



## Programmieren 2

Vorlesung im Wintersemester 2014/2015

Prof. Dr. habil. Christian Heinlein

### 3. Übungsblatt (28. Oktober 2014)

#### Aufgabe 3: Arrays und Strings in Java

Schreiben Sie eine Java-Klasse mit den folgenden öffentlichen statischen Methoden:

- Die Methode `print` erhält als Parameter ein zweidimensionales (nicht notwendigerweise rechteckiges) Array von ganzen Zahlen und gibt dieses zeilenweise aus, d. h. die Elemente der eindimensionalen Teilarrays werden jeweils durch Leerzeichen getrennt und nach jedem Teilarray wird eine neue Zeile begonnen.
- Die Methode `pascal` erhält als Parameter eine positive ganze Zahl `n` und liefert als Ergebnis die ersten `n` Zeilen des Pascalschen Dreiecks als zweidimensionales „dreieckförmiges“ Array von `int`-Werten.

			1			
		1		1		
	1		2		1	
	1	3		3		1
	1	4	6		4	1
1	5	10		10	5	1
⋮			⋮			⋮

1						
1	1					
1	2	1				
1	3	3	1			
1	4	6	4	1		
1	5	10	10	5	1	
			⋮			

Die Randwerte des Pascalschen Dreiecks sind alle 1, einen inneren Wert erhält man jeweils als Summe der beiden darüberstehenden Werte.

- Die Methode `isPalindrome` überprüft, ob der als Parameter übergebene String ein Palindrom ist, und liefert als Resultat entsprechend `true` oder `false`.  
Ein Palindrom ist eine Zeichenkette, die vorwärts und rückwärts gelesen gleich lautet.  
Nullstrings und leere Strings seien Sonderfälle von Palindromem.
- Die Methode `toLower` erhält als Parameter einen String, ersetzt darin alle Großbuchstaben durch die korrespondierenden Kleinbuchstaben und liefert das Resultat als neuen String.
- Die Hauptmethode `main` enthält für jede der obigen Methoden sinnvolle Testaufrufe und gibt deren Ergebnisse aus.

Außer den Methoden `length` und `charAt` der Klasse `String` und den Methoden `System.out.print` und `System.out.println` sollen keinerlei Java-Bibliotheksmethoden verwendet werden!

## Lösung

```
public class ArrStr {
    // Zweidimensionales Array a zeilenweise ausgeben.
    public static void print (int [] [] a) {
        for (int i = 0; i < a.length; i++) {
            for (int j = 0; j < a[i].length; j++) {
                System.out.print(a[i][j] + " ");
            }
            System.out.println();
        }
    }

    // Pascalsches Dreieck bis Zeile n berechnen.
    public static int [] [] pascal (int n) {
        // Array mit n Zeilen.
        int [] [] p = new int [n] [];

        // Erste Zeile mit einer 1.
        p[0] = new int [] { 1 };

        // Weitere Zeilen.
        for (int i = 1; i < n; i++) {
            // Zeile mit i+1 Elementen erzeugen.
            p[i] = new int [i+1];

            // Randwerte mit 1 initialisieren.
            p[i][0] = p[i][i] = 1;

            // Übrige Werte aus darüberstehenden Werten berechnen.
            for (int j = 1; j < i; j++) {
                p[i][j] = p[i-1][j-1] + p[i-1][j];
            }
        }

        // Dreieck zurückliefern.
        return p;
    }

    // Ist s ein Palindrom?
    public static boolean isPalindrome (String s) {
        // Sonderfall: s ist null.
        if (s == null) return true;

        // Zeichen von s paarweise vergleichen.
        // Resultatwert false, sobald eine Abweichung vorliegt.
        for (int i = 0, j = s.length() - 1; i < j; i++, j--) {
            if (s.charAt(i) != s.charAt(j)) return false;
        }

        // s ist ein Palindrom.
        return true;
    }
}
```

```

// Großbuchstaben in s in Kleinbuchstaben umwandeln.
public static String toLower (String s) {
    // Resultatstring (anfangs leer).
    String r = "";

    // String s zeichenweise durchlaufen.
    for (int i = 0; i < s.length(); i++) {
        // Wenn das aktuelle Zeichen c ein Großbuchstabe ist,
        // wird es durch den zugehörigen Kleinbuchstaben ersetzt.
        char c = s.charAt(i);
        if ('A' <= c && c <= 'Z') c += 'a' - 'A';

        // Zeichen c zum Resultatstring hinzufügen.
        r += c;
    }

    // Resultatstring zurückgeben.
    return r;
}

// Hauptmethode.
public static void main (String [] args) {
    print(pascal(1));
    print(pascal(10));

    System.out.println(isPalindrome("Otto"));           // false
    System.out.println(isPalindrome("otto"));           // true

    System.out.println(toLower("TEST 12 test!"));       // test 12 test!
}
}

```