

## §4 DATEI- UND STRINGSTRÖME

*Leitideen: Der Lesen und Schreiben von Dateien erfolgt in C++ einheitlich über Text- oder Binärströme, um von der speziellen Implementierung der Dateien im jeweiligen Betriebssystem unabhängig zu sein.*

*Dateiströme werden durch Dateistromklassenobjekte bereitgestellt, die Verknüpfung mit Dateien und Angabe des Zugriffsmodus erfolgt bei der Definition oder durch Aufruf der Komp.fkt. open.*

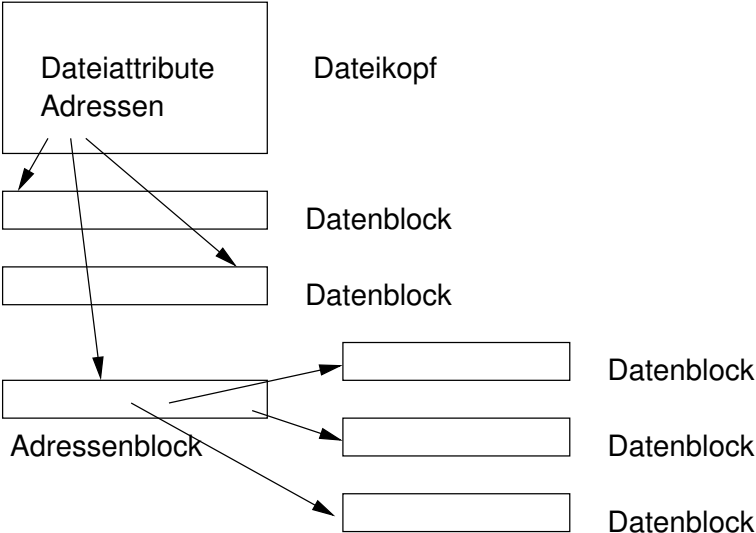
*Stringstromklassen ermögl. das Lesen/Beschreiben von Strings mit den üblichen Ein/Ausg.operatoren.*

*Ströme sind in der Regel gepuffert, d.h. Daten werden gruppenweise (Zeilen/Blöcke) übertragen.*

- Ein/Ausgabe auf Dateien I-III
- Ein/Ausgabe auf Strings
- Pufferung

# Ein/Ausgabe auf Dateien I

## Datei aus Betriebssystemansicht (Unix)



# Ein/Ausgabe auf Dateien II

## Datei aus Benutzersicht (Strom)



Folge von Bytes/Zeichen

## Dateitypen (Stromtypen) in C++ [basierend auf `char`]

- ▶ Binärdateien (für Daten)
- ▶ ASCII-Textdateien (Zeilenstruktur)

<i>DOS</i>	<i>Linux/Unix</i>
Zeile\r\n	Zeile\n
Zeile\r\n	Zeile\n
:	:
Ctrl-Z	

Einheitliche Behandlung von ASCII-Textdateien durch Öffnen als Textstrom: \n Zeilenende, kein Dateiendezeichen

## Ein/Ausgabe auf Dateien III

### Stromklassen für Dateien [basierend auf `char`]

- ▶ Headerdatei: `fstream`
- ▶ Stromklassen: `ifstream`, `ofstream`, `fstream`
- ▶ Stromklassenobjekt vereinbaren mit Dateiname und Zugriffsmodus als Konstruktorargumente.  
Dateiname muss in C++98 eine C-Zeichenkette sein:  
*Bsp.:* `ifstream ein("dat.bin", ios::binary)`  
*Daher:* In C++98 an Stelle des Strings *fname* die C-Zeichenkette `fname.c_str()` verwenden
- ▶ Fehlerbehandlung durchführen, falls Datei nicht zugreifbar
- ▶ Öffnen und Schließen von Dateien auch mit Komponentenfunktionen (`open`, `close`) möglich

*Bsp.:* → *Inf.bl.9, S.2*