Darshana pubudu keerthirathna

ICM 106 OR23106564

Object Oriented Programming WEEK – 02 ASSIGNMENT

**Question 01**

class Stack{

private int nextIndex;

private int[] dataArray;

private int loadFactor;

Stack(int size){

dataArray=new int[size];

nextIndex=0;

loadFactor=2;

}

private boolean isEmpty(){

return nextIndex<=0;

}

private boolean isFull(){

return nextIndex>=capacity();

}

private void extendArray(){

int[] tempArr = new int[capacity()\*loadFactor];

for (int i = 0; i < dataArray.length; i++){

tempArr[i]= dataArray[i];

}

dataArray=tempArr;

}

public void push(int data){

if (isFull()){

extendArray();

}

dataArray[nextIndex++]=data;

}

public void printStack(){

System.out.print("[");

for(int i=nextIndex-1;i>=0;i--){

System.out.print(dataArray[i]+", ");

}

System.out.println(isEmpty() ? "empty]": "\b\b]");

}

public void pop(){

if(isEmpty()){

System.out.println("Stack is empty...");

}else{

nextIndex--;

}

}

public int size(){

return nextIndex;

}

public void clear(){

nextIndex=0;

}

public int capacity(){

return dataArray.length;

}

public int[] toArray(){

int[] tempArr = new int[nextIndex];

for (int i = 0; i<nextIndex; i++){

tempArr[i] = dataArray[(nextIndex-1)-i];

}

return tempArr;

}

}

class Demo{

public static void main(String args[]){

Stack s1=new Stack(10); //Initial capacity of the stack is 10

s1.printStack(); //[empty]

System.out.println("Size of the stack is : "+s1.size()); //0

System.out.println("Capacity of the stack is : "+s1.capacity()); //10

s1.push(10);

s1.push(20);

s1.push(30);

s1.push(40);

s1.push(50);

s1.printStack(); //[50, 40, 30, 20, 10]

System.out.println("Size of the stack is : "+s1.size()); //5

System.out.println("Capacity of the stack is : "+s1.capacity()); //10

s1.push(60);

s1.push(70);

s1.push(80);

s1.push(90);

s1.push(100);

s1.printStack(); //[100,90,80,70,60,50,40, 30, 20, 10]

System.out.println("Size of the stack is : "+s1.size()); //10

System.out.println("Capacity of the stack is : "+s1.capacity()); //10

s1.push(111);

s1.printStack(); //[111,100,90,80,70,60,50,40, 30, 20, 10]

System.out.println("Size of the stack is : "+s1.size()); //11

System.out.println("Capacity of the stack is : "+s1.capacity()); //20

s1.push(222);

s1.push(333);

s1.push(444);

s1.printStack(); //[444,333,222,111,100,90,80,70,60,50,40, 30,20,10]

System.out.println("Size of the stack is : "+s1.size()); //14

System.out.println("Capacity of the stack is : "+s1.capacity()); //20

int[] ar=s1.toArray();

for(int a : ar){

System.out.print(a+" "); //444 333 222 111 100 90 80 70 60 50 40 30 20 10

}

}

}

**Question 02**

class PriorityQueue{

private int nextIndex;

private int[] dataArray;

PriorityQueue(int size){

nextIndex = 0;

dataArray = new int[size];

}

private int findMaxIndex(){

int max=dataArray[0];

int index = 0;

for (int i = 0; i < nextIndex; i++){

if(dataArray[i]>max){

max=dataArray[i];

index = i;

}

}

return index;

}

private void swapNum(int a, int b){

int temp = dataArray[a];

dataArray[a]=dataArray[b];

dataArray[b]=temp;

}

public void enQueue(int num){

dataArray[nextIndex++]=num;

int maxIndex=findMaxIndex();

swapNum(0,maxIndex);

}

public void deQueue(){

for (int i = 0; i < nextIndex; i++){

dataArray[i]=dataArray[i+1];

}

int maxIndex=findMaxIndex();

swapNum(0,maxIndex);

nextIndex--;

}

public void printQueue(){

System.out.print("[");

for (int i = 0; i < nextIndex; i++){

System.out.print(dataArray[i]+", ");

}

System.out.println(nextIndex==0?"empty]":"\b\b]");

}

}

class Demo{

public static void main(String args[]){

PriorityQueue pq=new PriorityQueue(10); //PriorityQueue(int initialSize)

pq.enQueue(12);

pq.enQueue(90);

pq.enQueue(16);

pq.enQueue(45);

pq.enQueue(96);

pq.enQueue(23);

pq.printQueue(); //[96, 16, 12, 90, 45, 23]

pq.deQueue();

pq.printQueue(); //[90, 16, 23, 45, 12]

pq.deQueue();

pq.printQueue(); //[45, 16, 23, 12]

}

}

**Question 03**

class PatientQueue{

private int nextIndex;

private Patient[] objectArray;

PatientQueue(){

nextIndex=0;

objectArray = new Patient[0];

}

private boolean isFull(){

return nextIndex>=objectArray.length;

}

private boolean isEmpty(){

return nextIndex<=0;

}

private void extendArray(){

Patient[] temp = new Patient[objectArray.length+1];

for (int i = 0; i < nextIndex; i++){

temp[i]=objectArray[i];

}

objectArray=temp;

}

public void enQueue(Patient obj){

if (isFull()){

extendArray();

}

objectArray[nextIndex++] = obj;

}

public Patient deQueue(){

Patient firsePatient = objectArray[0];

if (!isEmpty()){

for (int i = 0; i < nextIndex-1; i++){

objectArray[i]=objectArray[i+1];

}

nextIndex--;

}

return firsePatient;

}

public void printQueue(){

System.out.print("{");

for (int i = 0; i < nextIndex; i++){

System.out.print("["+objectArray[i].getNo()+"-"+objectArray[i].getName()+"], ");

}

System.out.println(nextIndex==0?"empty}":"\b\b}");

}

public int size(){

return nextIndex;

}

public void clear(){

nextIndex=0;

}

}

class Patient{

private int no;

private String name;

Patient(int no, String name){

this.no = no;

this.name=name;

}

public String getName(){

return name;

}

public int getNo(){

return no;

}

public String getPatientDetail(){

String strNo = String.valueOf(no);

return "["+strNo+"-"+name+"]";

}

}

class Demo{

public static void main(String args[]){

PatientQueue queue=new PatientQueue();

queue.enQueue(new Patient(101,"Amal"));

queue.enQueue(new Patient(102,"Nimal"));

queue.enQueue(new Patient(103,"Ramal"));

queue.enQueue(new Patient(104,"Bimal"));

queue.printQueue(); //{[101-Amal], [102-Niaml], [103-Ramal], [104-Bimal]}

Patient firstPatient= queue.deQueue();

System.out.println(firstPatient.getPatientDetail()); //[1001-Amal]

queue.printQueue(); //{[102-Niaml], [103-Ramal], [104-Bimal]}

System.out.println("No of patient of the queue : "+queue.size()); //3

queue.clear();

queue.printQueue(); //{Empty}

System.out.println("No of patient of the queue : "+queue.size()); //0

}

}

**Question 04**

class Registry{

private int nextIndex;

private int[] dataArray;

Registry(int size){

nextIndex=0;

dataArray = new int[size];

}

public void add(int num){

dataArray[nextIndex++]=num;

}

public void add(int index, int num){

for (int i = nextIndex-1; i >= index; i--){

dataArray[i+1]=dataArray[i];

}

dataArray[index]=num;

nextIndex++;

}

public void add(int[] arr){

for (int i = 0; i < arr.length; i++){

add(arr[i]);

}

}

public void add(int index, int[] arr){

for (int i = arr.length-1; i >=0; i--){

add(index, arr[i]);

}

}

public void remove(){

for (int i = 0; i<nextIndex; i++){

dataArray[i]=dataArray[i+1];

}

nextIndex--;

}

public void remove(int index){

for (int i = index; i<nextIndex; i++){

dataArray[i]=dataArray[i+1];

}

nextIndex--;

}

public void remove(int startIndex, int endIndex){

for (int i = endIndex-1; i >= startIndex; i--){

remove(i);

}

}

public void printRegistry(){

System.out.print("[");

for (int i = 0; i < nextIndex; i++){

System.out.print(dataArray[i]+", ");

}

System.out.println(nextIndex<=0?"empty]":"\b\b]");

}

}

class Demo{

public static void main(String args[]){

Registry reg=new Registry(100); //

reg.add(10);

reg.add(20);

reg.add(30);

reg.add(40);

reg.printRegistry(); //[10,20,30,40]

reg.remove(); //remove the first element

reg.printRegistry(); //[20,30,40]

reg.add(1,25);//add(int index, int data)

reg.printRegistry(); //[20,25,30,40]

reg.add(new int[]{100,200,300,400}); //add(int[] data)

reg.printRegistry(); //[20,25,30,40,100,200,300,400]

reg.remove(1); //remove(int index)

reg.printRegistry(); //[20,30,40,100,200,300,400]

reg.add(3,new int[]{1,2,3}); //add(int index, int[] data)

reg.printRegistry(); //[20,30,40,1,2,3,100,200,300,400]

reg.remove(3,6); //remove(int startIndex, int endIndex-1)

reg.printRegistry(); //[20,30,40,100,200,300,400]

}

}