

Vorbereitung:

```
*
**
***
****
*****
*****
*****
*****
*****
*****
*****
```

```
    *
  ***
 *****
*****
*****
*****
  **
```

```
    0
  00*
 *00**
***0***
*****00
  ||
```

Geben Sie die gewünschte Baumhöhe ein: 10

```

  *
 ***
*****
O*****O
*****
**OO*OO*OO*
O*OOOO*O*O***
*****O*O*****
*OO*****OO*****O*O*O
*****O*O*O*O*****O*O*O*
  | | | |
  | | | |
  | | | |
```

Tannenbaum mit Python - Grundaufgabe

Implementieren Sie ein Python-Programm, das einen Tannenbaum auf der Konsole ausgibt. Der Tannenbaum soll verschiedene Farben für Sterne und Kugeln verwenden. Verwenden Sie dazu die random-Bibliothek für die zufällige Platzierung von Kugeln.

Schritt 1: Benutzereingabe für Baumhöhe

Implementieren Sie die Benutzereingabe, um die gewünschte Baumhöhe abzufragen. Verwenden Sie die input-Funktion, um die Eingabe des Benutzers zu lesen.

Schritt 2: Tannenbaum-Zweige erstellen

Nutzen Sie eine Schleife, um die Zweige des Tannenbaums zu erstellen. Verwenden Sie zwei verschachtelte Schleifen: eine für die Leerzeichen vor den Symbolen und eine für die Symbole selbst. Achten Sie darauf, dass die Symbole zufällig entweder Kugeln (o) oder Sterne (*) sein können. Verwenden Sie ANSI Escape Codes, um die Farben in der Konsole zu ändern.

Schritt 3: Stamm des Tannenbaums erstellen

Implementieren Sie den Stamm des Tannenbaums. Berechnen Sie die Höhe und Breite des Stamms basierend auf der Baumhöhe des Benutzers. Stellen Sie sicher, dass der Stamm unterhalb der Baumkrone zentriert ist und verwenden Sie eine andere Farbe für den Stamm.

Schritt 4: Kommentare und Struktur

Fügen Sie sinnvolle Kommentare im Code hinzu, um die Funktionalität und Struktur des Programms zu erklären.

Schritt 5: Testen

Testen Sie Ihr Programm mehrmals, um sicherzustellen, dass es korrekt funktioniert.

Tannenbaum Java - Grundaufgabe

Erstellen Sie ein Java-Programm, das einen Tannenbaum auf der Konsole ausgibt.

Schritt 1: Grundgerüst aufbauen

Erstellen Sie ein Grundgerüst für das Java-Programm. Importieren Sie die notwendige **Scanner**-Klasse und erstellen Sie die Hauptklasse **Tannenbaum**. Fügen Sie die **main**-Methode hinzu.

Schritt 2: Benutzereingabe für Baumhöhe

Implementieren Sie die Benutzereingabe für die gewünschte Baumhöhe. Verwenden Sie dazu die **Scanner**-Klasse, um die Eingabe des Benutzers zu lesen.

Schritt 3: Tannenbaum-Zweige erstellen

Nutzen Sie eine Schleife, um die Zweige des Tannenbaums zu erstellen. Verwenden Sie zwei verschachtelte Schleifen: eine für die Leerzeichen vor den Sternen und eine für die Sterne selbst. Beachten Sie dabei die Anpassung der Schleifenbedingungen basierend auf der Benutzereingabe.

Schritt 4: Zufällige Kugeln hinzufügen

Fügen Sie innerhalb der Schleife, die die Sterne erstellt, die Möglichkeit hinzu, dass mit einer Wahrscheinlichkeit von 30% eine Kugel hinzugefügt wird. Verwenden Sie dazu **Math.random()**. Achten Sie auf die Unterscheidung der Farben für die Kugeln (rot, blau) und die Sterne (grün).

Schritt 5: Stamm des Tannenbaums erstellen

Implementieren Sie den Stamm des Tannenbaums. Berechnen Sie die Höhe und Breite des Stamms basierend auf der Baumhöhe des Benutzers. Stellen Sie sicher, dass der Stamm unterhalb der Baumkrone zentriert ist.

Schritt 6: Farbige Ausgabe in der Konsole

Verwenden Sie ANSI Escape Codes, um die Farben in der Konsole zu ändern. Achten Sie darauf, die Farben korrekt zu setzen und nach Bedarf zurückzusetzen.

Schritt 7: Programm testen

Führen Sie das Programm aus, indem Sie die Baumhöhe eingeben, und überprüfen Sie, ob der Tannenbaum mit korrekten Farben und Stamm erstellt wird.

Schritt 8: Kommentare hinzufügen

Fügen Sie sinnvolle Kommentare im Code hinzu, um die Funktionalität und Struktur des Programms zu erklären.

Schritt 9: Code-Optimierung

Überprüfen Sie den Code auf mögliche Optimierungen. Überlegen Sie, wie Sie den Code lesbarer und effizienter gestalten könnten.

Schritt 10: Programmieren und Testen

Vervollständigen Sie den Code entsprechend den oben genannten Schritten und testen Sie das Programm mehrmals, um sicherzustellen, dass es korrekt und wie erwartet funktioniert.

Zusatzaufgaben

Baum auf Kopf drehen:

```
      | | | |
      | | | |
      | | | |
*0**0***0*0**0*****
 0*0*0***00***00*
 ***0*0*****
 ***0*0*000*0
  *0*00*00**
  *****
  0*****0
  *****
   *****
    ***
     *
```

Eigene Kugeln wählen:

Wieviele Reihen mit Ästen?

10

Zeichen für Kugel 1 eingeben

o

Zeichen für Kugel 2 eingeben

a

```
|
      o
     *ao
    ****o
   ****oa*
  *****o**
 *aoa*a***ao
*ao***oao*****
***a*a***a**oa*
*****oa*****oa*
*****oao*oaoao
      #####
      #####
      #####
```

Kugeln und Farben wählen

Wieviele Reihen mit Ästen?

10

Zeichen für Kugel 1 eingeben

o

Farbe für Kugel 1 eingeben: Lila(0), Rot(1), Blau(2), Gelb (3)

0

Zeichen für Kugel 2 eingeben

&

Farbe für Kugel 2 eingeben: Lila(0), Rot(1), Blau(2), Gelb (3)

2

```
|
      o
      ***
      *&*&o
      ***&***o
      o***o&***
      **o&o&***o**
      *&*&***o&*****
      *&*****&*&*
      *****&*****o**&*&o
      o***o&*&o*****&*
```


#####

