Testat 1 der Vorlesung Programmierung

Dieses Testat besteht aus einer Klasse mit dem Namen Testat1A, die mehrere statische Methoden enthält. Die Signaturen der Methoden dürfen Sie nicht verändern. Darüber hinaus gibt es eine main-Methode, mit der Sie die einzelnen Aufgaben testen können. Den Inhalt der main-Methode können Sie beliebig ändern.

Aufgabe 1 (27 Punkte)
In der Methode **computeWithSwitch** finden Sie ein Switch-Statement, das auf Basis des Eingabeparameters **value** eine Ausgabe berechnet. Implementieren Sie die Methode **computeWithIf** so, dass diese Methode für alle Eingaben die gleiche Ausgabe wie **computeWithSwitch** erzeugt. Bei Ihrer Implementierung dürfen Sie jedoch **kein Switch-Statement** verwenden. D.h. Sie müssen eine andere geeignete Kontrollstruktur verwenden.

Aufgabe 2 (27 Punkte)
In der Methode **fillWithFor** finden Sie eine For-Schleife, die allen Elementen des übergebenen Arrays bestimmte Werte zuweist. Implementieren Sie die Methode **fillWithWhile** so, dass diese Methode das übergebene Array mit denselben Werten befüllt wie **fillWithFor**. Bei Ihrer Implementierung dürfen Sie jedoch **keine For-Schleife** verwenden. D.h. Sie müssen eine andere Kontrollstruktur verwenden.

Aufgabe 3 (27 Punkte)
Implementieren Sie die Methode **minArray** so, dass diese den minimalen Wert zurückliefert, der im übergebenen Array enthalten ist. Bei Ihrer Implementierung können Sie davon ausgehen, dass das Array mindestens ein Element enthält. Im Gegensatz zu Aufgabe 1 und 2 gibt es keine Einschränkungen bei der Nutzung der in Java verfügbaren Kontrollstrukturen.

Aufgabe 4 (19 Punkte)
Die Methode **sumRows** erhält als Eingabe ein 2-dimensionales Array beliebiger Größe. Die erste Dimension des Arrays adressiert die Spalten, die zweite Dimension die Zeilen. D.h. int x = array[0][1] weist x den Wert der Spalte 0 und Zeile 1 zu. Implementieren Sie die Methode so, dass diese ein eindimensionales Array zurückgibt bei dem der i-te Wert die Summe der i-ten Zeile des übergebenen Arrays enthält. Bei Ihrer Implementierung können Sie davon ausgehen, dass das übergebene Array mindestens einen Wert enthält. Weiterhin können Sie davon ausgehen, dass die Anzahl der Zeilen in jeder Spalte gleich ist. Wie in Aufgabe 3 gibt es auch hier keine Einschränkungen bei der Nutzung der in Java verfügbaren Kontrollstrukturen.

1	2	3
4	6	7
3	2	3

Beispieleingabe

1 + 2 + 3 = 6
4 + 6 + 7 = 17
3 + 2 + 3 = 8

Rückgabewert

Testat 1 der Vorlesung Programmierung

Dieses Testat besteht aus einer Klasse mit dem Namen Testat1B, die mehrere statische Methoden enthält. Die Signaturen der Methoden dürfen Sie nicht verändern. Darüber hinaus gibt es eine main-Methode, mit der Sie die einzelnen Aufgaben testen können. Den Inhalt der main-Methode können Sie beliebig ändern.

Aufgabe 1 (27 Punkte)
In der Methode **computeWithIf** finden Sie ein If-Statement, das auf Basis des Eingabeparameters **value** eine Ausgabe berechnet. Implementieren Sie die Methode **computeWithSwitch** so, dass diese Methode für alle Eingaben die gleiche Ausgabe wie **computeWithIf** erzeugt. Bei Ihrer Implementierung dürfen Sie jedoch **kein If-Statement** verwenden. D.h. Sie müssen eine andere geeignete Kontrollstruktur verwenden.

Aufgabe 2 (27 Punkte)
In der Methode **fillWithWhile** finden Sie eine While-Schleife, die allen Elementen des übergebenen Arrays bestimmte Werte zuweist. Implementieren Sie die Methode **fillWithFor** so, dass diese Methode das übergebene Array mit denselben Werten befüllt wie **fillWithWhile**. Bei Ihrer Implementierung dürfen Sie jedoch **keine While-Schleife** verwenden. D.h. Sie müssen eine andere Kontrollstruktur verwenden.

Aufgabe 3 (27 Punkte)
Implementieren Sie die Methode **maxArray** so, dass diese den maximalen Wert zurückliefert, der im übergebenen Array enthalten ist. Bei Ihrer Implementierung können Sie davon ausgehen, dass das Array mindestens ein Element enthält. Im Gegensatz zu Aufgabe 1 und 2 gibt es keine Einschränkungen bei der Nutzung der in Java verfügbaren Kontrollstrukturen.

Aufgabe 4 (19 Punkte)
Die Methode **sumColumns** erhält als Eingabe ein 2-dimensionales Array beliebiger Größe. Die erste Dimension des Arrays adressiert die Spalten, die zweite Dimension die Zeilen. D.h. int x = array[0][1] weist x den Wert der Spalte 0 und Zeile 1 zu. Implementieren Sie die Methode so, dass diese ein eindimensionales Array zurückgibt bei dem der i-te Wert die Summe der i-ten Spalte des übergebenen Arrays enthält. Bei Ihrer Implementierung können Sie davon ausgehen, dass das übergebene Array mindestens einen Wert enthält. Weiterhin können Sie davon ausgehen, dass die Anzahl der Zeilen in jeder Spalte gleich ist. Wie in Aufgabe 3 gibt es auch hier keine Einschränkungen bei der Nutzung der in Java verfügbaren Kontrollstrukturen.

1	2	3
4	6	7
3	2	3

Beispieleingabe

1 + 4 + 3 = 8
2 + 6 + 2 = 10
3 + 7 + 3 = 13

Rückgabewert