C++ - - Objektorientierte Programmierung

Standard-Library am Beispiel von Strings

C++ - Standardbibliothek

- Standardisierte Sammlung von häufig vorkommenden Funktionen und Klassen.
- 1998 erste Standardbibliothek bestehend aus den IO-Streams, Strings und der Standard Template Library.
- Erweiterung zum C++ 11-Standard: Reguläre Ausdrücke, Smart Pointer, Hashtabellen, Zufallszahlen und Zeit.

Bücher zur Standardbibliothek

- Rainer Grimm: C++-Standardbibliothek - kurz & gut. O'Reilly. 1. Auflage.

Informationen im Web

- http://www.cplusplus.com/reference/
- http://en.cppreference.com/w/

Leibniz Universität Hannover

Bereiche der Standardbibliothek

<table>
<tr><td>Speicherverwaltung</td><td rowspan="3">Container, Iteratoren, Funktoren und Algorithmen der Standard Template Library</td></tr>
<tr><td>Ein- und Ausgabe-Streams</td></tr>
<tr><td>Lokale Einstellungen</td></tr>
<tr><td>Strings</td><td>Exception Handling</td></tr>
<tr><td>Hilfsklassen</td><td>Numerische Berechnungen</td></tr>
<tr><td></td><td>C-Header-Dateien</td></tr>
</table>

C-Header-Dateien

- Header-Dateien, die mit dem Buchstaben c beginnen.
- Header-Dateien, die aus der Programmiersprache C übernommen wurden.
- Häufig gibt es Alternativen in der Programmiersprache C++.

Strings

- Sammlung von Buchstaben, Zahlen, Satzzeichen, Sonderzeichen und mathematischen Symbolen in einem Feld.
- Verkettung von beliebig vielen Elementen aus dem ASCII-Zeichensatz.
- Array vom Datentyp char[] in der Programmiersprache C.

Leibniz Universität Hannover

ASCII-Zeichentabelle

0 =		18 = ↕		36 = $		54 = 6		72 = H		90 = Z		108 = l	
1 = ☺		19 = ‼		37 = %		55 = 7		73 = I		91 = [		109 = m	
2 = ☻		20 = ¶		38 = &		56 = 8		74 = J		92 = \		110 = n	
3 = ♥		21 = §		39 = ’		57 = 9		75 = K		93 =]		111 = o	
4 = ♦		22 = ▬		40 = (		58 = :		76 = L		94 = ^		112 = p	
5 = ♣		23 = ↨		41 =)		59 = ;		77 = M		95 = _		113 = q	
6 = ♠		24 = ↑		42 = *		60 = <		78 = N		96 = ‘		114 = r	
7 = •		25 = ↓		43 = +		61 = =		79 = O		97 = a		115 = s	
8 = ◘		26 = →		44 = ,		62 = >		80 = P		98 = b		116 = t	
9 = ◇		27 = ←		45 = -		63 = ?		81 = Q		99 = c		117 = u	
10 = ◙		28 = ∟		46 = .		64 = @		82 = R		100 = d		118 = v	
11 = ♂		29 = ↔		47 = /		65 = A		83 = S		101 = e		119 = w	
12 = ♀		30 = ▲		48 = 0		66 = B		84 = T		102 = f		120 = x	
13 = ♪		31 = ▼		49 = 1		67 = C		85 = U		103 = g		121 = y	
14 = ♫		32 =		50 = 2		68 = D		86 = V		104 = h		122 = z	
15 = ☼		33 = !		51 = 3		69 = E		87 = W		105 = i		123 = {	
16 = ►		34 = "		52 = 4		70 = F		88 = X		106 = j		124 = ¦	
17 = ◄		35 = #		53 = 5		71 = G		89 = Y		107 = k		125 = }	

… im Internet

- https://de.wikipedia.org/wiki/American_Standard_Code_for_Information_Interchange
- https://de.wikibooks.org/wiki/C-Programmierung:_ASCII-Tabelle
- http://www.asciitable.com/

... und in der Microsoft Eingabeaufforderung

- Die Microsoft Eingabeaufforderung unterstützt standardmäßig nur ASCII-Zeichencode (siehe http://www.ascii-code.com/).
- Der Befehl *chcp* zeigt die aktuelle genutzte Codepage an.
- Der Befehl *chcp 1252* stellt die aktuelle Codepage auf „West European Latin“ um.

… in C++

```
#include <iostream>
#include <string>

using std::string;

int main() {
    string zitat;

    zitat  = " To be, or not to be--that is the question:
                        Whether 'tis nobler in the mind to suffer.\n";

    return 0;
}
```

Beispiele/cppOOP_005_String...

Header-Datei „string“

- Definition von Strings in der Programmiersprache C++.
- Die Klasse ist ein Container für die Nutzung von Arrays vom Typ `char`.
- Bereitstellung von Methoden zum Durchsuchen, Verketten, Kopieren etc. von Strings.
- Mit Hilfe der Anweisung `using std::string` wird das passende Paket frei gegeben.

Deklaration von String-Variablen

string	zitat	;
string	variable	;

- Definition einer Variablen von der Klasse string.
- Mit Hilfe des Bauplans string wird ein konkretes Objekt zitat erzeugt.
- Der Container wird mit Hilfe des Standard-Konstruktors erzeugt.
- Für das Objekt wird automatisiert genügend Speicher bereitgestellt.

Zuweisung von String-Literalen

```
string zitat;
zitat  = " To be, or not to be--that is the question:
                    Whether 'tis nobler in the mind to suffer.\n";

string autor = "Skapespeare";
```

Beispiele/cppOOP_005_String...

- Literale vom Typ string werden immer durch die Anführungszeichen begrenzt.
- Strings werden mit Hilfe des Gleichheitszeichen einer Variablen vom Typ string zugewiesen.

Nutzung von Konstruktoren

```
string stueck("Hamlet");
string schriftsteller(autor);
string dichter(angabe,0,11);
```

Beispiele/cppOOP_004_String...

- Direkt im Anschluss an den Namen des Strings können die runden Klammern folgen.
- Die runden Klammern sind leer. Die Instanziierung erfolgt mit dem Standard-Konstruktor.
- Die runden Klammern enthalten Parameter, getrennt durch Kommata. Das Objekt wird mit Hilfe des passenden allgemeinen Konstruktors erzeugt.

Parameter „String“

```
string autor = "Skapespeare";
string stueck("Hamlet");
string schriftsteller(autor);
```

Beispiele/cppOOP_005_String...

- String-Literal oder Variablen vom Typ string für die Instanziierung als Parameter genutzt werden.

Nutzung von Teilworten

```
string angabe = "Hamlet Skapespeare";

string dichter(angabe,0,11);
string schreiber(angabe,0,autor.length());
```

Beispiele/cppOOP_005_String...

- Dem Konstruktor wird als erster Parameter ein String übergeben.
- Aus diesen String werden ab der Start-Position (zweiter Parameter) x Zeichen (dritter Parameter) gelesen und in dem entsprechenden String gespeichert.
- Das erste Zeichen in einem String hat den Index 0.

Weitere Möglichkeit

```
string dna(3, 'T');
```

Beispiele/cppOOP_005_String...

- Das Zeichen vom Datentyp char (2. Parameter) wird x mal (1. Parameter) dupliziert.
- Literale vom Datentyp char werden durch Apostrophs begrenzt.

Array von Strings

```
const int maxTeilnehmer = 5;
string teilnehmer[maxTeilnehmer];
```

Beispiele/cppOOP_005_StringArray...

- Ein Array vom Typ string wird genauso deklariert und initialisiert wie Arrays vom Datentyp char, int etc.
- Die Elemente haben einen Index von 0 bis n.

Ausgabe von Strings

```
for(int index = 0; index < maxTeilnehmer; index++)
{
        cout << index << ". Teilnehmer: "
                << teilnehmer[index] << '\n';
}
```

Beispiele/cppOOP_005_StringArray...

Einlesen von Strings

```
for(int index = 0; index < maxTeilnehmer; index++)
    {
        cout << "Bitte geben Sie den Namen
                    des Teilnehmers ein: ";
        cin >> teilnehmer[index];
        cin.clear();
        cin.ignore(10000,'\n');
    }
```

Beispiele/cppOOP_005_StringArray...

Zeilenweises Einlesen

```
#include <iostream>
#include <string>
using std::getline;

int main() {
    const int maxTeilnehmer = 5;
    const int maxZeichen = 100;
    string teilnehmer[maxTeilnehmer];

    for(int index = 0; index < maxTeilnehmer; index++)
    {
        cout << "Bitte geben Sie den
                    Namen des Teilnehmers ein:\n ";
        getline(cin, teilnehmer[index]);
    }
```

Beispiele/cppOOP_005_StringArray...

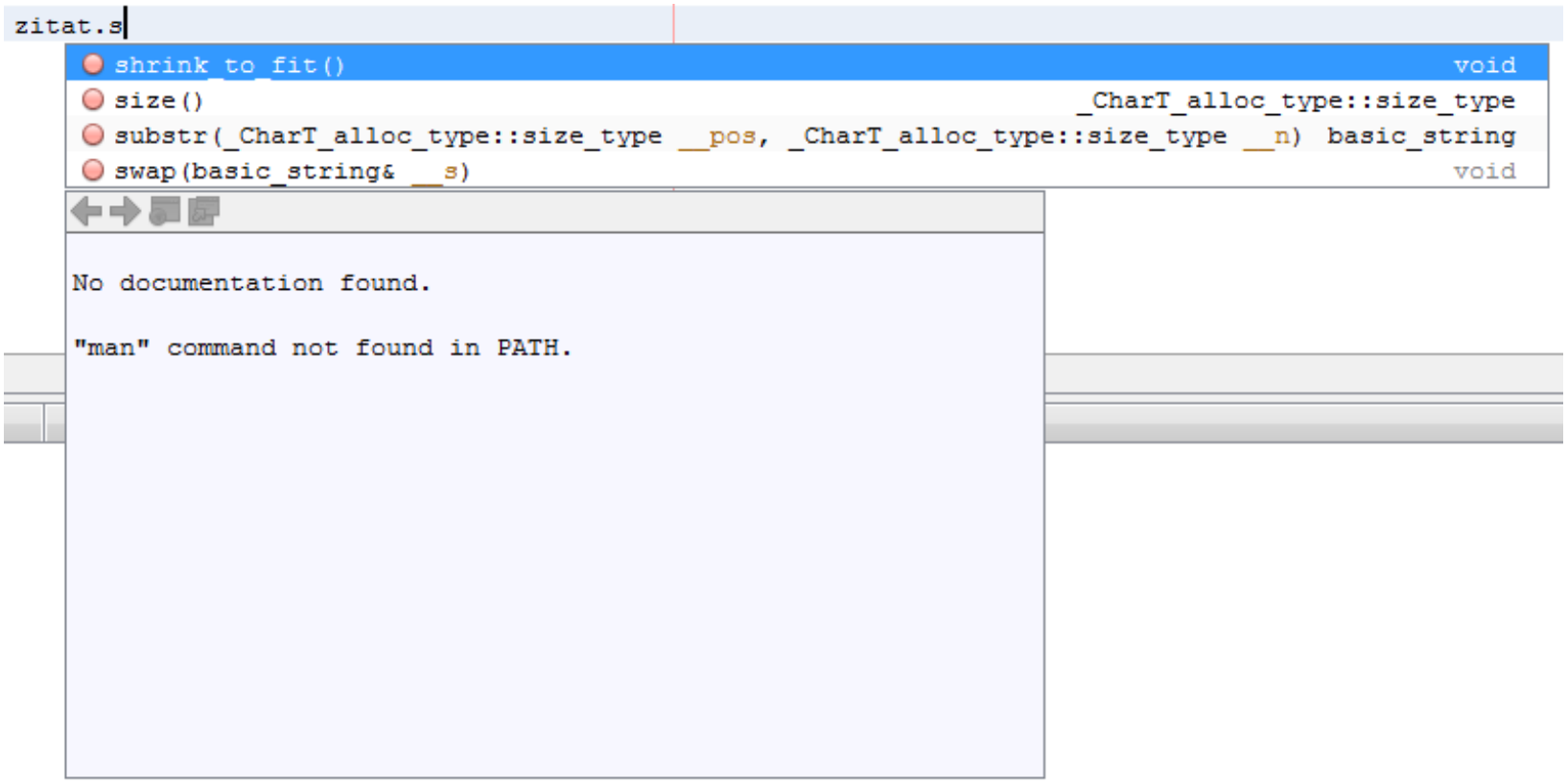
Hinweise

- Als erster Parameter wird der Funktion getline() die Quelle übergeben. Von welchem Medium soll eingelesen werden? In diesem Beispiel wird von der Standardeingabe eingelesen.
- Als zweiter Parameter wird der Funktion getline() der Speicherort der Zeile übergeben. In diesem Beispiel wird die Zeile in einem Array-Element gespeichert.

Methoden der Klasse „String“

- Methoden modifizieren einen String.
- Methoden können wie Funktionen mit Hilfe der Parameterliste und / oder der Anzahl der Parameter überladen werden.

Informationen zu einer Methode in Netbeans



… und im Web

- http://www.cplusplus.com/reference/
- http://openbook.rheinwerk-verlag.de/c_von_a_bis_z/030_c_anhang_b_001.htm

Länge eines Strings

```
string buchstaben = "abcdefg";

cout << "Anzahl der Zeichen: " << buchstaben.size();
```

Beispiele/cppOOP_005_StringMethode...

- Die Methode size() gibt die Anzahl von Zeichen in einem String zurück.
- Methoden für Objekte vom Datentyp string sind in der Klasse <string> definiert.

Zeichen in einem String

```
cout << "\nErstes Zeichen: " << buchstaben[0];
cout << "\nErstes Zeichen: " << buchstaben.at(0);
```

Beispiele/cppOOP_005_StringMethode...

- Eine Variable vom Datentyp string entspricht einem Array von char.
- Mit Hilfe des Indizes kann ein Zeichen in einem String ausgegeben oder in einer Variablen gespeichert werden.
- Das erste Zeichen in einem String hat den Index 0. Das letzte Zeichen hat den Index (Anzahl – 1).

Methode nutzen

```
cout << "\nErstes Zeichen: " << buchstaben[0];
cout << "\nErstes Zeichen: " << buchstaben.at(0);
```

Beispiele/cppOOP_005_StringMethode...

- Die Methode .at() gibt ein einzelnes Zeichen zurück.
- In den runden Klammern wird der Methode die Position des Zeichens in dem String übergeben.
- Das erste Zeichen hat die Position 0.
- Vorteil: Die Methode wirft ein Fehler, wenn die Unter- oder Obergrenze des Strings überschritten wird.

Vergleich von Strings

```
string kleinbuchstaben = "abcdefg";
string grossbuchstaben = "ABCDEFG";
bool ergebnis;

ergebnis = (kleinbuchstaben == grossbuchstaben);
```

Beispiele/cppOOP_005_StringMethode...

- Die Zeichen werden aufgrund ihrer ASCII-Codierung verglichen.
- Der Rückgabewert ist vom Datentyp bool.

Umwandlung von Groß- in Kleinbuchstaben

```
for(int index = 0; index < grossbuchstaben.size(); index++)
{
   kleinZeichen += std::tolower(grossbuchstaben[index]);
}
```

Beispiele/cppOOP_005_StringMethode...

- Die Funktion tolower() wandelt einzelne Zeichen in Kleinbuchstaben um.
- Zeichen, die keine Buchstaben sind, werden ignoriert.
- Die Funktion ist in der Bibliothek <locale> definiert.

Umwandlung von Klein- in Großbuchstaben

```
for(int index = 0; index < kleinbuchstaben.size(); index++)
{
   grossZeichen += (std::toupper(kleinbuchstaben[index]));
}
```

Beispiele/cppOOP_005_StringMethode...

- Die Funktion toupper() wandelt einzelne Zeichen in Großbuchstaben um.
- Zeichen, die keine Buchstaben sind, werden ignoriert.
- Die Funktion ist in der Bibliothek <locale> definiert.

Methode nutzen

```
string kleinbuchstaben = "abcdefg";
string grossbuchstaben = "ABCDEFG";
int result;

result = (kleinbuchstaben.compare(grossbuchstaben));
cout << "\nIst gleich? " << result << '\n';
```

Beispiele/cppOOP_005_StringMethode...

- Die Zeichen werden aufgrund ihrer ASCII-Codierung verglichen.
- Der Rückgabewert ist vom Datentyp int.

Rückgabe-Parameter der Methode

- = 0. Der String ist gleich.
- < 0. An der Position ist der Buchstabe kleiner als der andere.
- > 0. An der Position ist der Buchstabe größer als der andere.
- Hinweis: In der ASCII-Tabelle liegen die Großbuchstaben vor den Kleinbuchstaben. Kleinbuchstaben werden mit einer größeren Zahl als Großbuchstaben codiert.

Vergleich von Teilstrings

```
string kleinbuchstaben = "abcdefg";
string grossbuchstaben = "ABCDEFG";
int result;

result = (kleinbuchstaben.compare(2, 1, grossbuchstaben, 2, 1) );
cout << "\nIst gleich? " << result << '\n';
```

Beispiele/cppOOP_005_StringMethode...

Erläuterung

- In diesem Beispiel wird das dritte Zeichen in zwei Strings verglichen.
- Die ersten zwei Parameter beziehen sich auf die Variable, die die Methode aufruft. In diesem Beispiel: kleinbuchstaben.
- Die letzten zwei Parameter beziehen sich auf die Variable, mit der der String kleinbuchstaben verglichen werden soll. In diesem Beispiel grossbuchstaben (3. Parameter).
- In diesem Beispiel wird ein Zeichen (2. Parameter) mit einem anderen (5. Parameter) verglichen.
- Der Vergleich beginnt in dem String kleinbuchstaben und in dem String großbuchstaben an der zweiten Position (1. Parameter und 4. Parameter).

Verknüpfung von Strings

```
string kleinbuchstaben = "abcdefg";
string grossbuchstaben = "ABCDEFG";
string buchstaben;

buchstaben = kleinbuchstaben + grossbuchstaben;;
```

Beispiele/_cppOOP_005_StringMethode...

- Mit Hilfe des Pluszeichen können einzelne Zeichen oder Strings zu einem neuen String verknüpft werden.

Zeichen an das Ende eines Strings anhängen

```
grossZeichen += (std::toupper(kleinbuchstaben[index]));
```

Beispiele/cppOOP_005_StringMethode...

- Mit Hilfe des zusammengesetzten Operators =+ können einzelne Zeichen oder Strings an das Ende eines anderen Strings angehängt werden.

Erläuterung

- Zusammengesetzte Operatoren führen zuerst die Rechenoperation aus. In diesem Beispiel wird grossbuchstaben mit einem Zeichen aus dem String kleinbuchstaben verknüpft.
- Das Ergebnis der Operation wird in der Variablen links vom zusammengesetzten Operator gespeichert.

Mit Hilfe einer Methode

```
buchstaben.append(kleinbuchstaben);
```

Beispiele/cppOOP_005_StringMethode...

- Der Methode append() wird der String übergeben, der an das Ende eines Strings angehängt werden soll.
- Der übergebene Parameter wird an den String links vom Punkt angehängt. Die Methode verändert das dazugehörige Objekt.

Teilstrings anhängen

```
buchstaben.append(grossbuchstaben, 0, 3);
```

Beispiele/cppOOP_005_StringMethode...

- Der Methode append() wird als erster Parameter, der String übergeben. Aus diesen String soll ein Teilstring an das, zu der Methode gehörende Objekt angehängt werden.
- Als zweite Parameter wird die Start-Position übergeben. Hier beginnt der Teilstring an der ersten Position des Strings grossbuchstaben.
- Als dritter Parameter wird die Länge des Teilstrings übergeben. In diesem Beispiel hat der Teilstring eine Länge von 3 Zeichen.

Einfügung von Strings

```
buchstaben.insert(3, grossbuchstaben);
```

Beispiele/cppOOP_005_StringMethode...

- An der Position x (erster Parameter) wird ein String (zweiter Parameter) eingefügt.
- Der Text ab der Position x + 1 wird automatisch verschoben.

Ersetzung von Strings

```
buchstaben.replace(2,strlen("xyz"),"XYz", strlen("xyz"));
```

Beispiele/cppOOP_005_StringMethode...

- An der Position x (erster Parameter) werden x Zeichen (zweiter Parameter) ersetzt.
- Als dritter Parameter wird der Ersetzungsstring an die Methode übergeben.
- Der vierte Parameter gibt die Länge des Ersetzungsstrings an.
- In diesem Beispiel werden ab der zweiten Position drei Zeichen durch die ersten drei Zeichen aus dem String XYz ersetzt. Der String hat eine Länge von drei Zeichen.

Löschen von Zeichen aus einem String

```
buchstaben.erase(2, 2);
```

Beispiele/cppOOP_005_StringMethode...

- Ab der Position x (erster Parameter) werden x Zeichen (zweiter Parameter) gelöscht.

Löschen des Strings

```
buchstaben.clear();
```

Beispiele/cppOOP_005_StringMethode...

- Der String wird vollständig gelöscht.

Löschen des Strings

```
if (!(buchstaben.empty()))
{
        buchstaben.clear();
}
```

Beispiele/cppOOP_005_StringMethode...

- Das Ausrufezeichen negiert einen boolschen Wert.
- Mit Hilfe der Methode .empty() wird abgefragt, ob der String leer ist.
- Wenn der String nicht leer ist, lösche diesen.

Suchen in Strings

```
string zitat = " To be, or not to be--that is the question:
                Whether 'tis nobler in the mind to suffer.\n";
int zaehler = -1;
int pos = zitat.size() + 1;

do
{
   pos--;
   zaehler++;
   pos =  zitat.rfind("b", pos);
}while(pos != string::npos);
```

Beispiele/cppOOP_005_StringMethodeFind...

Erläuterung

- Die Methode rfind(string, pos) beginnt am Ende (right side) des Strings mit der Suche.
- Die Methode find(string, pos) beginnt am Anfang des Strings mit der Suche.
- Beiden Methoden wird als erster Parameter der String übergeben, nach dem gesucht werden soll. In diesem Beispiel wird nach dem Buchstaben b gesucht.
- Als zweiter Parameter wird der Beginn der Suche übergeben. Wo soll in dem Objekt mit der Suche begonnen werden? Der zweite Parameter ist optional.

Leibniz Universität Hannover

Suchen in Strings

```
string waehrung = "£123,456.78|£456.78|£25,456.78|£101,456.78";
std::size_t gefundenAn = waehrung.find_first_of(",.");

 while (gefundenAn !=std::string::npos){
    switch(waehrung[gefundenAn]){
                case '.':
                waehrung[gefundenAn] = ',';
                break;

                case ',':
                waehrung[gefundenAn] = '.';
          break;
    }

    gefundenAn= waehrung.find_first_of(",.",gefundenAn+1);
 }
```

Beispiele/cppOOP_005_StringMethodeFind...

Erläuterung

- Die Methode find_first_of(string) beginnt am Anfang des Strings mit der Suche.
- Die Methode find_last_of(string) beginnt am Ende des Strings mit der Suche.
- Der zu übergebende Suchstring enthält alle Zeichen, nach denen in dem dazugehörigen Objekt gesucht werden soll.

Typ „size_t“

```
std::size_t gefundenAn = waehrung.find_first_of(",.");
```

- size_t ist ein vorzeichenloser Integer-Typ.
- Rückgabewert von sizeof(datentyp).
- Der Typ ist in der Bibliothek <cstring> und vielen anderen Bibliotheken definiert.
- Speicherung eines Indizes.
- Angabe der Größe eines Objekts in Bytes.
- Implementierungsabhängig.

Leibniz Universität Hannover

Teilstrings

Beispiele/cppOOP_005_StringSubstring...

```
aktuellePos = waehrung.find("|");

while(aktuellePos != string::npos)
{
  strTemp =  waehrung.substr(vorherigePos,
                                    aktuellePos - vorherigePos);
  vorherigePos = aktuellePos + 1;
  aktuellePos = waehrung.find("|", vorherigePos + 1);

}
```

Erläuterung

- Die Methode substring(start, anzahl) gibt einen String, beginnend an der Start-Position (erster Parameter) mit x Zeichen (zweiter Parameter) zurück.
- Der zweite Parameter ist optional. Wenn keine Angaben zur Anzahl gemacht werden, wird der String ab der Start-Position bis zum letzten Zeichen zurückgegeben.

String-Streams

- Speicherung von Daten in einem String.
- Keine Verbindung zu einem Ein- und Ausgabe-Stream.
- Konvertierung von unterschiedlichen Datenformaten untereinander.
- Konvertierung von Strings in numerische Werte und umgekehrt.

Beispiel

Beispiele/cppOOP_005_StringStream

```
#include <sstream>
#include <iomanip>
using std::string;
using std::stringstream;

int main() {
    stringstream ausgabeText;
    string strTemperatur;
    stringstream streamTemperatur;
    double dblTemperatur;

    ausgabeText.str("Bitte geben Sie die durchschnittliche Temperatur ein:\n ");
    std::cout << ausgabeText.str();
    std::cin >> strTemperatur;

    streamTemperatur.str(strTemperatur);
    streamTemperatur >> std::setprecision(9) >> dblTemperatur;
```

Header-Datei <sstream>

- String-Streams werden in der Bibliothek <sstream> definiert.
- Mit Hilfe der Anweisung using std::stringstream werden String-Streams für die Ein- und Ausgabe von Daten vom Typ char zur Nutzung freigegeben.

Deklaration von String-Variablen

stringstream	streamTemperatur	;
stringstream	variable	;

- Definition einer Variablen von der Klasse stringstream.
- Mit Hilfe des Bauplans stringstream wird ein konkretes Objekt streamTemperatur erzeugt.
- Mit Hilfe der Deklaration wird eine „Leitung“ zu einem x-beliebigen String-Objekt gelegt.

„Leitung“ legen

```
stringstream ausgabeText;
string strTemperatur;
stringstream streamTemperatur;

ausgabeText.str("text\n");
streamTemperatur.str(strTemperatur);
```

Beispiele/cppOOP_005_StringStream...

- Mit Hilfe der Methode .str(string) wird ein String in ein String-Stream kopiert.

Konvertierung in ein String

```
cout << ausgabeText.str();
```

Beispiele/cppOOP_005_StringStream...

- Mit Hilfe der Methode .str() wird aus dem Inhalt im Stream ein String erzeugt.
- Hinweis: Die Methode gibt ein temporäres String-Objekt zurück. Es wird der aktuelle Zustand des String-Streams zurück gegeben.

Konvertierung in eine Gleitkommazahl

```
stringstream streamTemperatur;
double dblTemperatur;

streamTemperatur >> std::setprecision(9) >> dblTemperatur;
```

Beispiele/cppOOP_005_StringStream...

- Durch die Umleitungsoperatoren wird der Stream auf eine Variable vom Datentyp double umgeleitet.
- Durch die Umleitung wird der String-Stream in den entsprechenden Datentyp umgeleitet.

Gleitkommazahl → Stringstream

```
stringstream ausgabeText;
ausgabeText << maxTemperatur.temperatur << '.';
```

Beispiele/cppOOP_005_StringStream_ConvertFrom...

- Die Umleitungsoperatoren zeigen an, wo ein Wert hinfließt.
- In diesem Beispiel wird der Wert einer Variablen vom Datentyp double auf einen String-Stream umgeleitet.
- Die Konvertierung erfolgt automatisch.