

Die Eingabe von Zeichen `char *s` kann in C++ mit `cin>>s` erfolgen. Dabei werden führende "Whitespace"-Zeiche (Blank, Wagenrücklauf, Zeilenvorschub, Tab, ...) **überlesen**, das Einlesen **endet** vor dem **ersten** Whitespace-Zeichen bzw. mit dem **Ende der Datei (EOF)**, automatisch wird `s` mit `'\0'` terminiert. Der Speicherbereich für `char *s` muß vor dem Einlesen vereinbart worden sein:

```
#define N 10
char s[N];
```

Damit die Gefahr des **Überschreitens** der Speicherbereichsgrenze beim Einlesen gebannt wird, bietet sich der Manipulator `setw()` in folgendem Kontext an:

```
cin >> setw(sizeof s) >> s;
```

Die Headerfiles `iostream.h` und `iomanip.h` werden benötigt.

Neben der formatierten Eingabe über `cin >> s` existieren **unformatierte** Eingabemöglichkeiten für die Klasse `istream` :

```
int get(); // wie getchar() aus C
```

Liest Zeichen und liefert dessen numerischen Wert zurück. Falls kein Zeichen mehr vorhanden, dann wird **EOF** geliefert und das Stream-Objekt (z.B. `cin`) in den Zustand `eof()` versetzt.

```
istream& get( char& c);
```

Liest nächstes Zeichen nach `c` und gibt das **Stream-Objekt** über den Funktionsnamen zwecks weiterer Auswertungsmöglichkeit zurück. Falls kein Zeichen mehr vorhanden, dann wird das Stream-Objekt (z.B. `cin`) in den Zustand `eof()` versetzt.

```
istream& get(char *s, int anz, char ende='\n');
```

Liest bis zu **anz-1** Zeichen nach `s`, wobei spätestens mit dem Zeichen **ende** das Lesen abgebrochen wird. Das Zeichen `'\0'` wird automatisch am Ende angefügt. Das Zeichen **ende** wird nicht mit übernommen und verbleibt als erstes ungelesenes Zeichen im Eingabestrom. Diese Methode sollte niemals zweimal hintereinander aufgerufen werden, ohne zwischendurch das Zeichen **ende** zu lesen !

```
istream& getline(char *s, int anz, char ende='\n');
```

Arbeitet wie vorherige `get`-Methode, außer daß das Zeichen **ende** gelesen aber **nicht in `s` gespeichert** wird.

Die beiden letztgenannten Methoden lesen Folgen von maximal **anz** Zeichen, wobei **get** bzw. **getline** den Einlesevorgang abbrechen, wenn

1. das **Ende** des Eingabestroms erreicht ist, oder
2. das Begrenzerzeichen **ende** auftritt, oder
3. bereits **anz - 1** Zeichen gespeichert sind.

```
istream& putback(char c);
```

Stellt ein Zeichen `c` in den Eingabestrom zurück, damit es erneut gelesen werden kann.

Im Gegensatz zu `cin >> s` behandeln `get()` und `getline()` die Trennzeichen/Whitespacezeichen wie andere Zeichen.

Beispiele:

```
#include <iostream>    // LES2_6.CPP
#include <iomanip>
#define N 10
using namespace std;

void f();           // Prototyp

void main()
{ f();             // Achtung bei Aufruf    f;
  f();             // Repeated Call
}

void f()
{ char buf[N];
  do { cin.get(buf,sizeof(buf)); //cin.get(buf,sizeof(buf),'\n');
      if(strlen(buf)) cout<<"buf="<<buf<<endl;
    }
  while(strlen(buf)==sizeof(buf)-1);
  cin.get(); //'\n' aus Eingabepuffer lesen
}
```

```
#include <iostream>    // LES2_7.CPP
#include <iomanip>
#define N 10

using namespace std;

void f();           // Prototyp

void main()
{ f();             // Achtung bei Aufruf    f;
  f();             // Repeated call
}

void f()
{ char buf[N], c;
  do { cin>>setw(sizeof buf)>>buf; // sicher, jedoch Ende beim
      cout<<"buf="<<buf<<endl;    // 1. Whitespace-Zeichen !!
      if(cin.get(c) && c=='\n') return;
      cin.putback(c);
    }
  while(strlen(buf)==sizeof(buf)-1);
}
/*
0123456789abcdef
buf=012345678
buf=9abcdef
*/
```

```
#include <iostream>    // LES2_8.CPP
#include <iomanip>
#define N 10

using namespace std;

void f();           // Prototyp
void main()
{ f();             // Achtung bei Aufruf    f;
  f();             // Repeated call
}

void f()           // Universelle Funktion zum sicheren Einlesen
{ char buf[N];
  do { cin.getline(buf,sizeof(buf)); //cin.getline(buf,sizeof(buf),'\n');
      if(strlen(buf)) cout<<"buf="<<buf<<endl;
    }
  while(strlen(buf)==sizeof(buf)-1);
}
/*
0123456789abcdefghijklmnopq
buf=012345678
buf=9abcdefgh
buf=ijklmnopq
*/
```

```
// Les3_1.cpp
#include <iostream>
#include <iomanip>
#define N 10

using namespace std;

void main()
{
  char Eingabepuffer[N];
  cin.getline(Eingabepuffer,N,'\n');
  while(strlen(Eingabepuffer))
  {
    char *Zstr=new char[strlen(Eingabepuffer)+1];
    strcpy(Zstr,Eingabepuffer);
    cout<<"Zstr:"<<Zstr<<endl;
    delete [] Zstr; // bei Vektoren: []
    Zstr=0;         // Zeigerwert 0
    delete [] Zstr; // wiederholt bei 0 zulaessig
    cin.getline(Eingabepuffer,N,'\n');
  }
  cin.get();
}
/*
abcdefghijklmnopqrstuvwxyz
Zstr:abcdefghi
Zstr:ijklmnopqr
Zstr:stuvwxyz
*/
```