

**Aufgabe 1:** Eine Funktion  $f$  sei definiert durch:

$$f(n) = \begin{cases} f(n-1) + 3n(n-1) + 1 & \text{für } n > 1 \\ 1 & \text{sonst} \end{cases}$$

a) Füllen Sie die Wertetabelle mit den dazugehörigen Funktionswerten!

$n$		1		2		3		4		5		6
$f(n)$												

b) Welcher mathematische Zusammenhang besteht zwischen  $n$  und dem Funktionswert  $f(n)$ ?

c) Schreiben Sie eine rekursive Java-Funktion, die den Funktionswert  $f(n)$  berechnet.

```
int f(int n) { Rekursiver Algorithmus }
```

d) Schreiben Sie nun eine iterative Java-Funktion, die den Funktionswert  $f(n)$  berechnet.

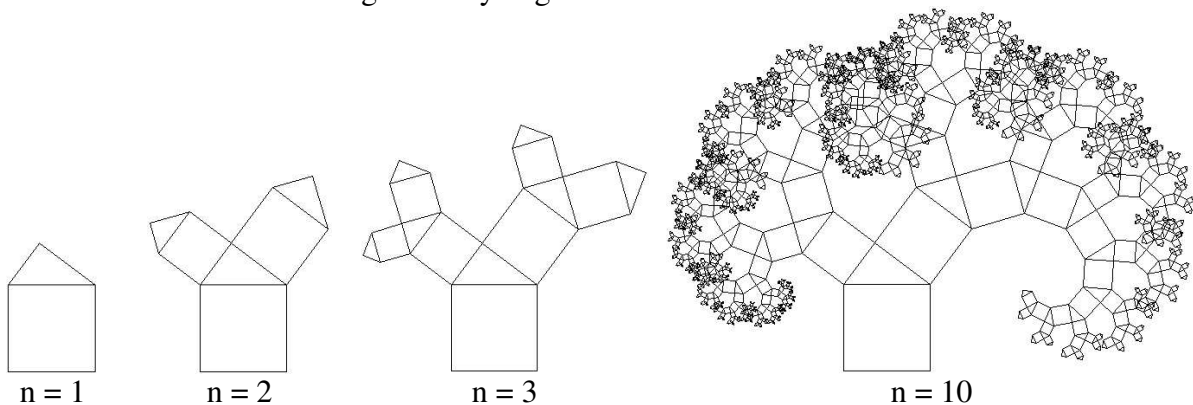
```
int f_it(int n) { Iterativer Algorithmus }
```

**Hinweis:** Iterativ heißt, dass die Rekursion durch eine Schleife ersetzt wird.

**Aufgabe 2:** Eine Rekursive Java-Prozedur

```
void pythagoras_Baum(int n, double hypotenuse)
```

zeichnet den folgenden Pythagoras-Baum:



a) Geben Sie die Implementierung der Prozedur an.

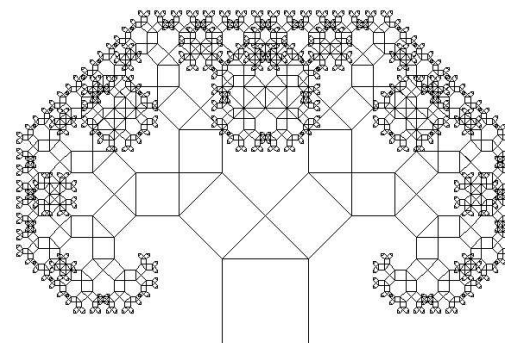
**Hinweis:** die rechtwinkligen Dreiecke haben stets die Winkel  $\gamma = 90^\circ$ ,  $\alpha = 53,1301^\circ$  und

$$\beta = 36,8699^\circ \text{ sowie die Seitenverhältnisse } \frac{a}{c} = \frac{4}{5} \text{ und } \frac{b}{c} = \frac{3}{5}.$$

b) Verallgemeinern Sie die Grafik, indem Sie dem Prozedurkopf eine Angabe für den Winkel  $\alpha$  hinzufügen:

```
void pythagoras_Baum(int n,
                    double hypotenuse,
                    double alpha)
```

Der Aufruf `pythagoras_Baum(10, 100, 53.1301)` würde das gleiche Ergebnis wie zuvor liefern, der Aufruf `pythagoras_Baum(10, 100, 45)` würde dagegen den rechts abgebildeten Baum erzeugen:



**Aufgabe 3:** Im *Morsealphabet* stehen zwei Zeichen, nämlich *Punkt* und *Strich* zur Verfügung. Dabei benötigt der Punkt *eine* Zeiteinheit, der Strich *zwei* Zeiteinheiten zur Übertragung.

- a) Die folgende Prozedur gibt alle Zeichenfolgen aus Punkt und Strich aus, welche in  $n$  Zeiteinheiten übertragbar sind:

```
void ausgabe(int n, String zeichenkette)
{
    if (n == 0)
        JTextArea1.append(zeichenkette+"\n");
        // In TextArea wird eine neue Zeile, die bis hier
        // zusammengesetzte Zeichenkette, ausgegeben.
        // REKURSIONSANKER
    else {
        // Es stehen noch Zeiteinheiten zur Verfügung,
        // also muss die Rekursion neu gestartet werden.
        ausgabe(n-1, zeichenkette+ "•");
        if (n > 1)
            ausgabe(n-2, zeichenkette+ "-");
    }
}
```

Welche Ausgabe erfolgt im TextArea, wenn die Prozedur mit dem Aufruf `ausgabe(5, "")` gestartet wird?

**Hinweis:** Es werden insgesamt 8 Zeilen im TextArea ausgegeben. Beachten Sie dabei die Reihenfolge der Morsecodes, welche von der Prozedur erzeugt werden.

- b) Wie viele verschiedene Zeichenfolgen (Nachrichten) aus Punkt und Strich lassen sich in  $n$  Zeiteinheiten übertragen? Schreiben Sie eine rekursive Funktion, welche diese Anzahl ermittelt:

```
int anzahl(int n)
{
    // ... hier kommen ihre Programmzeilen hin ...
}
```

**Aufgabe 4:** Gegeben ist die folgende Java-Funktion:

```
int wbi(int a, int b)
{
    if (b == 0)
        return 1;
    else if (b == 1)
        return a;
    else
        return wbi(a,b / 2) * wbi(a,b-b / 2);
}
```

- a) Beschreiben Sie den (mathematischen) Zusammenhang zwischen dem Funktionsergebnis `wbi` und den Variablen `a` und `b`, wobei  $a, b \geq 0$  sein sollen.
- b) Stellen Sie den rekursiven Abstieg des Funktionsaufrufs `wbi(10,7)` baumartig dar. und bestätigen Sie damit die Aussage aus Teilaufgabe a.