1 **import** java.util.ArrayList;

2

3 **public class** GenericQueue <T>

4 {

5 **private int** size;

6 **private int** numOfNodes = 0;

7 **private int** front = 0;

8 **private int** rear = 0;

9 **private** ArrayList <T> data;

10

11 **public** GenericQueue(**int** n)

12 {

13 size = n;

14 data = new ArrayList <T> (size);

15 }

16

17 **public boolean** enQueue(T newItem)

18 {

19 **if**(numOfNodes == size) **//The queue is full**

20 {

21 **return false**;

22 }

23 **else** **// add the object to the structure**

24 {

25 numOfNodes = numOfNodes + 1;

26 data.add(rear, newItem); **//data[rear] = (T) item.clone();**

27 rear = (rear + 1) % size;

28 return true;

29 }

30 }

31

32 **public** T deQueue( ) **//fetch and delete an object**

33 {

34 **int** frontLocation;

35 **if**(numOfNodes == 0) **//The queue is empty**

36 {

37 **return null**;

38 }

39 **else**

40 {

41 frontLocation = front;

42 front = (front + 1) % size;

43 numOfNodes = numOfNodes - 1;

44 **return** data.get(frontLocation);

45 }

46 }

47 }

**Figure 13.23 The class GenericQueue.**