Java

```
class NeighborSum {

    public NeighborSum(int[][] grid) {

    }

    public int adjacentSum(int value) {

    }

    public int diagonalSum(int value) {

    }
}

/**
 * Your NeighborSum object will be instantiated and called as such:
 * NeighborSum obj = new NeighborSum(grid);
 * int param_1 = obj.adjacentSum(value);
 * int param_2 = obj.diagonalSum(value);
 */
--------------------------------------------------------------------
```

JavaScript

```
/**
 * @param {number[][]} grid
 */
var NeighborSum = function(grid) {
};

/** 
 * @param {number} value
 * @return {number}
 */
NeighborSum.prototype.adjacentSum = function(value) {

};

/** 
 * @param {number} value
 * @return {number}
 */
NeighborSum.prototype.diagonalSum = function(value) {

};

/** 
 * Your NeighborSum object will be instantiated and called as such:
 * var obj = new NeighborSum(grid)
 * var param_1 = obj.adjacentSum(value)
 * var param_2 = obj.diagonalSum(value)
 */
```

TypeScript

```
class NeighborSum {
    constructor(grid: number[][]) {

    }
    adjacentSum(value: number): number {

    }

    diagonalSum(value: number): number {

    }
}

/**
 * Your NeighborSum object will be instantiated and called as such:
 * var obj = new NeighborSum(grid)
 * var param_1 = obj.adjacentSum(value)
 * var param_2 = obj.diagonalSum(value)
 */
```

C++

```
class NeighborSum {
public:
    NeighborSum(vector<vector<int>>& grid) {

    }

    int adjacentSum(int value) {

    }

    int diagonalSum(int value) {
    }
};

/**
 * Your NeighborSum object will be instantiated and called as such:
 * NeighborSum* obj = new NeighborSum(grid);
 * int param_1 = obj->adjacentSum(value);
 * int param_2 = obj->diagonalSum(value);
 */
--------------------------------------------------------------------
```

C#

```
public class NeighborSum {

    public NeighborSum(int[][] grid) {

    }

    public int AdjacentSum(int value) {

    }

    public int DiagonalSum(int value) {

    }
}

/**
 * Your NeighborSum object will be instantiated and called as such:
 * NeighborSum obj = new NeighborSum(grid);
 * int param_1 = obj.AdjacentSum(value);
 * int param_2 = obj.DiagonalSum(value);
 */
--------------------------------------------------------------------
```

Kotlin

```kotlin
class NeighborSum(grid: Array<IntArray>) {

    fun adjacentSum(value: Int): Int {

    }

    fun diagonalSum(value: Int): Int {

    }

}

/**
 * Your NeighborSum object will be instantiated and called as such:
 * var obj = NeighborSum(grid)
 * var param_1 = obj.adjacentSum(value)
 * var param_2 = obj.diagonalSum(value)
 */
--------------------------------------------------------------------
```

Go

```go
type NeighborSum struct {

}
```

```
func Constructor(grid [][]int) NeighborSum {

}


func (this *NeighborSum) AdjacentSum(value int) int {

}


func (this *NeighborSum) DiagonalSum(value int) int {

}


/**
 * Your NeighborSum object will be instantiated and called as such:
 * obj := Constructor(grid);
 * param_1 := obj.AdjacentSum(value);
 * param_2 := obj.DiagonalSum(value);
 */
--------------------------------------------------------------------
```