**Java program to implement Queue Using Array And Class.**

**package** pblms;

**import** java.util.\*;

**public** **class** QueueImplement

{

**public** **static** **void** main(String[] args)

{

Scanner scan = **new** Scanner(System.***in***);

System.***out***.println("Array Queue Test\n");

System.***out***.println("Enter Size of Integer Queue ");

**int** n = scan.nextInt();

arrayQueue q = **new** arrayQueue(n);

**char** ch;

**do**{

System.***out***.println("\nQueue Operations");

System.***out***.println("1. insert");

System.***out***.println("2. remove");

System.***out***.println("3. peek");

System.***out***.println("4. check empty");

System.***out***.println("5. check full");

System.***out***.println("6. size");

**int** choice = scan.nextInt();

**switch** (choice)

{

**case** 1 :

System.***out***.println("Enter integer element to insert");

**try**

{

q.insert( scan.nextInt() );

}

**catch**(Exception e)

{

System.***out***.println("Error : " +e.getMessage());

}

**break**;

**case** 2 :

**try**

{

System.***out***.println("Removed Element = "+q.remove());

}

**catch**(Exception e)

{

System.***out***.println("Error : " +e.getMessage());

}

**break**;

**case** 3 :

**try**

{

System.***out***.println("Peek Element = "+q.peek());

}

**catch**(Exception e)

{

System.***out***.println("Error : "+e.getMessage());

}

**break**;

**case** 4 :

System.***out***.println("Empty status = "+q.isEmpty());

**break**;

**case** 5 :

System.***out***.println("Full status = "+q.isFull());

**break**;

**case** 6 :

System.***out***.println("Size = "+ q.getSize());

**break**;

**default** : System.***out***.println("Wrong Entry \n ");

**break**;

}

q.display();

System.***out***.println("\nDo you want to continue (Type y or n) \n");

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

} **while** (ch == 'Y'|| ch == 'y');

}

}

**package** pblms;

**import** java.util.\*;

**class** arrayQueue

{

**protected** **int** Queue[] ;

**protected** **int** front, rear, size, len;

**public** arrayQueue(**int** n)

{

size = n;

len = 0;

Queue = **new** **int**[size];

front = -1;

rear = -1;

}

**public** **boolean** isEmpty()

{

**return** front == -1;

}

**public** **boolean** isFull()

{

**return** front==0 && rear == size -1 ;

}

**public** **int** getSize()

{

**return** len ;

}

**public** **int** peek()

{

**if** (isEmpty())

**throw** **new** NoSuchElementException("Underflow Exception");

**return** Queue[front];

}

**public** **void** insert(**int** i)

{

**if** (rear == -1)

{

front = 0;

rear = 0;

Queue[rear] = i;

}

**else** **if** (rear + 1 >= size)

**throw** **new** IndexOutOfBoundsException("Overflow Exception");

**else** **if** ( rear + 1 < size)

Queue[++rear] = i;

len++ ;

}

**public** **int** remove()

{

**if** (isEmpty())

**throw** **new** NoSuchElementException("Underflow Exception");

**else**

{

len-- ;

**int** ele = Queue[front];

**if** ( front == rear)

{

front = -1;

rear = -1;

}

**else**

front++;

**return** ele;

}

}

**public** **void** display()

{

System.***out***.print("\nQueue = ");

**if** (len == 0)

{

System.***out***.print("Empty\n");

**return** ;

}

**for** (**int** i = front; i <= rear; i++)

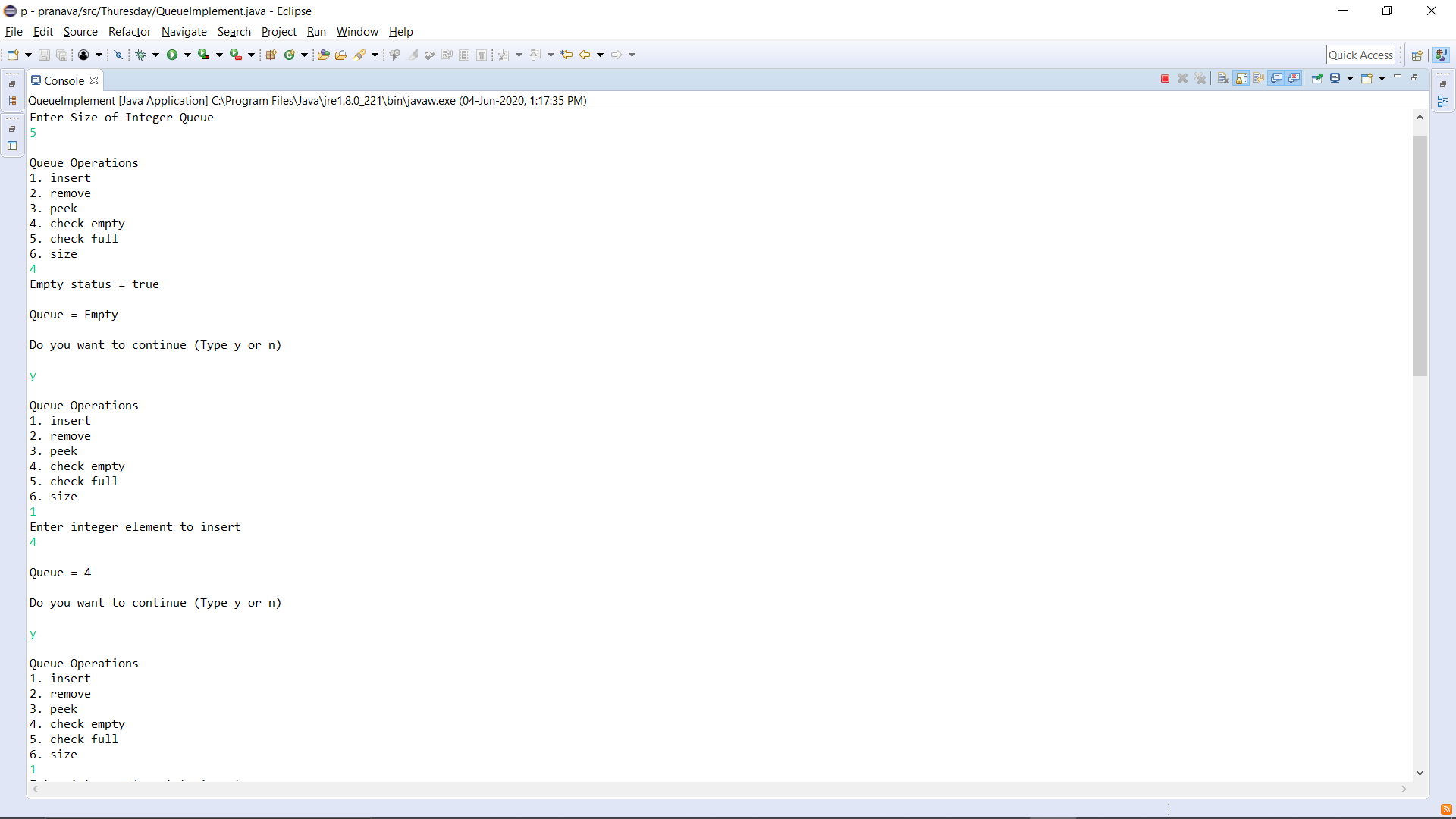
System.***out***.print(Queue[i]+" ");

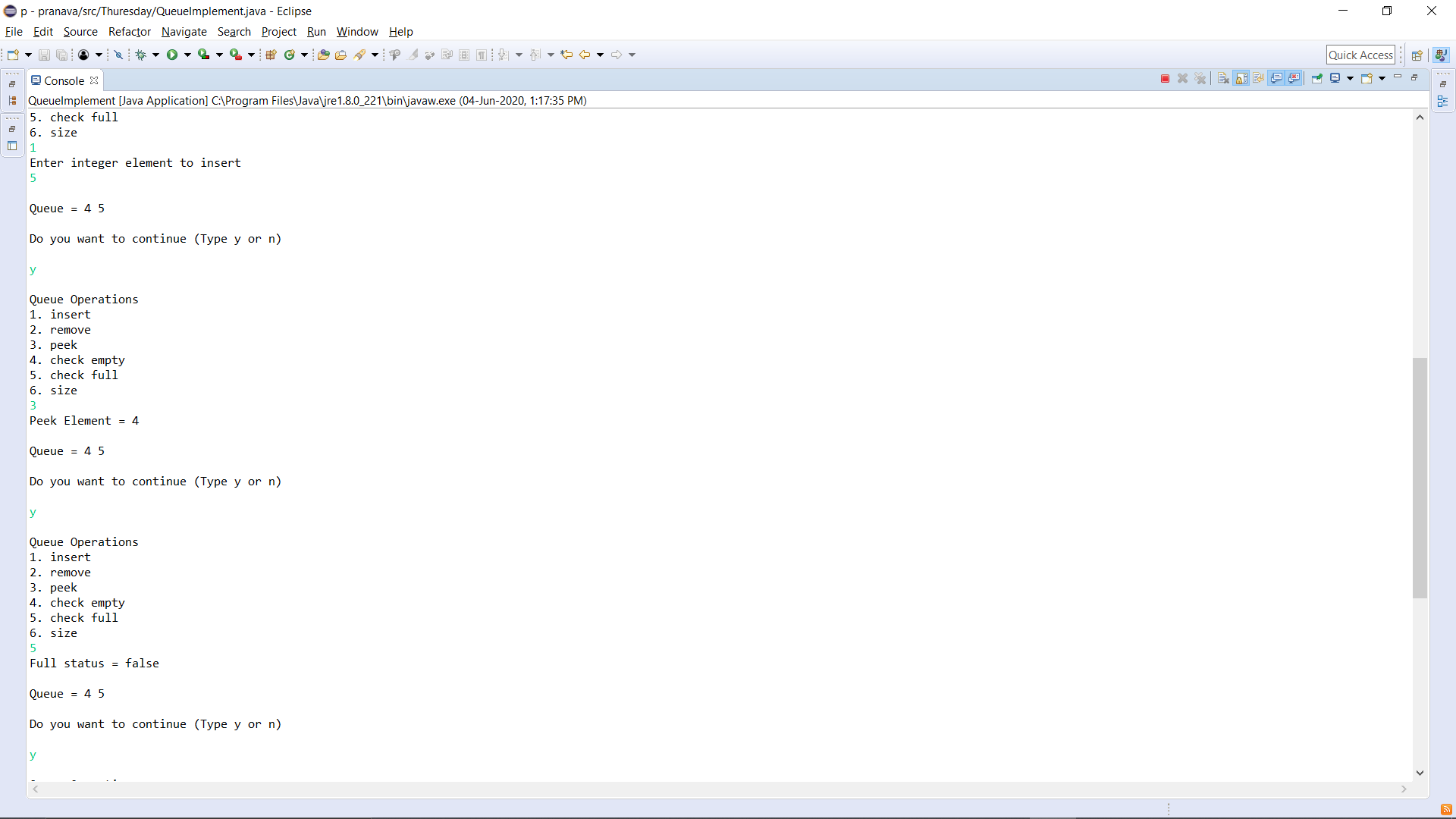
System.***out***.println();

}

}

**Output:**

****

****

