

Aufgabe 1: Magisches Quadrat (10 Punkte)

Implementieren Sie eine Methode `static boolean isMagicSquare(int[][] square)`. Diese soll prüfen, ob es sich bei einem übergebenen, zwei-dimensionalen Array `square` um ein s. g. magisches Quadrat handelt. Darunter versteht man eine quadratische Matrix, für die die Summe jeder Zeile, jeder Spalte und der beiden Diagonalen denselben Wert ergibt. Sie können davon ausgehen, dass das übergebene Array `square` quadratisch ist.

Folgendes Beispiel zeigt ein magisches Quadrat der Größe 3 x 3:

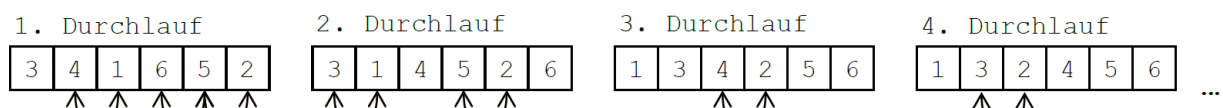
2	7	6	→15
9	5	1	→15
4	3	8	→15
15	↓15	↓15	↓15

Abzugeben ist ein Eclipse-Projekt mit dem Quelltext Ihres Java-Programms, das die Methode `isMagicSquare` sowie eine JUnit-Testmethode `testIsMagicSquare` enthält. Eine `main`-Methode ist nicht erforderlich. Verwenden Sie für die Implementierung der JUnit-Testmethode die Assert-Anweisungen `assertTrue` und `assertFalse`. Achten Sie darauf, eine vollständige Testabdeckung zu erreichen.

Aufgabe 2: Sortieren von Zahlen in einem Array (14 Punkte)

Um ein Array aufsteigend zu sortieren, kann man so vorgehen, dass man das Array zunächst einmal von links nach rechts durchläuft und in jedem Schritt das aktuelle Element mit seinem rechten Nachbarn vergleicht. Falls die beiden Elemente das Sortierkriterium verletzen, werden sie getauscht. Will man das Array beispielsweise aufsteigend sortieren, so steht nach dem 1. Durchlauf das größte Element ganz rechts im Array. Dieser Vorgang wird nun so lange wiederholt, bis das gesamte Array vollständig sortiert ist, bis also keine Vertauschungen mehr möglich sind. Beachten Sie, dass das letzte Element des jeweils vorherigen Durchlaufes nicht mehr betrachtet werden muss.

Folgendes Beispiel verdeutlicht die Vorgehensweise:



Schreiben Sie ein Programm, das Folgen von ganzen Zahlen zeilenweise aus einer Datei einliest, diese jeweils in einem Array speichert, die Arrays aufsteigend sortiert und die Ergebnisse in eine neue Datei ausgibt. In der Ein- und Ausgabedatei seien die Zahlen in folgendem Format gespeichert:

- jede Zeile speichert eine Folge von Zahlen
- die erste Zahl einer Zeile bestimmt die Länge n der Folge
- darauf folgen die n Zahlen, welche durch Leerzeichen getrennt sind

Beispiel:

Eingabedatei *input.txt*:

```
0
1 -30
2 500 -2
3 -2 -10000 200
5 17 17 17 17 17
6 1 2 3 4 5 6
8 70 60 50 40 30 20 10 0
...
```

Ausgabedatei *output.txt*:

```
0
1 -30
2 -2 500
3 -10000 -2 200
5 17 17 17 17 17
6 1 2 3 4 5 6
8 0 10 20 30 40 50 60 70
...
```

Implementieren Sie den oben beschriebenen Sortieralgorithmus in einer Methode *static void sort(int[] a)*. Testen Sie die Methode mit Hilfe der in der Lernplattform bereitgestellten JUnit-Testmethode *public void testSort()*.

Sie finden in der Lernplattform außerdem eine Eingabedatei *input.txt*, die Sie anschließend in Ihrer *main*-Methode einlesen und sortiert in eine Datei *output.txt* ausgeben sollen. Sie können davon ausgehen, dass die Eingabedatei korrekte Daten enthält. Verwenden Sie für das Einlesen und Ausgeben des Arrays ebenfalls Methoden.

Hinweis:

Die Methode *main* könnte wie folgt aussehen:

```
public static void main(String[] args) {
    In.open("input.txt");
    Out.open("output.txt");

    int row[] = readRow();
    while (row != null) {
        sort(row);
        print(row);
        row = readRow();
    }

    In.close();
    Out.close();
}
```

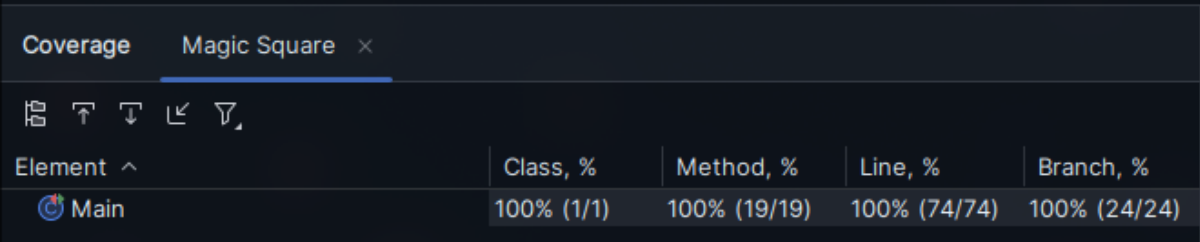
Abzugeben ist ein Eclipse-Projekt mit dem Quelltext Ihres Java-Programms (inklusive der JUnit-Testmethode) sowie der verwendeten Ein- und Ausgabedatei.

Aufgabe 1 - Magisches Quadrat


Aufgabe 1/1: Java Program

Der Quellcode des Programms kann den beigefügten Dateien im Ordner “magic-square” entnommen werden.

Durch das Aufteilen der Tests und das Testen aller ausgelagerten Methoden wurde eine komplette Testabdeckung erreicht.



The screenshot shows the 'Coverage' window in IntelliJ IDEA for a project named 'Magic Square'. The window has a tab labeled 'Magic Square' with a close button. Below the tab are icons for coverage visualization: a grid, a line graph, a bar chart, a pie chart, and a funnel. A table displays the coverage statistics for the 'Main' class.

Element ^	Class, %	Method, %	Line, %	Branch, %
 Main	100% (1/1)	100% (19/19)	100% (74/74)	100% (24/24)

Aufgabe 2 - Sortieren von Zahlen

Aufgabe 2/1: Java Program

Der Quellcode des Programms kann den beigefügten Dateien im Ordner “number-sorting” entnommen werden.