/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Online Java Compiler.

Code, Compile, Run and Debug java program online.

Write your code in this editor and press "Run" button to execute it.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

import java.util.\*;

class using\_queue

{

public static Queue create\_queue()

{

Scanner s=new Scanner(System.in);

Queue qu = new LinkedList();

System.out.println("Enter Queue Elements (integer only):");

int a=0,i=1;

char ch='y';

System.out.println("Press y/Y to add another element.");

System.out.println("Press anything else to stop.");

System.out.println("Input Choice:");

ch=s.next().charAt(0);

while(ch=='y' || ch=='Y')

{

System.out.println("Enter Element "+i+":");

a=s.nextInt();

qu.add(a);

i++;

System.out.println("Wish to add another:");

ch=s.next().charAt(0);

}

return qu;

}

static void queue\_entire\_print(Queue qu){

Iterator itr=qu.iterator();

int i=1;

while(itr.hasNext())

{

System.out.println("Element "+i+" is:" +itr.next());

i++;

}

}

public using\_queue()

{

Scanner s=new Scanner(System.in);

System.out.println("Creating queue........");

Queue qu=new LinkedList();

qu=create\_queue();

System.out.println("Queue Operations............");

System.out.println("Press 1. to check if queue is empty or not.");

System.out.println("Press 2. to check the top element.");

System.out.println("Press 3. to remove an element.");

System.out.println("Press 4. to add a new element.");

System.out.println("Press 5. to search for an element in queue.");

System.out.println("Press 6. to display entire queue.");

System.out.println("Press 7. to display size of queue.");

System.out.println("Input Choice:");

int choice=s.nextInt();

switch(choice)

{

case 1:

boolean a=qu.isEmpty();

System.out.println("Output whether queue is empty or not:"+a);

break;

case 2:

System.out.println("Current queue is:"+qu.peek());

break;

case 3:

qu.remove();

break;

case 4:

System.out.println("Enter an element (for pushing into queue):");

int d=s.nextInt();

qu.add(d);

break;

case 5:

System.out.println("Enter an element (to search within queue):");

int e=s.nextInt();

boolean f=qu.contains(e);

System.out.println("Output for the element entered for searching is:"+f);

break;

case 6:

queue\_entire\_print(qu);

break;

case 7:

System.out.println("Size of the queue is:"+qu.size());

break;

default:System.out.println("Wrong Choice entered................");

System.exit(0);

}

System.out.println("Final queue is:");

queue\_entire\_print(qu);

}

}

class using\_stack

{

public static Stack create\_stack()

{

Scanner s=new Scanner(System.in);

Stack st = new Stack();

System.out.println("Enter Stack Elements (integer only):");

int a=0,i=1;

char ch='y';

System.out.println("Press y/Y to add another element.");

System.out.println("Press anything else to stop.");

System.out.println("Input Choice:");

ch=s.next().charAt(0);

while(ch=='y' || ch=='Y')

{

System.out.println("Enter Element "+i+":");

a=s.nextInt();

st.push(a);

i++;

System.out.println("Wish to add another:");

ch=s.next().charAt(0);

}

return st;

}

static void stack\_entire\_print(Stack st){

ListIterator itr=st.listIterator();

int i=1;

while(itr.hasNext())

{

System.out.println("Element "+i+" is:" +itr.next());

i++;

}

}

public using\_stack()

{

Scanner s=new Scanner(System.in);

System.out.println("Creating Stack........");

Stack st=new Stack();

st=create\_stack();

System.out.println("Stack Operations............");

System.out.println("Press 1. to check if stack is empty or not.");

System.out.println("Press 2. to check the top element.");

System.out.println("Press 3. to pop an element.");

System.out.println("Press 4. to push a new element.");

System.out.println("Press 5. to search for an element in stack.");

System.out.println("Press 6. to display entire stack.");

System.out.println("Press 7. to display size of stack.");

System.out.println("Press 8. to insert element at a specific position.");

System.out.println("Press 9. to remove element from a specific position.");

System.out.println("Input Choice:");

int choice=s.nextInt();

switch(choice)

{

case 1:

boolean a=st.empty();

System.out.println("Output whether stack is empty or not:"+a);

break;

case 2:

System.out.println("Top of the element of the stack is:"+st.peek());

break;

case 3:

st.pop();

break;

case 4:

System.out.println("Enter an element (for pushing into stack):");

int d=s.nextInt();

st.push(d);

break;

case 5:

System.out.println("Enter an element (to search within stack):");

int e=s.nextInt();

int f=st.search(e);

System.out.println("Index of element entered for searching is:"+f);

break;

case 6:

stack\_entire\_print(st);

break;

case 7:

System.out.println("Size of the stack is:"+st.size());

break;

case 8:

System.out.println("Enter element value(for inserting into stack):");

int b=s.nextInt();

System.out.println("Enter index/position for inserting:");

int c=s.nextInt();

st.insertElementAt(b,c);

break;

case 9:

System.out.println("Enter index/position for removing:");

int k=s.nextInt();

st.removeElementAt(k);

break;

default:System.out.println("Wrong Choice entered................");

System.exit(0);

}

System.out.println("Final Stack is:");

stack\_entire\_print(st);

}

}

public class Main

{

public static void main(String[] args) {

System.out.println("Implementing Stack and Queue (Collections)...............");

System.out.println("Press 1. to implement Stack.");

System.out.println("Press 2. to implement Queue.");

System.out.println("Press anything else to exit.");

System.out.println("Enter choice:");

Scanner s=new Scanner(System.in);

int ch=s.nextInt();

switch(ch)

{

case 1: using\_stack obj\_st=new using\_stack();

break;

case 2: using\_queue obj\_qu=new using\_queue();

break;

default: System.out.println("Exitting.................");

System.exit(0);

}

System.out.println("Implemented Stacks and Queues ................");

}

}