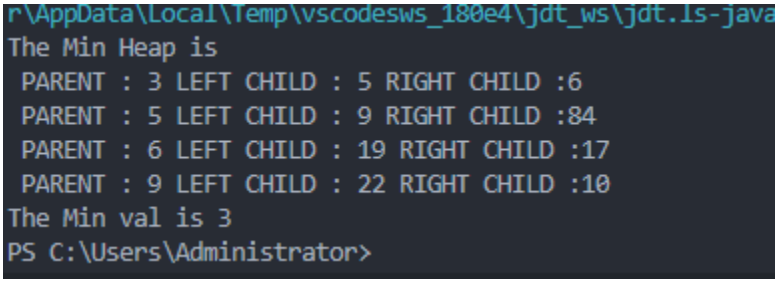
```java
public class MinHeap {

    private int[] Heap;

    private int size;

    private int maxsize;



    private static final int FRONT = 1;



    public MinHeap(int maxsize) {

        this.maxsize = maxsize;

        this.size = 0;

        Heap = new int[this.maxsize + 1];

        Heap[0] = Integer.MIN_VALUE;

    }

    private int parent(int pos) {  return pos / 2;  }

    private int leftChild(int pos) { return (2 * pos); }

    private int rightChild(int pos) { return (2 * pos) + 1; }

    private boolean isLeaf(int pos) {

        if (pos >= (size / 2) && pos <= size) { return true; }

        return false;

    }
    private void swap(int fpos, int spos) {

        int tmp;

        tmp = Heap[fpos];

        Heap[fpos] = Heap[spos];

        Heap[spos] = tmp;

```

```
    }
    private void minHeapify(int pos) {

        if (!isLeaf(pos)) {

            if (Heap[pos] > Heap[leftChild(pos)] || Heap[pos] > Heap[rightChild(pos)]) {

                if (Heap[leftChild(pos)] < Heap[rightChild(pos)]) {

                    swap(pos, leftChild(pos));

                    minHeapify(leftChild(pos));

                }

                else {

                    swap(pos, rightChild(pos));

                    minHeapify(rightChild(pos));

                }

            }

        }

    }

    public void insert(int element) {

        if (size >= maxsize) {  return;  }
        Heap[++size] = element;
        int current = size;
        while (Heap[current] < Heap[parent(current)]) {
            swap(current, parent(current));
            current = parent(current);

        }

    }



public void print()  {
     for (int i = 1; i <= size / 2; i++) {
         System.out.print(" PARENT : " + Heap[i] + " LEFT CHILD : " + Heap[2 * i] + " RIGHT CHILD :" + Heap[2 * i + 1]);
         System.out.println();
     }
```

```
99
100     }
101
102     public int remove() {
103         int popped = Heap[FRONT];
104
105         Heap[FRONT] = Heap[size--];
106
107         minHeapify(FRONT);
108
109         return popped;
110
111     }
112
113
114
        Run | Debug
115     public static void main(String[] arg) {
116
117         System.out.println(x: "The Min Heap is ");
118
119         MinHeap minHeap = new MinHeap(maxsize: 15);
120
121         minHeap.insert(element: 5);
122
123         minHeap.insert(element: 3);
124
125         minHeap.insert(element: 17);
126
127         minHeap.insert(element: 10);
128
129         minHeap.insert(element: 84);
130
131         minHeap.insert(element: 19);
132
133         minHeap.insert(element: 6);
134
135         minHeap.insert(element: 22);
136
137         minHeap.insert(element: 9);
138
139         minHeap.print();
140
141         System.out.println("The Min val is " + minHeap.remove());
142
143     }
144
145 }
```

Output:


r\AppData\Local\Temp\vscodesws_180e4\jdt_ws\jdt.ls-java
The Min Heap is
PARENT : 3 LEFT CHILD : 5 RIGHT CHILD :6
PARENT : 5 LEFT CHILD : 9 RIGHT CHILD :84
PARENT : 6 LEFT CHILD : 19 RIGHT CHILD :17
PARENT : 9 LEFT CHILD : 22 RIGHT CHILD :10
The Min val is 3
PS C:\Users\Administrator>