

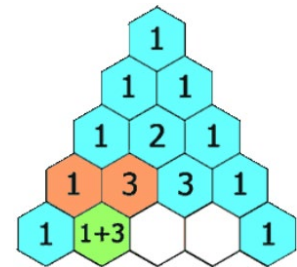
5. Praktikumsaufgabe Felder

Prof. Dr. Uwe Wienkop

In dieser Woche soll der Umgang mit Feldern geübt werden. Programmieren Sie hierfür die folgenden Funktionen:

1 Pascalsches Dreieck

Das **Pascalsche Dreieck** ist eine Form der grafischen Darstellung der Binomialkoeffizienten. Sie sind im Dreieck derart angeordnet, dass jeder Eintrag die Summe der zwei darüberstehenden Einträge ist. (https://de.wikipedia.org/wiki/Pascalsches_Dreieck)



Programmieren Sie eine Funktion `static int[] PascalDreieck(int[] f)`, die ein `int`-Feld `f` als Parameter erhält (Eine Zeile des Dreiecks) und hierzu die nächste Zeile bestimmt und diese als Ergebnis zurückliefert. Für das Eingangsfeld `{1, 2, 1}` würde somit das Ergebnisfeld `{1, 3, 3, 1}` zurückgeliefert werden. Wenn keine Eingabe geliefert wurde, ist das Feld mit dem einen `1` – Element zurück zu liefern.

2 Ausgabe

Programmieren Sie eine zweite Funktion `static void Ausgabe(int[] f)`, die alle Elemente des Feldes in einer Zeile ausgibt und dann die Zeile mit einem `Console.WriteLine()`; abschließt.

3 Primzahlen

Schreiben Sie eine Funktion, welche die Primzahlen bis zu einer Grenze `n` (inklusive) berechnet und die Anzahl der Primzahlen bis zu dieser Grenze zurückliefert.

`static int Primzahlen(int n, bool ausgabe = false)`

Wenn das `ausgabe`-Flag auf `true` gesetzt ist, sollen die Primzahlen auch auf dem Bildschirm ausgegeben werden. Im Default-Fall (`false`) ist nur die Anzahl zurückzuliefern.

Bei der Berechnung der Primzahlen ist zwingend das Verfahren „Sieb des Eratosthenes“ zu verwenden – Es soll ja der Umgang mit Feldern geübt werden!

1. Legen Sie ein `bool`-Feld `Primzahlen` mit den Elementen `0..n` an
2. 2 ist die erste Primzahl. Also sind alle Vielfachen von 2 keine Primzahlen. Diese Zahlen können somit im `bool`-Feld entsprechend als „keine Primzahlen“ markiert werden
3. Nun ist die nächste unmarkierte Zahl zu suchen. Diese Zahl muss wieder eine Primzahl sein, denn sonst wäre sie ja bereits markiert worden. Auch von dieser Zahl sind dann alle Vielfachen als „keine Primzahlen“ zu markieren, usw.
4. Zählen Sie bei den gefundenen Primzahlen mit und geben Sie diese je nach `ausgabe`-Flag ggf. aus.

Bei der Korrektur ist zu überprüfen, ob dieses Verfahren angewendet wurde! Ansonsten sind in der Kategorie Funktionalität nur noch 20 Punkte erreichbar!

Zu den Feldern finden Sie im Prog-CMS alle benötigten Informationen (VPN/Intranet):

<http://cms.informatik.fh-nuernberg.de/programmierung/index.php/prog-1/uebersicht-arrays/>

Verwenden Sie als Test folgendes Main():

```
static void Main(string[] args)
{
    Console.WriteLine("Pascalsches Dreieck:");
    int[] f=null;
    for (int i = 0; i < 8; i++)
    {
        f = PascalDreieck(f);
        Ausgabe(f);
    }
    Console.WriteLine();

    Console.WriteLine("Primzahlen bis 20");
    Primzahlen(20, true);
    const int Primzahlgrenze = 750000;
    Console.WriteLine($"Anzahl der Primzahlen bis {Primzahlgrenze}: {Primzahlen(Primzahlgrenze)}");
}
```

Dieses Main soll die folgenden Ausgaben produzieren:

```
Pascalsches Dreieck:
1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
1 5 10 10 5 1
1 6 15 20 15 6 1
1 7 21 35 35 21 7 1

Primzahlen bis 20
    2    3    5    7   11   13   17   19
Anzahl der Primzahlen bis 750000: 60238
```