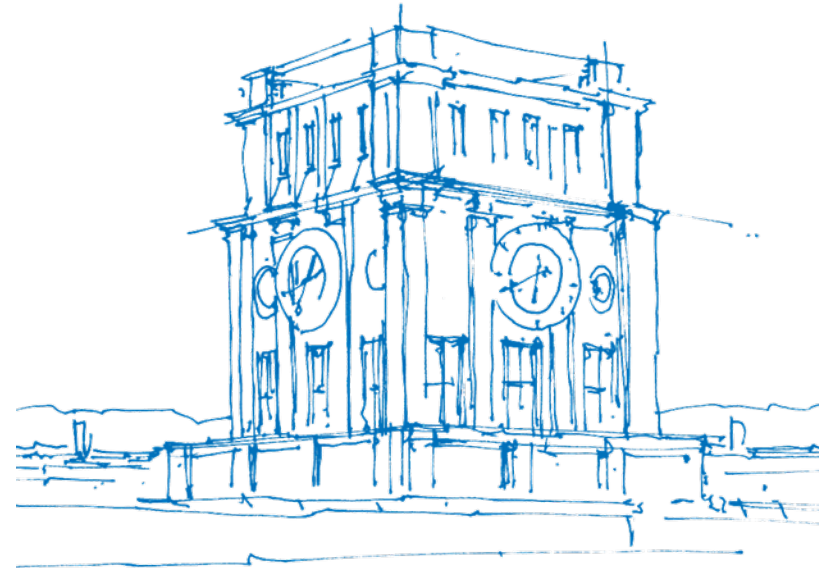


Praktikum: Grundlagen der Programmierung

Wenjie Hou

Technical University of Munich
Fakultät für Informatik

3. Tutorübung



TUM Uhrenturm

P01: Syntaxbaum

For the following MiniJava program, draw the syntax tree from the lecture according to the MiniJava grammar.

```
1  int x,r;  
2  int n;  
3  r = 1;  
4  n = 1;  
5  x = readInt();  
6  while (n < x) {  
7      if (r % 1 == 0)  
8          r = r * n;  
9      else {  
10         r = r * (-n);  
11     }  
12     n = n + 1;  
13     write (r);  
14 }
```

P02: Binäre Zahlen

Regular Expression

Metacharacter	Description
?	Matches the preceding element zero or one time.
*	Matches the preceding element zero or more times.
	The choice (also known as alternation or set union) operator matches either the expression before or the expression after the operator.
{ <i>m</i> , <i>n</i> }	Matches the preceding element at least <i>m</i> and not more than <i>n</i> times.

P02: Binäre Zahlen

Regular Expression

Problem statement: See Artemis <https://artemis.ase.in.tum.de/overview/37/exercises/789>

P02: Binäre Zahlen

Regular Expression

Problem statement: See Artemis <https://artemis.ase.in.tum.de/overview/37/exercises/789>

Solution: $0(b|B)(0|1)((_|0|1) * (0|1))?$

P02: Binäre Zahlen

Regular Expression

Problem statement: See Artemis <https://artemis.ase.in.tum.de/overview/37/exercises/789>

Solution: $0(b|B)(0|1)((_|0|1) * (0|1))?$

Or: $0(b|B)(_ (0|1)\{4\}) * (0|1)\{1, 3\}$

P03: Palindrome

In this task you should write a program which checks for a number whether the number is a palindrome. A number is a palindrome if it represents the same value read forwards and backwards.

First the user should be asked for a positive number. Repeat the query until the user actually enters a positive number. The number entered is then to be converted into an array of digits on which the palindrome property is finally checked. Then either print *"palindrome"* or text *"Kein Palindrom"*.

The number is to be read by calling text `readInt()`. It must not be converted into a string at any time. Only use MiniJava methods. In particular, do not use the method `Math.log10(double a)`.

Example Output:

```
1 <Geben Sie eine Zahl n >= 0 ein.  
2 >1221  
3 <Palindrom  
4 <Geben Sie eine Zahl n >= 0 ein.  
5 >123  
6 <Kein Palindrom
```

Code - 1

```
1 public static void main(String[] args) {
2     int n = read("Geben Sie eine Zahl n >= 0 ein.");
3     while (n < 0) {
4         n = read("Geben Sie eine Zahl n >= 0 ein.");
5     }
6
7     // We first count the number of digits of the number n by dividing the number by 10
8     //until it has the value 0. The number of divisions corresponds to the number of digits.
9     int numberOfDigits = 0;
10    int t = n;
11    while (t != 0) {
12        numberOfDigits++;
13        t = t / 10;
14    }
15
16    int[] digits = new int[numberOfDigits];
```


Code - 2

```
1 // We now read the digits into an array. We get a digit as division remainder by 10.
2 //The order in which we place the number in the array does
3 //not matter for the palindrome test.
4
5 int i = 0;
6 while (n != 0) {
7     int digit = n % 10;
8     digits[i] = digit;
9     n = n / 10;
10    i++;
11 }
```

Code - 3

```

1      int notMatching = 0;
2      i = 0;
3      while (i < numberOfDigits / 2) {
4          if (digits[i] != digits[numberOfDigits - i - 1])
5              notMatching++;
6          i++;
7      }
8
9      if (notMatching == 0)
10         write("Palindrom");
11     else
12         write("Kein Palindrom");
13 }

```

P04: Pascalsches Dreieck

The Pascal triangle is built step by step, starting with line 0. To do this, calculate the n th line from the $(n - 1)^{th}$ line as follows:

- The number of elements of line n is $n + 1$.
- The first and last number of each line is always 1.
- The i^{th} element of line n corresponds to the sum of the i^{th} and $(i - 1)^{th}$ elements of line $(n - 1)$.

Write a Java method calls `txtsl int[][] pascalDreieck(int n)` , which calculates the Pascal triangle. The parameter n specifies the number of lines to be calculated; an array containing the Pascal triangle is returned. Also implement the main method, in which a Pascal triangle for a user-defined size n is output. Assume that the input is $n \geq 0$.

P04: Pascalsches Dreieck

Example Output:

```
1 <Gib die Zeilenzahl an:  
2 >5  
3 <n=0      1  
4 <n=1      1  1  
5 <n=2      1  2  1  
6 <n=3      1  3  3  1  
7 <n=4      1  4  6  4  1
```

Code

```
1 // Funktion zur Berechnung der ersten n Zeilen des Pascalschen Dreiecks.
2 public static int[][] pascalDreieck(int n) {
3     int[][] dreieck = new int[n][];
4     for (int m = 0; m < n; m++) {
5         dreieck[m] = new int[m + 1];
6         dreieck[m][0] = 1;
7         dreieck[m][m] = 1;
8         for (int i = 1; i < m; i++)
9             dreieck[m][i] = dreieck[m - 1][i - 1] + dreieck[m - 1][i];
10    }
11    return dreieck;
12 }
```

Code

```
1  public static void main(String[] args) {  
2      int zeilenzahl = read("Gib die Zeilenzahl an:");  
3      int[][] dreieck = pascalDreieck(zeilenzahl);  
4      for (int i = 0; i < zeilenzahl; i++) {  
5          writeConsole("n=" + i);  
6          for (int j = 0; j < dreieck[i].length; j++) {  
7              writeConsole("\t");  
8              writeConsole(dreieck[i][j]);  
9          }  
10         writeLineConsole();  
11     }  
12 }
```

Thank You!