

Programmierpraktikum WS 20/21

Prüfungsaufgabe

23.11.2020

Bitte beachten: Sie müssen Ihre Lösung bis spätestens 15:15 Uhr in Ihr persönliches git-Repository hochladen.

Aufgabenbeschreibung

Mit einem Java-Programm sollen Sie folgendes Problem lösen:

Gegeben ist ein zweidimensionales 6x6 Feld mit `int`-Werten! Wir suchen innerhalb dieses Feldes eine Sanduhren-Struktur, wo die Summe der einzelnen Werte am größten ist! Eine Sanduhr-Struktur sieht wie folgt aus:

```
a b c
  d
e f g
```

Beispiel:

```
1 1 1 0 0 0
0 1 0 0 0 0
1 1 1 0 0 0
0 0 0 0 0 0
0 0 0 0 0 0
0 0 0 0 0 0
```

Nachfolgend sind alle Felder aufgeführt die eine Sanduhr-Struktur aus dem o.a. Beispiel ableiten.

```
1 1 1      1 1 0      1 0 0      0 0 0
  1          0          0          0
1 1 1      1 1 0      1 0 0      0 0 0

0 1 0      1 0 0      0 0 0      0 0 0
  1          1          0          0
0 0 0      0 0 0      0 0 0      0 0 0

1 1 1      1 1 0      1 0 0      0 0 0
  0          0          0          0
0 0 0      0 0 0      0 0 0      0 0 0

0 0 0      0 0 0      0 0 0      0 0 0
  0          0          0          0
0 0 0      0 0 0      0 0 0      0 0 0
```

Von diesen Strukturen ist die erste Sanduhr die mit der maximalen Summe der Einzelwerte (7) der „Gewinner“. Beachten Sie bitte, dass dies nur ein Beispiel ist und jedes Arrayelement ein eigener `int`-Wert sein kann (nicht nur 0 oder 1)!!!

Aufgabenstellung

Gegeben ist bereits das Hauptprogramm `Main`. Sie sollen in dem Hauptprogramm `Main.java` nichts ändern!!

Erstellen Sie eine neue Java-Klasse `YourClock`, in der Sie dann eine Methode `getMaxValue` anlegen die als Parameter ein zweidimensionales `int`-Array erhält und einen einzelnen `int`-Wert zurückgibt (siehe Aufruf der Methode in `Main.java`). Das zweidimensionale `int`-Array ist dabei ein `6x6 int`-Array so wie oben angegeben. Sie implementieren dann die Methode `getMaxValue` nach der o.a. Aufgabenbeschreibung. Als Rückgabewert geben Sie dann den größten Wert aller möglichen Sanduhren zurück.

Im `Main`-Programm wird eine Instanz Ihrer Klasse `YourClass` erstellt und mit dem o.a. Beispielarray aufgerufen. Sie sehen dann ob ihre Lösung richtig ist. Starten Sie das Programm mit `java Main` nachdem Sie die `Main.java` und `YourClock.java` kompiliert haben!

Aus diesem Grunde ist es wichtig, dass Sie sich genau an die o.a. Vorgaben halten und exakt die vorgegeben Klassennamen und Methodennamen verwenden!!! Denn wir werden zur Überprüfung Ihrer Lösung Ihre Klasse `YourClock` mit weiteren Testfällen aufrufen!!!

Hinweise:

Sie können selbstverständlich die Java-API online verwenden. Ob es hilfreich ist, im Internet nach Lösungen zu suchen oder per email, WhatsApp usw. um Hilfe zu ersuchen ist nicht nur aufgrund der kurzen Bearbeitungszeit fraglich. Die Lösungen werden auf Duplikate überprüft. Wenn Duplikate gefunden werden, werden sowohl das Original als auch die Duplikate als nicht bestanden bewertet. Sie können das ProPra dann erst im SoSe 2021 wieder besuchen.

Geben Sie auch eine Lösung ab, selbst wenn diese noch nicht vollständig ist bzw. fehlerhaft ist. Wir bewerten alle Lösungen die abgegeben werden!! Sie können also auch bestehen, selbst wenn die Lösung nicht vollständig ist bzw. nicht korrekt funktioniert!!!

Bitte denken Sie daran Ihre Lösung in Ihr eigenes git-Repository (ppXYZ) hochzuladen. Legen Sie dazu das Verzeichnis `pruefung` in ihrem Repo an. Laden Sie dazu Ihre Lösungsdatei `YourClock.java` in ihr git-Repo hoch. Wichtig ist das Sie die `.java` Datei Ihrer Lösung hochladen!!!!

Eine Abgabe per email o.ä. wird NICHT akzeptiert!

Damit Sie selbst überprüfen können, ob das Hochladen in Ihr git-Repo funktioniert hat können Sie einfach in ein neues Verzeichnis auf ihrem Rechner wechseln und dann ihr git-Repo nochmal dorthin klonen! Dann sehen Sie ob Ihre Lösung vorhanden ist oder nicht!

Sollten Sie eine **Verständnisfrage** haben so können Sie die während der Prüfung im Mattermost Chat stellen (ProPra WS 20/21).

Viel Erfolg!