

Übung 3: Funktionstemplates

Abgabe bis: 22.05.2017

in Verzeichnis: \$HOME/PRG-SPR/ueb03

1. Array

Mit Hilfe von Funktionstemplates ist es möglich generische Funktionen zu erstellen, die für verschiedene Datentypen verwendbar sind. Betrachten Sie den folgenden Programmausschnitt:

```
#include "ArrayTools.h"

...

void starteTests() {
    const size_t SIZE = 10;
    cout << "int-Array testen" << endl;
    int itab[SIZE];
    testeArray(itab, SIZE);

    cout << "\ndouble-Array testen" << endl;
    double dtab[SIZE];
    testeArray(dtab, SIZE);
}

int main() {
    starteTests();
}
```

Ergänzen Sie diesen Programmausschnitt um das Funktionstemplate `testeArray`, das für ein Array mit beliebigen numerischen Datentypen folgendes tun soll:

- zunächst das Array von der Standardeingabe einlesen,
- dann das Array ausgeben
- den maximalen Wert bestimmen und ausgeben
- das Array mit Hilfe des BubbleSort-Algorithmus sortieren (siehe z. B. hier <http://www.sorting-algorithms.com/bubble-sort>)
- und schließlich nochmal ausgeben.

Für diese Funktionalitäten sind in der Headerdatei `ArrayTools.h` jeweils passende Funktionstemplates zu erstellen.

Die Ausgabe des obigen Programms könnte dann z. B. so aussehen

<pre>int-Array testen Bitte 10 Werte eingeben tab[0] = 5 tab[1] = 2 tab[2] = 9 tab[3] = 8 tab[4] = 6 tab[5] = 1 tab[6] = 12 tab[7] = 43 tab[8] = 6 tab[9] = 10 [5, 2, 9, 8, 6, 1, 12, 43, 6, 10] Maximaler Wert: 43 [1, 2, 5, 6, 6, 8, 9, 10, 12, 43]</pre>	<pre>double-Array testen Bitte 10 Werte eingeben tab[0] = 1.5 tab[1] = 5.6 tab[2] = 3.4 tab[3] = 2.4 tab[4] = 99.0 tab[5] = 12.3 tab[6] = 34.5 tab[7] = 56.7 tab[8] = 23.4 tab[9] = 4.5 [1.5, 5.6, 3.4, 2.4, 99, 12.3, 34.5, 56.7, 23.4, 4.5] Maximaler Wert: 99 [1.5, 2.4, 3.4, 4.5, 5.6, 12.3, 23.4, 34.5, 56.7, 99]</pre>
---	--