -1)  item[index + 1 ] = x;  **else**  System.out.println("인덱스 범위가 잘못되었다.");  }  **public** **int** indexOf(Integer x) {  **int** i = 0;  **for** (i = 0; i < numItems; i++) {  **if** (((Comparable)item[i]).compareTo(x) == 0)  **return** i;  }  **return** -1;  } |
| 연습문제 04. |
| **public** **void** printInterval(**int** i, **int** j) {  **for(;i<=j; i ++) {**  **System.*out*.println(item[i]);**  **}**  } |
| 연습문제 05. |
| **public** **void** printInterval(**int** i, **int** j) {  **Node<E> tNode = getNode(i);**  **for(;i<=j; i ++) {**  **System.*out*.println(tNode.item);**  **tNode = tNode.next;**  **}**  } |
| 연습문제 06. |
| 1.  2. |
| 연습문제 07. |
| **public** **void** remove(**int** index, **int** k) {  **if**(index>=0 && index< numItems) {  **Node prevNode = getNode(index -1);**  **for(int i = 0; i<k; i++) {**  **if(prevNode.next != null) {**  **prevNode.next = prevNode.next.next;**  **numItems--;**  }  **else**  **break**;    }  }  } |
| 연습문제 09. |
| **public** **void** concat(LinkedList l1, LinkedList l2) {  Node<E> t = l1.head;  **while**(t.next!= **null**) {  t=t.next;  }  t.next=l2.head.next;  l1.numItems+=l2.numItems;  } |
| 연습문제 10. |
| **public** **boolean** Linkednode() {  **ListInterface<E> list = null;**  **E node1 = null;**  **E node2 = null;**  **if (list.indexOf(node1) >= 0 && list.indexOf(node2) >= 0)**  **return true;**  **else**  **return false;**  } |
| 연습문제 11. |
| 문자 처음부터 마지막까지 하나씩 돌린다음 가장 마지막 x의 값을 반환하면된다! |
| 연습문제 12. |
| public int lastIndexOf(E x) {  Node<E> q = head.next;  int a = 0;  for(int i = 0; q != null; q = q.next, i++) {  if(q.item == x) {  a = i;  }  }  return a;  } |
| 합 |
| **public** **int** hap() {  Node<E> t;  **int** sum = 0;  **for** (t = head.next; t != **null**; t = t.next) {  sum += (**int**) t.item;  }  System.***out***.print("Hap (#items=" + numItems + ") " + sum);  System.***out***.println();  **return** sum;  } |
| 최대값 |
| **public** **int** maxList() {  Node<E> t;  **int** sum = 0;  **for** (t = head.next; t != **null**; t = t.next) {  **if** (sum <= (**int**) t.item)  sum = (**int**) t.item;  }  System.***out***.print("Maxlist (#items=" + numItems + ") " + sum);  System.***out***.println();  **return** sum;  } |
| 최소값 |
| **public** **int** minList() {  Node<E> t;  **int** sum = 999999;  **for** (t = head.next; t != **null**; t = t.next) {  **if** (sum >= (**int**) t.item)  sum = (**int**) t.item;  }  System.***out***.print("Minlist (#items=" + numItems + ") "+ sum);  System.***out***.println();  **return** sum;  } |
| 결과값 |
|  |