

Übungsblatt 2

Fragen ?

Übung zur VL „Grundlagen der Programmierung“

6. Übung

Dr. Zubow

Zahlendarstellung

- // konvertiere einen String aus 0 und 1 in ein Integer
- `int fromBinaryString(String s)`
- // konvertiere ein int in ein String aus 0 und 1
- `String toBinaryString(int n)`

Suchverfahren

- Lineare Suche
- Binäre Suche
 - Was muss gelten?
 - Siehe: g0a demo-binary-search.pptx
- Ges.: Groovy-Implementierung
- Welche Komplexität liegt vor?

Russische Bauernmultiplikation

- Die Multiplikation von natürlichen Zahlen ($m, n > 0$) kann man wie folgt auf das Addieren, Subtrahieren, Halbieren und Verdoppeln zurückführen (Russische Bauernmultiplikation):

$$m \cdot n = \begin{cases} m & \text{falls } n=1 \\ m \cdot (n-1) + m & \text{falls } n>1, \text{ ungerade} \\ 2m \cdot \frac{n}{2} & \text{falls } n \text{ gerade} \end{cases}$$

- Berechnen Sie $257 \cdot 37$ und geben Sie alle Zwischenergebnisse an!
- Schreiben Sie eine Funktion ohne Verwendung des Multiplikationsoperators, das die obige Rechenregel benutzt.

Russische Bauernmultiplikation (2)

- Bsp.

```
257 * 37
= 257 + 257*36
= 257 + 514*18
= 257 + 1028*9
= 257 + 1028 + 1028*8
= 257 + 1028 + 2056*4
= 257 + 1028 + 4112*2
= 257 + 1028 + 8224
= 9509
```

Restprodukt	Summand
257*37	
257*36	257
514*18	
1028*9	
1028*8	1028
2056*4	
4112*2	
8224*1	
	8224
Summe	9509

$$m \cdot n = \begin{cases} m & \text{falls } n=1 \\ m \cdot (n-1) + m & \text{falls } n>1, \text{ ungerade} \\ 2m \cdot \frac{n}{2} & \text{falls } n \text{ gerade} \end{cases}$$

Russische Bauernmultiplikation (3)

- In Groovy:

```
def pawn_multiply(int m, int n) {
    if (n == 1) {
        return m;
    }
    int summe = 0;
    while (n > 0) {
        if (n % 2 == 1) {
            summe += m;
        }
        n = n / 2;
        m = m * 2;
    }
    return summe;
}

pawn_multiply(257, 37)
```

Stars

- Schreiben Sie eine Funktion, welche die natürlichen Zahlen von 1 bis 99 wie folgt ausgibt:
 - Zahlen, die entweder durch 7 teilbar sind oder deren Quersumme gleich 7 ist, werden durch einen Stern (*) ersetzt, alle anderen werden einfach ausgegeben.

```
int s,j;

for (def i = 1; i <= 99; i++) {
    // Fuer alle Zahlen im Intervall [1,99]
    s = 0; j = i;
    while (j > 0) { //solange j noch Ziffern >0 enthaelt
        s = s + (j % 10); //letzte Stelle von j zu Quersumme s addieren
        j = j / 10; //letzte Stelle von j abschneiden
    }
    if ( (s == 7) || /*Wenn Quersumme=7 oder*/
        (i % 7 == 0) ) { /*Zahl durch 7 teilbar,*/
        print(" *"); /*dann "*" ausgeben,*/
    } else {
        print(i); /*sonst die Zahl selbst*/
    }
    println();
}
```