

MINHEAP

```
package DAY_7;
import java.util.ArrayList;
import java.util.List;

public class MinHeap {
    private List<Integer> heap;

    public MinHeap() {
        heap = new ArrayList<>();
    }

    public void insert(int value) {
        heap.add(value);
        heapifyUp(heap.size() - 1);
    }

    public int deleteMin() {
        if (heap.size() == 0) {
            throw new IllegalStateException("Heap is empty");
        }
        if (heap.size() == 1) {
            return heap.remove(0);
        }

        int minValue = heap.get(0);
        heap.set(0, heap.remove(heap.size() - 1));
        heapifyDown(0);
        return minValue;
    }

    public int getMin() {
        if (heap.size() == 0) {
            throw new IllegalStateException("Heap is empty");
        }
        return heap.get(0);
    }

    private void heapifyUp(int index) {
        int parentIndex = (index - 1) / 2;
        if (index > 0 && heap.get(index) < heap.get(parentIndex)) {
            swap(index, parentIndex);
            heapifyUp(parentIndex);
        }
    }

    private void heapifyDown(int index) {
        int leftChild = 2 * index + 1;
        int rightChild = 2 * index + 2;
        int smallest = index;

        if (leftChild < heap.size() && heap.get(leftChild) < heap.get(smallest)) {
            smallest = leftChild;
        }
        if (rightChild < heap.size() && heap.get(rightChild) < heap.get(smallest)) {
            smallest = rightChild;
        }
    }
}
```

```

    }
    if (smallest != index) {
        swap(index, smallest);
        heapifyDown(smallest);
    }
}

private void swap(int index1, int index2) {
    int temp = heap.get(index1);
    heap.set(index1, heap.get(index2));
    heap.set(index2, temp);
}

public static void main(String[] args) {
    MinHeap minHeap = new MinHeap();
    minHeap.insert(3);
    minHeap.insert(1);
    minHeap.insert(6);
    minHeap.insert(5);
    minHeap.insert(2);
    minHeap.insert(4);
    System.out.println(minHeap.heap);
    System.out.println("Min value: " + minHeap.getMin());
    System.out.println("Removed min value: " + minHeap.deleteMin());
    System.out.println("New min value: " + minHeap.getMin());
}
}

```

The screenshot shows an IDE with the following components:

- Editor:** Displays the `MinHeap.java` file. The code is as follows:


```

1 package DAY_7;
2 import java.util.ArrayList;
3 import java.util.List;
4
5 public class MinHeap {
6     private List<Integer> heap;
7
8     public MinHeap() {
9         heap = new ArrayList<>();
10    }
11
12    public void insert(int value) {
13        heap.add(value);
14        heapifyUp(heap.size() - 1);
15    }
16
17    public int deleteMin() {
18        if (heap.size() == 0) {
19            throw new IllegalStateException("Heap is empty");
20        }
21        if (heap.size() == 1) {
22            return heap.remove(0);
23        }
24
25        int minValue = heap.get(0);
26        heap.set(0, heap.remove(heap.size() - 1));
27        heapifyDown(0);
28        return minValue;
29    }

```
- Outline:** Shows the class structure:
 - DAY_7
 - MinHeap
 - heap: List<Integer>
 - MinHeap()
 - insert(int): void
 - deleteMin(): int
 - getMin(): int
 - heapifyUp(int): void
 - heapifyDown(int): void
 - swap(int, int): void
 - main(String[]): void
- Console:** Shows the output of the program:


```

[1, 2, 4, 5, 3, 6]
Min value: 1
Removed min value: 1
New min value: 2

```