Materialsammlung - generische Programmierung

Prof. Dr. Hans-Jürgen Buhl

Winersemester 2010/2011

Fachgruppe Mathematik und Informatik
FB C
Bergische Universität Wuppertal

Praktische Informatik
PIBUW - WS10/11
Oktober 2010
1. Auflage, 2010

Vorbemerkungen:

Einordnung in Programmierparadigmen
imperativ, objektbasiert, objektorientiert, funktional, generisch, ...

Die Entwicklung der Aussagekraft der formalen generischen Parameter

- von einfallslosen Parameternamen wie `class T1, class T2, ...`
  vergleiche http://www.cplusplus.com/doc/tutorial/templates/
- über semantisch inhaltsvolle Parameternamen wie `typename InputIterator1, typename InputIterator2, typename NumericT, ...`
  vergleiche http://www.iue.tuwien.ac.at/phd/heinzl/node32.html#SECTION01022300000000000000
- hin zur Nennung der Requirements an die zur Instantiierung benutzbaren aktuellen Parameter wie `T shall meet the requirements of CopyConstructible and CopyAssignable types`
  (Seite 970 von http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2010/n3126.pdf)
  mit der Erläuterung:

Table 33 — `DefaultConstructible` requirements [**defaultconstructible**]

Expression	Post-condition
`T t;`	object `t` is default-initialized
`T u{};`	object `u` is value-initialized
`T()` `T{}`	a temporary object of type `T` is value-initialized

Table 34 — `MoveConstructible` requirements [**moveconstructible**]

Expression	Post-condition
`T u(rv);`	`u` is equivalent to the value of `rv` before the construction
`T(rv)`	`T(rv)` is equivalent to the value of `rv` before the construction
[*Note:* `rv` remains a valid object. Its state is unspecified — *end note*]	

Table 35 — `CopyConstructible` requirements (in addition to MoveConstructible) [**copyconstructible**]

Expression	Post-condition
`T u(v);`	the value of `v` is unchanged and is equivalent to `u`
`T(v)`	the value of `v` is unchanged and is equivalent to `T(v)`

Table 36 — `MoveAssignable` requirements [**moveassignable**]

Expression	Return type	Return value	Post-condition
`t = rv`	`T&`	`t`	`t` is equivalent to the value of `rv` before the assigment
[*Note:* `rv` remains a valid object. Its state is unspecified. — *end note*]			

(Seite 483 des Drafts)

C++0x frühestens 2010 und ohne „Concepts“
http://www.heise.de/developer/meldung/C-0x-fruehestens-2010-und-ohne-Concepts-7559.html

TR1 - ein „Zwischenstandard“ für C++-Bibliotheken
TR1

Was hätten Concepts gebracht?
http://en.wikipedia.org/wiki/Concepts_(C%2B%2B)

C1x – der neue C-Draft
http://en.wikipedia.org/wiki/C1X

Der C++-Draft
http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2010/n3126.pdf

Ziele des Draft-Desings
http://www.artima.com/cppsource/cpp0x.html

TR2 call for proposals
http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2005/n1810.html

1 Generische Programmierung

1.1 Was ist generische Programmierung?

http://pdfcast.org/pdf/the-java-generic-programming-system
http://foldoc.org/generic+programming
http://www.boost.org/community/generic_programming.html
http://en.wikipedia.org/wiki/Generic_programming

1.2 Beispiel einer generische Funktion mit einem generischen Parameter

```
#include <iostream>
template <typename T>
/*
 * Requirements: T muss einen Kopierkonstruktor haben,
 *               T muss einen Zuweisungsoperator zu T haben.
 */
void swap(T& a, T& b)
{
   T  old_a(a);

   a = b;
   b = old_a;
}
int main(){
   int k(1);
   int l(5);
   std::cout << k << " " << l << std::endl;
   swap(k,l);
   std::cout << k << " " << l << std::endl;
   double d1(3.1415);
   double d2(15.1055);
   std::cout << d1 << " " << d2 << std::endl;
   swap(d1, d2);
   std::cout << d1 << " " << d2 << std::endl;
}
```

1.3 Benutzung von `std::swap()`

```
#include <iostream>
using std::cout;

int main(){

  int i = 5;
  int j = 6;
  cout << "i = " << i << "; j = " << j << std::endl;
  std::swap(i,j);
  cout << "i = " << i << "; j = " << j<< std::endl;

  int a[2] = {2, 3};
  int b[2] = {12, 13};
  cout << a[0] << " " << a[1] << std::endl;
  cout << b[0] << " " << b[1] << std::endl;
  std::swap(a, b);
  cout << a[0] << " " << a[1] << std::endl;
  cout << b[0] << " " << b[1] << std::endl;
}
```

unter Benutzung der „gerneral utilities library“ (Kapitel 20 des C++-Drafts):

20.3.2 swap **[utility.swap]**

```
template<class T> void swap(T& a, T& b);
```

1 *Requires:* Type `T` shall be `MoveConstructible` (Table 34) and `MoveAssignable` (Table 36).

2 *Effects:* Exchanges values stored in two locations.

```
template<class T, size_t N>
  void swap(T (&a)[N], T (&b)[N]);
```

3 *Requires:* `a[i]` shall be swappable with (20.2.2) `b[i]` for all `i` in the range `[0,N)`.

4 *Effects:* `swap_ranges(a, a + N, b)`

Zu den Requirements an den generischen Parameter siehe Seite 1. Die `swappable.requirements` findet man in Aschnitt 20.2.2 des Drafts. Fassen Sie sie in eigenen Worten zusammen.

1.4 Einsatzgebiete und Beispielrepositorien für generische Konstrukte: die STL, ...

http://www.sgi.com/tech/stl/
generische Java-Datentypen
Die Boost C++-Bibliotheken

1.5 Objekt-Dateien und shared Bibliotheken

http://en.wikipedia.org/wiki/Executable_and_Linkable_Format
http://www.ibm.com/developerworks/aix/library/au-unixtools/index.html
http://cpp.comsci.us/process/build.html

ldd und was es zeigt:

```
> ls
swap1.cpp
> cat swap1.cpp
#include <iostream>
template <typename T>
/*
 * Requirements: T muss einen Kopierkonstruktor haben,
 *               T muss einen Zuweisungsoperator zu T haben.
 */
void swap(T& a, T& b)
{
   T  old_a(a);

   a = b;
   b = old_a;
}
int main(){
...
   int k(1);
   int l(5);
   swap(k,l);
...
}

> make swap1
g++  -g -I. -I/home/username/include   swap1.cpp   -o swap1

> ldd ./swap1
        linux-vdso.so.1 =>  (0x00007fffe1b0d000)
        libstdc++.so.6 => /usr/lib64/libstdc++.so.6 (0x00007fa6005e9000)
        libm.so.6 => /lib64/libm.so.6 (0x00007fa600392000)
        libgcc_s.so.1 => /lib64/libgcc_s.so.1 (0x00007fa60017c000)
        libc.so.6 => /lib64/libc.so.6 (0x00007fa5ffe1c000)
        /lib64/ld-linux-x86-64.so.2 (0x00007fa6008f3000)
```

Beim Programmlauf werden nacheinander die „shared object“-Bibliotheken geöffnet und nötige Teile in das auszuführende Binary eingebunden:

```
> strace ./swap1
execve("./swap1", ["./swap1"], [/* 66 vars */]) = 0
brk(0)                                  = 0x602000
mmap(NULL, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7f17e3146000
access("/etc/ld.so.preload", R_OK)      = -1 ENOENT (No such file or directory)
open("/etc/ld.so.cache", O_RDONLY)      = 3
fstat(3, {st_mode=S_IFREG|0644, st_size=335735, ...}) = 0
mmap(NULL, 335735, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7f17e30f4000
close(3)                                = 0
open("/usr/lib64/libstdc++.so.6", O_RDONLY) = 3
read(3, "\177ELF\2\1\1\0\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\260\305\5\0\0\0\0\0"..., 832) = 832
fstat(3, {st_mode=S_IFREG|0755, st_size=1003544, ...}) = 0
mmap(NULL, 3182936, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7f17e2c1f000
fadvise64(3, 0, 3182936, POSIX_FADV_WILLNEED) = 0
mprotect(0x7f17e2d0b000, 2093056, PROT_NONE) = 0
mmap(0x7f17e2f0a000, 40960, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0xeb000) = 0x7f17e2f0a000
mmap(0x7f17e2f14000, 82264, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7f17e2f14000
close(3)                                = 0
open("/lib64/libm.so.6", O_RDONLY)      = 3
read(3, "\177ELF\2\1\1\0\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0‘>\0\0\0\0\0\0"..., 832) = 832
fstat(3, {st_mode=S_IFREG|0755, st_size=391908, ...}) = 0
mmap(NULL, 2449592, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7f17e29c8000
fadvise64(3, 0, 2449592, POSIX_FADV_WILLNEED) = 0
mprotect(0x7f17e2a1e000, 2093056, PROT_NONE) = 0
mmap(0x7f17e2c1d000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x55000) = 0x7f17e2c1d000
close(3)                                = 0
open("/lib64/libgcc_s.so.1", O_RDONLY)  = 3
read(3, "\177ELF\2\1\1\0\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0‘-\0\0\0\0\0\0"..., 832) = 832
fstat(3, {st_mode=S_IFREG|0755, st_size=88544, ...}) = 0
mmap(NULL, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7f17e30f3000
mmap(NULL, 2184184, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7f17e27b2000
fadvise64(3, 0, 2184184, POSIX_FADV_WILLNEED) = 0
mprotect(0x7f17e27c7000, 2093056, PROT_NONE) = 0
mmap(0x7f17e29c6000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x14000) = 0x7f17e29c6000
close(3)                                = 0
open("/lib64/libc.so.6", O_RDONLY)      = 3
read(3, "\177ELF\2\1\1\0\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\220\354\1\0\0\0\0\0"..., 832) = 832
fstat(3, {st_mode=S_IFREG|0755, st_size=1670469, ...}) = 0
mmap(NULL, 3537800, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7f17e2452000
fadvise64(3, 0, 3537800, POSIX_FADV_WILLNEED) = 0
mprotect(0x7f17e25a8000, 2097152, PROT_NONE) = 0
mmap(0x7f17e27a8000, 20480, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x156000) = 0x7f17e27a8000
mmap(0x7f17e27ad000, 19336, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7f17e27ad000
close(3)                                = 0
mmap(NULL, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7f17e30f2000
mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7f17e30f0000
arch_prctl(ARCH_SET_FS, 0x7f17e30f0720) = 0
mprotect(0x7f17e27a8000, 16384, PROT_READ) = 0
mprotect(0x7f17e29c6000, 4096, PROT_READ) = 0
mprotect(0x7f17e2c1d000, 4096, PROT_READ) = 0
mprotect(0x7f17e2f0a000, 32768, PROT_READ) = 0
mprotect(0x600000, 4096, PROT_READ)     = 0
mprotect(0x7f17e3147000, 4096, PROT_READ) = 0
munmap(0x7f17e30f4000, 335735)          = 0
fstat(1, {st_mode=S_IFCHR|0620, st_rdev=makedev(136, 2), ...}) = 0
mmap(NULL, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7f17e3145000
write(1, "1 5\n", 41 5
)                       = 4
write(1, "5 1\n", 45 1
)                       = 4
write(1, "3.1415 15.1055\n", 153.1415 15.1055
)        = 15
write(1, "15.1055 3.1415\n", 1515.1055 3.1415
)        = 15
exit_group(0)                           = ?
```

Neben gcc, g++, as, ld, gprof und gdb/ddd sind die folgenden Tools von Interesse.

Die GNU-Binutils:

- nm
- objdump
- objcopy
- readelf
- strip
- size
- c++filt
- ar
- ranlib

```
> file ./swap1
./swap1: ELF 64-bit LSB executable, x86-64, version 1 (SYSV),
   dynamically linked (uses shared libs), for GNU/Linux 2.6.15, not stripped
> nm ./swap1
0000000000600e20 d _DYNAMIC
0000000000600fe8 d _GLOBAL_OFFSET_TABLE_
0000000000400a75 t _GLOBAL__I_main
0000000000400bd8 R _IO_stdin_used
                 w _Jv_RegisterClasses
0000000000400a35 t _Z41__static_initialization_and_destruction_0ii
0000000000400ab6 W _Z4swapIdEvRT_S1_
0000000000400a8a W _Z4swapIiEvRT_S1_
                 U _ZNSolsEPFRSoS_E@@GLIBCXX_3.4
                 U _ZNSolsEd@@GLIBCXX_3.4
                 U _ZNSolsEi@@GLIBCXX_3.4
                 U _ZNSt8ios_base4InitC1Ev@@GLIBCXX_3.4
                 U _ZNSt8ios_base4InitD1Ev@@GLIBCXX_3.4
0000000000601060 B _ZSt4cout@@GLIBCXX_3.4
                 U _ZSt4endlIcSt11char_traitsIcEERSt13basic_ostreamIT_T0_ES6_@@GLIBCXX_3.
0000000000601180 b _ZStL8__ioinit
                 U _ZStlsISt11char_traitsIcEERSt13basic_ostreamIcT_ES5_PKc@@GLIBCXX_3.4
0000000000600e00 d __CTOR_END__
0000000000600df0 d __CTOR_LIST__
0000000000600e10 D __DTOR_END__
0000000000600e08 d __DTOR_LIST__
```

```
0000000000400d40 r __FRAME_END__
0000000000600e18 d __JCR_END__
0000000000600e18 d __JCR_LIST__
0000000000601060 A __bss_start
                 U __cxa_atexit@@GLIBC_2.2.5
0000000000601050 D __data_start
0000000000400b90 t __do_global_ctors_aux
0000000000400850 t __do_global_dtors_aux
0000000000601058 D __dso_handle
                 w __gmon_start__
                 U __gxx_personality_v0@@CXXABI_1.3
0000000000600dec d __init_array_end
0000000000600dec d __init_array_start
0000000000400af0 T __libc_csu_fini
0000000000400b00 T __libc_csu_init
                 U __libc_start_main@@GLIBC_2.2.5
0000000000601060 A _edata
0000000000601188 A _end
0000000000400bc8 T _fini
0000000000400730 T _init
0000000000400800 T _start
000000000040082c t call_gmon_start
0000000000601170 b completed.7424
0000000000601050 W data_start
0000000000601178 b dtor_idx.7426
00000000004008c0 t frame_dummy
00000000004008e4 T main
```

... und mit demangled Symbolen:

```
nm ./swap1 | c++filt
0000000000600e20 d _DYNAMIC
0000000000600fe8 d _GLOBAL_OFFSET_TABLE_
0000000000400a75 t global constructors keyed to main
0000000000400bd8 R _IO_stdin_used
                 w _Jv_RegisterClasses
0000000000400a35 t __static_initialization_and_destruction_0(int, int)
0000000000400ab6 W void swap<double>(double&, double&)
0000000000400a8a W void swap<int>(int&, int&)
                 U std::basic_ostream<char, std::char_traits<char> >::operator<<(
                 U std::basic_ostream<char, std::char_traits<char> >::operator<<(
                 U std::basic_ostream<char, std::char_traits<char> >::operator<<(
                 U std::ios_base::Init::Init()@@GLIBCXX_3.4
                 U std::ios_base::Init::~Init()@@GLIBCXX_3.4
```

```
0000000000601060 B std::cout@@GLIBCXX_3.4
                 U std::basic_ostream<char, std::char_traits<char> >& std::endl<char, std
0000000000601180 b std::__ioinit
                 U std::basic_ostream<char, std::char_traits<char> >& std::operator<< <st
0000000000600e00 d __CTOR_END__
0000000000600df0 d __CTOR_LIST__
0000000000600e10 D __DTOR_END__
0000000000600e08 d __DTOR_LIST__
0000000000400d40 r __FRAME_END__
0000000000600e18 d __JCR_END__
0000000000600e18 d __JCR_LIST__
0000000000601060 A __bss_start
                 U __cxa_atexit@@GLIBC_2.2.5
0000000000601050 D __data_start
0000000000400b90 t __do_global_ctors_aux
0000000000400850 t __do_global_dtors_aux
0000000000601058 D __dso_handle
                 w __gmon_start__
                 U __gxx_personality_v0@@CXXABI_1.3
0000000000600dec d __init_array_end
0000000000600dec d __init_array_start
0000000000400af0 T __libc_csu_fini
0000000000400b00 T __libc_csu_init
                 U __libc_start_main@@GLIBC_2.2.5
0000000000601060 A _edata
0000000000601188 A _end
0000000000400bc8 T _fini
0000000000400730 T _init
0000000000400800 T _start
000000000040082c t call_gmon_start
0000000000601170 b completed.7424
0000000000601050 W data_start
0000000000601178 b dtor_idx.7426
00000000004008c0 t frame_dummy
00000000004008e4 T main

> c++filt -n _Z4swapIiEvRT_S1_
void swap<int>(int&, int&)
```

Vergleiche:
C++ name mangling
names in object files
name mangling in Java
Getting the best from g++

```
> objdump -x swap1 | c++filt

swap1:     file format elf64-x86-64
swap1
architecture: i386:x86-64, flags 0x00000112:
EXEC_P, HAS_SYMS, D_PAGED
start address 0x00000000004007e0

Program Header:
    PHDR off    0x0000000000000040 vaddr 0x0000000000400040 paddr 0x0000000000400040 align 2**3
         filesz 0x00000000000001f8 memsz 0x00000000000001f8 flags r-x
  INTERP off    0x0000000000000238 vaddr 0x0000000000400238 paddr 0x0000000000400238 align 2**0
         filesz 0x000000000000001c memsz 0x000000000000001c flags r--
...
Dynamic Section:
  NEEDED               libstdc++.so.6
  NEEDED               libm.so.6
  NEEDED               libgcc_s.so.1
  NEEDED               libc.so.6
  INIT                 0x0000000000400720
  FINI                 0x0000000000400ba8
...
Version References:
  required from libc.so.6:
    0x09691a75 0x00 03 GLIBC_2.2.5
  required from libstdc++.so.6:
    0x08922974 0x00 02 GLIBCXX_3.4
Sections:
Idx Name          Size      VMA               LMA               File off  Algn
  0 .interp       0000001c  0000000000400238  0000000000400238  00000238  2**0
                  CONTENTS, ALLOC, LOAD, READONLY, DATA
  1 .note.ABI-tag 00000020  0000000000400254  0000000000400254  00000254  2**2
                  CONTENTS, ALLOC, LOAD, READONLY, DATA
  2 .note.SuSE    00000018  0000000000400274  0000000000400274  00000274  2**2
                  CONTENTS, ALLOC, LOAD, READONLY, DATA
  3 .note.gnu.build-id 00000024  000000000040028c  000000000040028c  0000028c  2**2
                  CONTENTS, ALLOC, LOAD, READONLY, DATA
  4 .hash         00000048  00000000004002b0  00000000004002b0  000002b0  2**3
                  CONTENTS, ALLOC, LOAD, READONLY, DATA
  5 .gnu.hash     00000030  00000000004002f8  00000000004002f8  000002f8  2**3
                  CONTENTS, ALLOC, LOAD, READONLY, DATA
  6 .dynsym       00000138  0000000000400328  0000000000400328  00000328  2**3
                  CONTENTS, ALLOC, LOAD, READONLY, DATA
  7 .dynstr       0000015e  0000000000400460  0000000000400460  00000460  2**0
                  CONTENTS, ALLOC, LOAD, READONLY, DATA
  8 .gnu.version  0000001a  00000000004005be  00000000004005be  000005be  2**1
                  CONTENTS, ALLOC, LOAD, READONLY, DATA
  9 .gnu.version_r 00000040  00000000004005d8  00000000004005d8  000005d8  2**3
                  CONTENTS, ALLOC, LOAD, READONLY, DATA
 10 .rela.dyn     00000030  0000000000400618  0000000000400618  00000618  2**3
                  CONTENTS, ALLOC, LOAD, READONLY, DATA
 11 .rela.plt     000000d8  0000000000400648  0000000000400648  00000648  2**3
                  CONTENTS, ALLOC, LOAD, READONLY, DATA
 12 .init         00000018  0000000000400720  0000000000400720  00000720  2**2
                  CONTENTS, ALLOC, LOAD, READONLY, CODE
 13 .plt          000000a0  0000000000400738  0000000000400738  00000738  2**2
                  CONTENTS, ALLOC, LOAD, READONLY, CODE
 14 .text         000003c8  00000000004007e0  00000000004007e0  000007e0  2**4
                  CONTENTS, ALLOC, LOAD, READONLY, CODE
 15 .fini         0000000e  0000000000400ba8  0000000000400ba8  00000ba8  2**2
                  CONTENTS, ALLOC, LOAD, READONLY, CODE
 16 .rodata       00000006  0000000000400bb8  0000000000400bb8  00000bb8  2**2
                  CONTENTS, ALLOC, LOAD, READONLY, DATA
 17 .eh_frame_hdr 00000044  0000000000400bc0  0000000000400bc0  00000bc0  2**2
                  CONTENTS, ALLOC, LOAD, READONLY, DATA
 18 .eh_frame     00000104  0000000000400c08  0000000000400c08  00000c08  2**3
```

```
                 CONTENTS, ALLOC, LOAD, READONLY, DATA
 19 .ctors       00000018  0000000000600de0  0000000000600de0  00000de0  2**3
                 CONTENTS, ALLOC, LOAD, DATA
 20 .dtors       00000010  0000000000600df8  0000000000600df8  00000df8  2**3
                 CONTENTS, ALLOC, LOAD, DATA
...
25 .data         00000010  0000000000601048  0000000000601048  00001048  2**3
                 CONTENTS, ALLOC, LOAD, DATA
26 .bss          00000128  0000000000601060  0000000000601060  00001058  2**5
                 ALLOC
...
SYMBOL TABLE:
0000000000400238 l    d  .interp        0000000000000000              .interp
0000000000400254 l    d  .note.ABI-tag  0000000000000000              .note.ABI-tag
0000000000400274 l    d  .note.SuSE     0000000000000000              .note.SuSE
000000000040028c l    d  .note.gnu.build-id     0000000000000000              .note.gnu.build-id
00000000004002b0 l    d  .hash  0000000000000000              .hash
00000000004002f8 l    d  .gnu.hash      0000000000000000              .gnu.hash
0000000000400328 l    d  .dynsym        0000000000000000              .dynsym
0000000000400460 l    d  .dynstr        0000000000000000              .dynstr
00000000004005be l    d  .gnu.version   0000000000000000              .gnu.version
00000000004005d8 l    d  .gnu.version_r 0000000000000000              .gnu.version_r
0000000000400618 l    d  .rela.dyn      0000000000000000              .rela.dyn
0000000000400648 l    d  .rela.plt      0000000000000000              .rela.plt
0000000000400720 l    d  .init  0000000000000000              .init
0000000000400738 l    d  .plt   0000000000000000              .plt
00000000004007e0 l    d  .text  0000000000000000              .text
0000000000400ba8 l    d  .fini  0000000000000000              .fini
0000000000400bb8 l    d  .rodata        0000000000000000              .rodata
...
0000000000400a96  w    F .text  0000000000000032              void swap<double>(double&, double&)
...
0000000000400a6a  w    F .text  000000000000002c              void swap<int>(int&, int&)
0000000000601058 g       *ABS*  0000000000000000              _edata
00000000004008c4 g     F .text  0000000000000151              main
0000000000400720 g     F .init  0000000000000000              _init


> objdump -t swap1 | c++filt
swap1:     file format elf64-x86-64
SYMBOL TABLE:
0000000000400238 l    d  .interp        0000000000000000              .interp
0000000000400254 l    d  .note.ABI-tag  0000000000000000              .note.ABI-tag
0000000000400274 l    d  .note.SuSE     0000000000000000              .note.SuSE
000000000040028c l    d  .note.gnu.build-id     0000000000000000              .note.gnu.build-id
00000000004002b0 l    d  .hash  0000000000000000              .hash
00000000004002f8 l    d  .gnu.hash      0000000000000000              .gnu.hash
0000000000400328 l    d  .dynsym        0000000000000000              .dynsym
0000000000400460 l    d  .dynstr        0000000000000000              .dynstr
00000000004005be l    d  .gnu.version   0000000000000000              .gnu.version
00000000004005d8 l    d  .gnu.version_r 0000000000000000              .gnu.version_r
0000000000400618 l    d  .rela.dyn      0000000000000000              .rela.dyn
0000000000400648 l    d  .rela.plt      0000000000000000              .rela.plt
0000000000400720 l    d  .init  0000000000000000              .init
0000000000400738 l    d  .plt   0000000000000000              .plt
00000000004007e0 l    d  .text  0000000000000000              .text
0000000000400ba8 l    d  .fini  0000000000000000              .fini
0000000000400bb8 l    d  .rodata        0000000000000000              .rodata
0000000000400bc0 l    d  .eh_frame_hdr  0000000000000000              .eh_frame_hdr
0000000000400c08 l    d  .eh_frame      0000000000000000              .eh_frame
0000000000600de0 l    d  .ctors 0000000000000000              .ctors
0000000000600df8 l    d  .dtors 0000000000000000              .dtors
0000000000600e08 l    d  .jcr   0000000000000000              .jcr
0000000000600e10 l    d  .dynamic       0000000000000000              .dynamic
0000000000600fe0 l    d  .got   0000000000000000              .got
0000000000600fe8 l    d  .got.plt       0000000000000000              .got.plt
0000000000601048 l    d  .data  0000000000000000              .data
```

```
0000000000601060 l    d  .bss   0000000000000000              .bss
0000000000000000 l    d  .comment.SUSE.OPTs      0000000000000000              .comment.SUSE.OPTs
0000000000000000 l    d  .comment       0000000000000000              .comment
0000000000000000 l    d  .debug_aranges 0000000000000000              .debug_aranges
0000000000000000 l    d  .debug_pubnames         0000000000000000              .debug_pubnames
0000000000000000 l    d  .debug_info    0000000000000000              .debug_info
0000000000000000 l    d  .debug_abbrev  0000000000000000              .debug_abbrev
0000000000000000 l    d  .debug_line    0000000000000000              .debug_line
0000000000000000 l    d  .debug_str     0000000000000000              .debug_str
0000000000000000 l    d  .debug_loc     0000000000000000              .debug_loc
0000000000000000 l    d  .debug_pubtypes         0000000000000000              .debug_pubtypes
0000000000000000 l    d  .debug_ranges  0000000000000000              .debug_ranges
0000000000000000 l    df *ABS*  0000000000000000              init.c
0000000000000000 l    df *ABS*  0000000000000000              initfini.c
000000000040080c l     F .text  0000000000000000              call_gmon_start
0000000000000000 l    df *ABS*  0000000000000000              crtstuff.c
0000000000600de0 l     O .ctors 0000000000000000              __CTOR_LIST__
0000000000600df8 l     O .dtors 0000000000000000              __DTOR_LIST__
0000000000600e08 l     O .jcr   0000000000000000              __JCR_LIST__
0000000000400830 l     F .text  0000000000000000              __do_global_dtors_aux
0000000000601170 l     O .bss   0000000000000001              completed.5939
0000000000601178 l     O .bss   0000000000000008              dtor_idx.5941
00000000004008a0 l     F .text  0000000000000000              frame_dummy
0000000000000000 l    df *ABS*  0000000000000000              crtstuff.c
0000000000600df0 l     O .ctors 0000000000000000              __CTOR_END__
0000000000400d08 l     O .eh_frame       0000000000000000              __FRAME_END__
0000000000600e08 l     O .jcr   0000000000000000              __JCR_END__
0000000000400b70 l     F .text  0000000000000000              __do_global_ctors_aux
0000000000000000 l    df *ABS*  0000000000000000              initfini.c
0000000000000000 l    df *ABS*  0000000000000000              swap1.cpp
0000000000601180 l     O .bss   0000000000000001              std::__ioinit
0000000000400a15 l     F .text  0000000000000040              __static_initialization_and_destruction_0(int, int)
0000000000400a55 l     F .text  0000000000000015              global constructors keyed to main
0000000000000000 l    df *ABS*  0000000000000000              elf-init.c
0000000000600fe8 l     O .got.plt        0000000000000000              .hidden _GLOBAL_OFFSET_TABLE_
0000000000600ddc l       .ctors 0000000000000000              .hidden __init_array_end
0000000000600ddc l       .ctors 0000000000000000              .hidden __init_array_start
0000000000600e10 l     O .dynamic        0000000000000000              .hidden _DYNAMIC
0000000000601048  w      .data  0000000000000000              data_start
0000000000000000       F *UND*  0000000000000000              std::basic_ostream<char, std::char_traits<char> >::oper
0000000000000000       F *UND*  0000000000000000              std::basic_ostream<char, std::char_traits<char> >::oper
0000000000400b60 g     F .text  0000000000000002              __libc_csu_fini
00000000004007e0 g     F .text  0000000000000000              _start
0000000000000000  w      *UND*  0000000000000000              __gmon_start__
0000000000000000  w      *UND*  0000000000000000              _Jv_RegisterClasses
0000000000400ba8 g     F .fini  0000000000000000              _fini
0000000000000000       F *UND*  0000000000000000              std::ios_base::Init::Init()@@GLIBCXX_3.4
0000000000000000       F *UND*  0000000000000000              __libc_start_main@@GLIBC_2.2.5
0000000000000000       F *UND*  0000000000000000              __cxa_atexit@@GLIBC_2.2.5
0000000000400798       F *UND*  0000000000000000              std::ios_base::Init::~Init()@@GLIBCXX_3.4
0000000000000000       F *UND*  0000000000000000              std::basic_ostream<char, std::char_traits<char> >& std:
0000000000400bb8 g     O .rodata         0000000000000004              _IO_stdin_used
0000000000601048 g       .data  0000000000000000              __data_start
0000000000400a96  w    F .text  0000000000000032              void swap<double>(double&, double&)
0000000000601060 g     O .bss   0000000000000110              std::cout@@GLIBCXX_3.4
0000000000601050 g     O .data  0000000000000000              .hidden __dso_handle
0000000000600e00 g     O .dtors 0000000000000000              .hidden __DTOR_END__
0000000000400ad0 g     F .text  0000000000000089              __libc_csu_init
0000000000601058 g       *ABS*  0000000000000000              __bss_start
0000000000601188 g       *ABS*  0000000000000000              _end
0000000000000000       F *UND*  0000000000000000              std::basic_ostream<char, std::char_traits<char> >::oper
00000000004007c8       F *UND*  0000000000000000              std::basic_ostream<char, std::char_traits<char> >& std:
0000000000400a6a  w    F .text  000000000000002c              void swap<int>(int&, int&)
0000000000601058 g       *ABS*  0000000000000000              _edata
00000000004008c4 g     F .text  0000000000000151              main
0000000000400720 g     F .init  0000000000000000              _init
```

```
> readelf -s swap1 | c++filt

Symbol table '.dynsym' contains 13 entries:
   Num:    Value          Size Type    Bind   Vis      Ndx Name
     0: 0000000000000000     0 NOTYPE  LOCAL  DEFAULT  UND
     1: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND std::basic_ostream<char, std::char_traits<char> >::operator<<(double)@GLIBCXX_3.4 (2)
     2: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND std::basic_ostream<char, std::char_traits<char> >::operator<<(int)@GLIBCXX_3.4 (2)
     3: 0000000000000000     0 NOTYPE  WEAK   DEFAULT  UND __gmon_start__
     4: 0000000000000000     0 NOTYPE  WEAK   DEFAULT  UND _Jv_RegisterClasses
     5: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND std::ios_base::Init::Init()@GLIBCXX_3.4 (2)
     6: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND __libc_start_main@GLIBC_2.2.5 (3)
     7: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND __cxa_atexit@GLIBC_2.2.5 (3)
     8: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND _ZStlsISt11char_traitsIcE@GLIBCXX_3.4 (2)
     9: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND std::basic_ostream<char, std::char_traits<char> >::operator<<(std::basic_ostream<char, std::char_traits<char> >&
    10: 00000000004007c8     0 FUNC    GLOBAL DEFAULT  UND _ZSt4endlIcSt11char_trait@GLIBCXX_3.4 (2)
    11: 0000000000400798     0 FUNC    GLOBAL DEFAULT  UND std::ios_base::Init::~Init()@GLIBCXX_3.4 (2)
    12: 0000000000601060   272 OBJECT  GLOBAL DEFAULT   27 std::cout@GLIBCXX_3.4 (2)

Symbol table '.symtab' contains 93 entries:
   Num:    Value          Size Type    Bind   Vis      Ndx Name
     0: 0000000000000000     0 NOTYPE  LOCAL  DEFAULT  UND
     1: 0000000000400238     0 SECTION LOCAL  DEFAULT    1
     2: 0000000000400254     0 SECTION LOCAL  DEFAULT    2
     3: 0000000000400274     0 SECTION LOCAL  DEFAULT    3
     4: 000000000040028c     0 SECTION LOCAL  DEFAULT    4
     5: 00000000004002b0     0 SECTION LOCAL  DEFAULT    5
     6: 00000000004002f8     0 SECTION LOCAL  DEFAULT    6
     7: 0000000000400328     0 SECTION LOCAL  DEFAULT    7
     8: 0000000000400460     0 SECTION LOCAL  DEFAULT    8
     9: 00000000004005be     0 SECTION LOCAL  DEFAULT    9
    10: 00000000004005d8     0 SECTION LOCAL  DEFAULT   10
    11: 0000000000400618     0 SECTION LOCAL  DEFAULT   11
    12: 0000000000400648     0 SECTION LOCAL  DEFAULT   12
    13: 0000000000400720     0 SECTION LOCAL  DEFAULT   13
    14: 0000000000400738     0 SECTION LOCAL  DEFAULT   14
    15: 00000000004007e0     0 SECTION LOCAL  DEFAULT   15
    16: 0000000000400ba8     0 SECTION LOCAL  DEFAULT   16
    17: 0000000000400bb8     0 SECTION LOCAL  DEFAULT   17
    18: 0000000000400bc0     0 SECTION LOCAL  DEFAULT   18
    19: 0000000000400c08     0 SECTION LOCAL  DEFAULT   19
    20: 0000000000600de0     0 SECTION LOCAL  DEFAULT   20
    21: 0000000000600df8     0 SECTION LOCAL  DEFAULT   21
    22: 0000000000600e08     0 SECTION LOCAL  DEFAULT   22
    23: 0000000000600e10     0 SECTION LOCAL  DEFAULT   23
    24: 0000000000600fe0     0 SECTION LOCAL  DEFAULT   24
    25: 0000000000600fe8     0 SECTION LOCAL  DEFAULT   25
    26: 0000000000601048     0 SECTION LOCAL  DEFAULT   26
    27: 0000000000601060     0 SECTION LOCAL  DEFAULT   27
    28: 0000000000000000     0 SECTION LOCAL  DEFAULT   28
    29: 0000000000000000     0 SECTION LOCAL  DEFAULT   29
    30: 0000000000000000     0 SECTION LOCAL  DEFAULT   30
    31: 0000000000000000     0 SECTION LOCAL  DEFAULT   31
    32: 0000000000000000     0 SECTION LOCAL  DEFAULT   32
    33: 0000000000000000     0 SECTION LOCAL  DEFAULT   33
    34: 0000000000000000     0 SECTION LOCAL  DEFAULT   34
    35: 0000000000000000     0 SECTION LOCAL  DEFAULT   35
    36: 0000000000000000     0 SECTION LOCAL  DEFAULT   36
    37: 0000000000000000     0 SECTION LOCAL  DEFAULT   37
    38: 0000000000000000     0 SECTION LOCAL  DEFAULT   38
    39: 0000000000000000     0 FILE    LOCAL  DEFAULT  ABS init.c
    40: 0000000000000000     0 FILE    LOCAL  DEFAULT  ABS initfini.c
    41: 000000000040080c     0 FUNC    LOCAL  DEFAULT   15 call_gmon_start
    42: 0000000000000000     0 FILE    LOCAL  DEFAULT  ABS crtstuff.c
    43: 0000000000600de0     0 OBJECT  LOCAL  DEFAULT   20 __CTOR_LIST__
    44: 0000000000600df8     0 OBJECT  LOCAL  DEFAULT   21 __DTOR_LIST__
    45: 0000000000600e08     0 OBJECT  LOCAL  DEFAULT   22 __JCR_LIST__
    46: 0000000000400830     0 FUNC    LOCAL  DEFAULT   15 __do_global_dtors_aux
    47: 0000000000601170     1 OBJECT  LOCAL  DEFAULT   27 completed.5939
    48: 0000000000601178     8 OBJECT  LOCAL  DEFAULT   27 dtor_idx.5941
    49: 00000000004008a0     0 FUNC    LOCAL  DEFAULT   15 frame_dummy
    50: 0000000000000000     0 FILE    LOCAL  DEFAULT  ABS crtstuff.c
    51: 0000000000600df0     0 OBJECT  LOCAL  DEFAULT   20 __CTOR_END__
    52: 0000000000400d08     0 OBJECT  LOCAL  DEFAULT   19 __FRAME_END__
    53: 0000000000600e08     0 OBJECT  LOCAL  DEFAULT   22 __JCR_END__
    54: 0000000000400b70     0 FUNC    LOCAL  DEFAULT   15 __do_global_ctors_aux
    55: 0000000000000000     0 FILE    LOCAL  DEFAULT  ABS initfini.c
    56: 0000000000000000     0 FILE    LOCAL  DEFAULT  ABS swap1.cpp
    57: 0000000000601180     1 OBJECT  LOCAL  DEFAULT   27 std::__ioinit
    58: 0000000000400a15    64 FUNC    LOCAL  DEFAULT   15 _Z41__static_initializati
    59: 0000000000400a55    21 FUNC    LOCAL  DEFAULT   15 global constructors keyed to main
    60: 0000000000000000     0 FILE    LOCAL  DEFAULT  ABS elf-init.c
    61: 0000000000600fe8     0 OBJECT  LOCAL  HIDDEN   25 _GLOBAL_OFFSET_TABLE_
    62: 0000000000600ddc     0 NOTYPE  LOCAL  HIDDEN   20 __init_array_end
    63: 0000000000600ddc     0 NOTYPE  LOCAL  HIDDEN   20 __init_array_start
    64: 0000000000600e10     0 OBJECT  LOCAL  HIDDEN   23 _DYNAMIC
    65: 0000000000601048     0 NOTYPE  WEAK   DEFAULT   26 data_start
    66: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND std::basic_ostream<char, std::char_traits<char> >::operator<<(double)@@GLIBCXX_3.4
    67: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND std::basic_ostream<char, std::char_traits<char> >::operator<<(int)@@GLIBCXX_3.4
    68: 0000000000400b60     2 FUNC    GLOBAL DEFAULT   15 __libc_csu_fini
    69: 00000000004007e0     0 FUNC    GLOBAL DEFAULT   15 _start
```

```
70: 0000000000000000     0 NOTYPE  WEAK   DEFAULT  UND __gmon_start__
71: 0000000000000000     0 NOTYPE  WEAK   DEFAULT  UND _Jv_RegisterClasses
72: 0000000000400ba8     0 FUNC    GLOBAL DEFAULT   16 _fini
73: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND std::ios_base::Init::Init()@@
74: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND __libc_start_main@@GLIBC_
75: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND __cxa_atexit@@GLIBC_2.2.5
76: 0000000000400798     0 FUNC    GLOBAL DEFAULT  UND std::ios_base::Init::~Init()@@
77: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND _ZStlsISt11char_traitsIcE
78: 0000000000400bb8     4 OBJECT  GLOBAL DEFAULT   17 _IO_stdin_used
79: 0000000000601048     0 NOTYPE  GLOBAL DEFAULT   26 __data_start
80: 0000000000400a96    50 FUNC    WEAK   DEFAULT   15 void swap<double>(double&, double&)
81: 0000000000601060   272 OBJECT  GLOBAL DEFAULT   27 std::cout@@GLIBCXX_3.4
82: 0000000000601050     0 OBJECT  GLOBAL HIDDEN   26 __dso_handle
83: 0000000000600e00     0 OBJECT  GLOBAL HIDDEN   21 __DTOR_END__
84: 0000000000400ad0   137 FUNC    GLOBAL DEFAULT   15 __libc_csu_init
85: 0000000000601058     0 NOTYPE  GLOBAL DEFAULT  ABS __bss_start
86: 0000000000601188     0 NOTYPE  GLOBAL DEFAULT  ABS _end
87: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND std::basic_ostream<char, std::char_traits<char> >::operator<<(std::basic_ostream<char, std::char_t
88: 00000000004007c8     0 FUNC    GLOBAL DEFAULT  UND _ZSt4endlIcSt11char_trait
89: 0000000000400a6a    44 FUNC    WEAK   DEFAULT   15 void swap<int>(int&, int&)
90: 0000000000601058     0 NOTYPE  GLOBAL DEFAULT  ABS _edata
91: 00000000004008c4   337 FUNC    GLOBAL DEFAULT   15 main
92: 0000000000400720     0 FUNC    GLOBAL DEFAULT   13 _init
```

Die Sektionstypen von Objektdateien:
test, data und bss

Hinweis zu verfügbaren Softwareentwicklungssystemen:
GNU g++ für Linux
cygwin für Windows
Visual Studio 2010 für Windows

1.5.1 Erstellen und Benutzen von statischen Bibliotheken

*.a-Bibiotheken als Sammlungen von Objektdateien

Erzeugen statischer Bibliotheken

ar Manualpage

Static Libraries

```
> cat swap1.cpp
#include <iostream>
template <typename T>
/*
 * Requirements: T muss einen Kopierkonstruktor haben,
 *               T muss einen Zuweisungsoperator zu T haben.
 */
void swap(T& a, T& b)
{
   T  old_a(a);

   a = b;
   b = old_a;
}

int main(){
   int k(1);
   int l(5);
   std::cout << k << " " << l << std::endl;
   swap(k,l);
   std::cout << k << " " << l << std::endl;

   double d1(3.1415);
   double d2(15.1055);
   std::cout << d1 << " " << d2 << std::endl;
   swap(d1, d2);
   std::cout << d1 << " " << d2 << std::endl;
}

> make CXXFLAGS=-g swap1
g++ -g  swap1.cpp   -o swap1
> nm swap1 | grep swap | c++filt
0000000000400a96 W void swap<double>(double&, double&)
```

```
0000000000400a6a W void swap<int>(int&, int&)

> g++ -c swap1.cpp
> ls -al swap1.o
-rw-r--r-- 1 user1 users 4464  9. Nov 13:59 swap1.o

> ar rc libswap.a swap1.o
> ls -al libswap.a
-rw-r--r-- 1 user1 users 4650  9. Nov 14:02 libswap.a

> nm libswap.a | c++filt
0000000000000000 W void swap<double>(double&, double&)
0000000000000000 W void swap<int>(int&, int&)
                 U std::basic_ostream<char, std::char_traits<char> >::operator<<(
                 U std::basic_ostream<char, std::char_traits<char> >::operator<<(
0000000000000000 T main
```

oder ein vollständiges Beispiel:

```
> cat person.h
/*
 * person.h
 */
class Person
{
 public:
  Person(){};
  ~Person(){};

  void speak(const char * sentence);
};

> cat person.cpp
#include "person.h"
#include <iostream>

void Person::speak(const char * sentence)
{
  std::cout << sentence << std::endl;
}

> cat main.cpp
/*
 * main.cpp
```

```
 */

#include "person.h"
#include <iostream>

int main()
{
  Person person;
  person.speak("Hello world!");

  return 0;
}

> g++ -c person.cpp
> g++ -c main.cpp
> ar rc libperson.a person.o
> g++ -o main main.o -L. -lperson
> ldd main
        linux-vdso.so.1 =>  (0x00007fffdfbff000)
        libstdc++.so.6 => /usr/lib64/libstdc++.so.6 (0x00007f1f48ef2000)
        libm.so.6 => /lib64/libm.so.6 (0x00007f1f48c9b000)
        libgcc_s.so.1 => /lib64/libgcc_s.so.1 (0x00007f1f48a85000)
        libc.so.6 => /lib64/libc.so.6 (0x00007f1f48725000)
        /lib64/ld-linux-x86-64.so.2 (0x00007f1f491fc000)

> ls -al main
-rwxr-xr-x 1 user1 users 13015  9. Nov 14:13 main
```

Bei den impliziten make-Regeln benutzte Environment-Variablen

1.5.2 Erstellen und Benutzen einer „shared object“- Bibliothek

```
> g++ -fPIC -c person.cpp
> g++ -shared -o libperson.so person.o
> g++ -o main main.o -L. -lperson
> ls -al main
-rwxr-xr-x 1 user1 users 12570  9. Nov 16:27 main

> ldd main
        linux-vdso.so.1 =>  (0x00007fffc85fa000)
        libperson.so => not found
        libstdc++.so.6 => /usr/lib64/libstdc++.so.6 (0x00007f990e302000)
        libm.so.6 => /lib64/libm.so.6 (0x00007f990e0ab000)
```

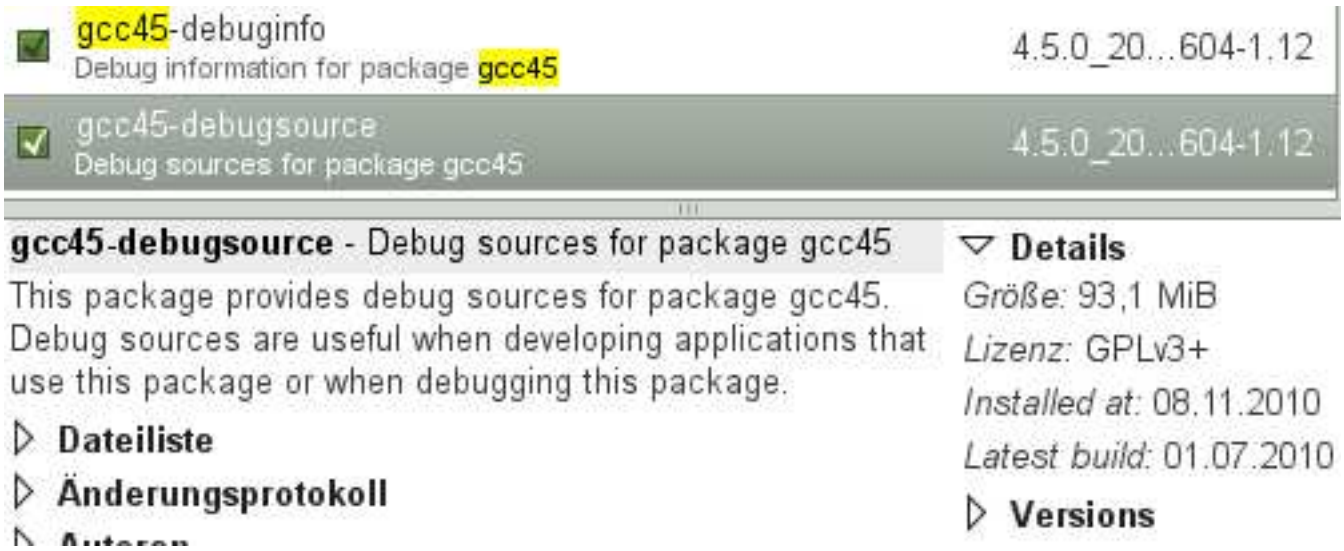
```
        libgcc_s.so.1 => /lib64/libgcc_s.so.1 (0x00007f990de95000)
        libc.so.6 => /lib64/libc.so.6 (0x00007f990db35000)
        /lib64/ld-linux-x86-64.so.2 (0x00007f990e60c000)
> ./main
./main: error while loading shared libraries: libperson.so: cannot
   open shared object file: No such file or directory

> export LD_LIBRARY_PATH=.
> ldd main
        linux-vdso.so.1 =>  (0x00007fff6ebff000)
        libperson.so => ./libperson.so (0x00007f76ec213000)
        libstdc++.so.6 => /usr/lib64/libstdc++.so.6 (0x00007f76ebf09000)
        libm.so.6 => /lib64/libm.so.6 (0x00007f76ebcb2000)
        libgcc_s.so.1 => /lib64/libgcc_s.so.1 (0x00007f76eba9c000)
        libc.so.6 => /lib64/libc.so.6 (0x00007f76eb73c000)
        /lib64/ld-linux-x86-64.so.2 (0x00007f76ec415000)
> ./main
Hello world!
```

Oder besser (mit Versionsinformationen):
.so-Erzeugung mit Versionsnummern
YoLinux shared libraries
Creating dynamic libraries
.so Versionsnummern
Dissecting shared libraries
fix shared library load
Workarount für fehlende .so

1.5.3 Bibliotheksmanagement insbesondere unter verschiedenen Betriebssystemen

Using static and shared libraries across platforms
Writing and Using Libraries

1.5.4 STL-Templatequellen unter SuSE-Linux fürs zeilenweise Debuggen auch innerhalb der STL-Routinen



1.6 Automatisch überprüfte Requirements von Type-Template-Parametern

Ein traditionelles Template-Beispiel:

```
#include <iostream>
#include <cmath>
#include <limits>

using namespace std;

template <typename T1, typename T2>
double geomMittel2(const T1& a, const T2& b)
{
    return sqrt(abs(a*b));
}

int main()
{
    cout << geomMittel2(3.0, 300.0) << endl;
    cout << geomMittel2(3, 300.0) << endl;
    cout << geomMittel2(-3, 300.0) << endl;
    cout << geomMittel2(-3, 300) << endl;
    cout << geomMittel2(3.0, 'c') << endl;
    cout << geomMittel2(3.0, "c") << endl;

    return 0;
}
```

und die Fehlermeldung für den letzten Testfall:

```
In Funktion ≫double geomMittel2(const T1&, const T2&) [with T1 = double, T2 = char [2]]≪:
geomMittel2-0.cpp:20:33:   instantiated from here
geomMittel2-0.cpp:10:25: Fehler: ungültige Operanden der Typen ≫const double≪ und ≫const char [2]≪ für binäres ≫operator*≪
```

Nach einer verbesserten Bezeichnerwahl:

```
#include <iostream>
#include <cmath>
#include <limits>

using namespace std;

template <typename ArithmeticLike1, typename ArithmeticLike2>
double geomMittel2(const ArithmeticLike1& a, const
   ArithmeticLike2& b)
{
```

```
    return sqrt(abs(a*b));
}

int main()
{
    cout << geomMittel2(3.0, 300.0) << endl;
    cout << geomMittel2(3, 300.0) << endl;
    cout << geomMittel2(-3, 300.0) << endl;
    cout << geomMittel2(-3, 300) << endl;
    cout << geomMittel2(3.0, 'c') << endl;
    cout << geomMittel2(3.0, "c") << endl;

    return 0;
}
```

und nach einer maschinellen Überprüfung der Typeigenschaften der generischen Parameter:

1.6.1 mit Hilfe von BOOST_STATIC_ASSERT()

```
#include <iostream>
#include <cmath>
#include <limits>
#include <boost/static_assert.hpp>
#include <boost/type_traits.hpp>

using namespace std;

BOOST_STATIC_ASSERT(std::numeric_limits<int>::digits >= 32);

template <typename ArithmeticLike1, typename ArithmeticLike2>
double geomMittel2(const ArithmeticLike1& a, const
   ArithmeticLike2& b)
{
BOOST_STATIC_ASSERT(::boost::is_arithmetic<ArithmeticLike1>::
   value);
BOOST_STATIC_ASSERT(::boost::is_arithmetic<ArithmeticLike2>::
   value);

    return sqrt(abs(a*b));
}

int main()
{
```

```
    cout << geomMittel2(3.0, 300.0) << endl;
    cout << geomMittel2(3, 300.0) << endl;
    cout << geomMittel2(-3, 300.0) << endl;
    cout << geomMittel2(-3, 300) << endl;
    cout << geomMittel2(3.0, 'c') << endl;
    cout << geomMittel2(3.0, "c") << endl;

    return 0;
}
```

mit der Compiler-Fehlermeldung:

```
geomMittel2.cpp:9:1: Fehler: Ungültige Anwendung von ≫sizeof≪ auf unvollständigen Typen ≫boost::STATIC_ASSERTION_FAILURE<false>≪
geomMittel2.cpp: In Funktion ≫double geomMittel2(const ArithmeticLike1&, const ArithmeticLike2&) [with ArithmeticLike1 = double, ArithmeticLike2 = char [2]
geomMittel2.cpp:28:33:   instantiated from here
geomMittel2.cpp:15:1: Fehler: Ungültige Anwendung von ≫sizeof≪ auf unvollständigen Typen ≫boost::STATIC_ASSERTION_FAILURE<false>≪
geomMittel2.cpp:17:25: Fehler: ungültige Operanden der Typen ≫const double≪ und ≫const char [2]≪ für binäres ≫operator*≪
```

oder

1.6.2 mit Hilfe des c++0x-Modus des g++

```
#include <iostream>
#include <cmath>
#include <limits>
#include <boost/type_traits.hpp>

using namespace std;

static_assert(std::numeric_limits<int>::digits >= 32, "int not
   enough digits");

template <typename ArithmeticLike1, typename ArithmeticLike2>
double geomMittel2(const ArithmeticLike1& a, const
   ArithmeticLike2& b)
{
static_assert(::boost::is_arithmetic<ArithmeticLike1>::value, "
   ArithmeticLike1 is not arithmetic");
static_assert(::boost::is_arithmetic<ArithmeticLike2>::value, "
   ArithmeticLike2 is not arithmetic");

    return sqrt(abs(a*b));
}

// uebersetze mit -std=c++0x
// oder make CXXFLAGS="-std=c++0x" ...

int main()
```

```
{
    cout << geomMittel2(3.0, 300.0) << endl;
    cout << geomMittel2(3, 300.0) << endl;
    cout << geomMittel2(-3, 300.0) << endl;
    cout << geomMittel2(-3, 300) << endl;
    cout << geomMittel2(3.0, 'c') << endl;
    cout << geomMittel2(3.0, "c") << endl;

    return 0;
}
```

mit der Compiler-Fehlermeldung:

```
geomMittel-sa2.cpp: In Funktion »double geomMittel2(const ArithmeticLike1&, const ArithmeticLike2&) [with ArithmeticLike1 = double, ArithmeticLike2 = char [2]]«:
geomMittel-sa2.cpp:29:33:   instantiated from here
geomMittel-sa2.cpp:14:1: Fehler: statische Behauptung gescheitert: "ArithmeticLike2 is not arithmetic"
geomMittel-sa2.cpp:16:25: Fehler: ungültige Operanden der Typen »const double« und »const char [2]« für binäres »operator*«
make: *** [geomMittel-sa2] Fehler 1
```

1.6.3 C++0x type_traits

Abschnitt 20.7.2:

```
template <class T> struct is_void;
template <class T> struct is_integral;
...
template <class T> struct is_arithmetic;
...
template <class T> struct is_const;
template <class T> struct is_trivially_copyable;
...
template <class T> struct is_abstract;

template <class T> struct is_same;
template <class T> struct is_base_of;
template <class T> struct is_convertible;
```

1.6.4 BOOST type_traits

Boost: Categorizing a Type

```
template <class T> struct is_array;
template <class T> struct is_complex;
template <class T> struct is_void;
template <class T> struct is_integral;
...
template <class T> struct is_arithmetic;
...
```

```
template <class T> struct is_const;
template <class T> struct is_trivially_copyable;
...
template <class T> struct is_abstract;

template <class T> struct is_same;
template <class T> struct is_base_of;
template <class T> struct is_convertible;
template <class T> struct has_new_operator;
template <class T> struct has_nothrow_assign;
template <class T> struct has_nothrow_constructor;
...
template <class T> struct is_empty;
template <class T> struct is_polymorphic;
template <class T> struct has virtual_destructor;
```

1.7 numeric_limits als Typ-Abbildung

numeric_limits als generische Klasse

mit traits-ähnlichem Charakter für die Benutung zum Beispiel für Requirements von Tempate-Parametern:
UnsignedInt Template-Parameter

```
#include <limits>
#include <boost/static_assert.hpp>

template <class UnsignedInt>
class myclass
{
private:
  static_assert(std::numeric_limits<UnsignedInt>::digits >= 16,
                "UnsignedInt isn't long enough");
  static_assert(std::numeric_limits<UnsignedInt>::is_specialized
,
                "UnsignedInt isn't specialized");
  static_assert(std::numeric_limits<UnsignedInt>::is_integer,
                "UnsignedInt isn't integer");
  static_assert(!std::numeric_limits<UnsignedInt>::is_signed,
                "UnsignedInt isn't unsigned");

public:
   /* details here */
};
 myclass<unsigned> m1;
 //myclass<int> m2;
 myclass<unsigned char> m3;

int main()
{
 return 0;
}
```

1.8 Template-Deklarationen zur Erzeugung von Objektdateien mit einer Ansammlung von Template-Instanzen

```
#include <iostream>
#include <cmath>
using namespace std;

template <typename ArithmeticLike1, typename ArithmeticLike2>
double geomMittel2(const ArithmeticLike1& a,
                   const ArithmeticLike2& b)
{
    return sqrt(abs(a*b));
}

template double geomMittel2<short, float>(const short&, const
   float&);
template double geomMittel2<int, float>(const int&, const float
   &);
template double geomMittel2<long, float>(const long&, const
   float&);
template double geomMittel2<float, float>(const float&, const
   float&);
template double geomMittel2<double, float>(const double&, const
    float&);
template double geomMittel2<long double, float>(const long
   double&, const float&);
// ...
template double geomMittel2<short, double>(const short&, const
   double&);
template double geomMittel2<int, double>(const int&, const
   double&);
template double geomMittel2<long, double>(const long&, const
   double&);
template double geomMittel2<float, double>(const float&, const
   double&);
template double geomMittel2<double, double>(const double&,
   const double&);
template double geomMittel2<long double, double>(const long
   double&, const double&);
// ...
template double geomMittel2<short, long double>(const short&,
```

```
    const long double&);
template double geomMittel2<int, long double>(const int&, const
    long double&);
template double geomMittel2<long, long double>(const long&,
    const long double&);
template double geomMittel2<float, long double>(const float&,
    const long double&);
template double geomMittel2<double, long double>(const double&,
    const long double&);
template double geomMittel2<long double, long double>(const
    long double&, const long double&);
```

1.9 Wo ist die Template-Instanz?

Abschnitt 7.5: Where's the template?
SunWorkshop C++ Template Repository: CCadmin, SunWS_cache Unterverzeichnisse, ...
Abschnitt 7.9: GNU-Compilerunterstützung für Type Traits

g++-Compileroptionen mit Template-Relevanz:

```
-fno-implicit-templates
-fno-implicit-inline-templates
-fno-pretty-templates
-frepo
```

(siehe g++-Manual, Kapitel 3).

1.10 C++0x extern template

C++0x Draft

1.11 Orte, wo statische Zusicherungen benutzt werden

Boost.StaticAssert

1.12 Fehlermeldungen bei uneingeschränkter Generizität

```
#include <vector>
#include <complex>
#include <algorithm>

int main()
 {
     std::vector<std::complex<float> > v;
     std::stable_sort(v.begin(), v.end());
 }
```

und die Fehlermeldung:

```
In file included from /usr/include/c++/4.5/algorithm:63:0,
                 from bad_error_eg.cpp:3:
/usr/include/c++/4.5/bits/stl_algo.h: In Funktion »void std::__insertion_sort(_RandomAccessIterator, _RandomAccessIterator) [witl
/usr/include/c++/4.5/bits/stl_algo.h:3358:4:   instantiated from »void std::__inplace_stable_sort(_RandomAccessIterator, _Random
/usr/include/c++/4.5/bits/stl_algo.h:5415:2:   instantiated from »void std::stable_sort(_RAIter, _RAIter) [with _RAIter = __gnu_
bad_error_eg.cpp:8:41:   instantiated from here
/usr/include/c++/4.5/bits/stl_algo.h:2103:4: Fehler: no match for »operator<« in »__i.__gnu_cxx::__normal_iterator<_Iterator, _
/usr/include/c++/4.5/bits/stl_algo.h: In Funktion »void std::__merge_without_buffer(_BidirectionalIterator, _BidirectionalIterato
/usr/include/c++/4.5/bits/stl_algo.h:3364:7:   instantiated from »void std::__inplace_stable_sort(_RandomAccessIterator, _Random
/usr/include/c++/4.5/bits/stl_algo.h:5415:2:   instantiated from »void std::stable_sort(_RAIter, _RAIter) [with _RAIter = __gnu_
bad_error_eg.cpp:8:41:   instantiated from here
/usr/include/c++/4.5/bits/stl_algo.h:2963:4: Fehler: no match for »operator<« in »__middle.__gnu_cxx::__normal_iterator<_Iterato
/usr/include/c++/4.5/bits/stl_algo.h: In Funktion »void std::__unguarded_linear_insert(_RandomAccessIterator) [with _RandomAccess
/usr/include/c++/4.5/bits/stl_algo.h:2111:6:   instantiated from »void std::__insertion_sort(_RandomAccessIterator, _RandomAccess
/usr/include/c++/4.5/bits/stl_algo.h:3358:4:   instantiated from »void std::__inplace_stable_sort(_RandomAccessIterator, _Random
/usr/include/c++/4.5/bits/stl_algo.h:5415:2:   instantiated from »void std::stable_sort(_RAIter, _RAIter) [with _RAIter = __gnu_
bad_error_eg.cpp:8:41:   instantiated from here
/usr/include/c++/4.5/bits/stl_algo.h:2064:7: Fehler: no match for »operator<« in »__val < __next.__gnu_cxx::__normal_iterator<_I
In file included from /usr/include/c++/4.5/vector:61:0,
                 from bad_error_eg.cpp:1:
/usr/include/c++/4.5/bits/stl_algobase.h: In Funktion »_ForwardIterator std::lower_bound(_ForwardIterator, _ForwardIterator, cons
/usr/include/c++/4.5/bits/stl_algo.h:2975:4:   instantiated from »void std::__merge_without_buffer(_BidirectionalIterator, _Bidi
/usr/include/c++/4.5/bits/stl_algo.h:3364:7:   instantiated from »void std::__inplace_stable_sort(_RandomAccessIterator, _Random
/usr/include/c++/4.5/bits/stl_algo.h:5415:2:   instantiated from »void std::stable_sort(_RAIter, _RAIter) [with _RAIter = __gnu_
bad_error_eg.cpp:8:41:   instantiated from here
/usr/include/c++/4.5/bits/stl_algobase.h:976:4: Fehler: no match for »operator<« in »__middle.__gnu_cxx::__normal_iterator<_Iter
In file included from /usr/include/c++/4.5/algorithm:63:0,
                 from bad_error_eg.cpp:3:
/usr/include/c++/4.5/bits/stl_algo.h: In Funktion »_FIter std::upper_bound(_FIter, _FIter, const _Tp&) [with _FIter = __gnu_cxx:
/usr/include/c++/4.5/bits/stl_algo.h:2982:4:   instantiated from »void std::__merge_without_buffer(_BidirectionalIterator, _Bidi
/usr/include/c++/4.5/bits/stl_algo.h:3364:7:   instantiated from »void std::__inplace_stable_sort(_RandomAccessIterator, _Random
/usr/include/c++/4.5/bits/stl_algo.h:5415:2:   instantiated from »void std::stable_sort(_RAIter, _RAIter) [with _RAIter = __gnu_
bad_error_eg.cpp:8:41:   instantiated from here
/usr/include/c++/4.5/bits/stl_algo.h:2461:4: Fehler: no match for »operator<« in »__val < __middle.__gnu_cxx::__normal_iterator<
/usr/include/c++/4.5/bits/stl_algo.h: In Funktion »_OIter std::merge(_IIter1, _IIter1, _IIter2, _IIter2, _OIter) [with _IIter1 =
/usr/include/c++/4.5/bits/stl_algo.h:2838:4:   instantiated from »void std::__merge_adaptive(_BidirectionalIterator, _Bidirectio
/usr/include/c++/4.5/bits/stl_algo.h:3315:7:   instantiated from »void std::__stable_sort_adaptive(_RandomAccessIterator, _Rando
/usr/include/c++/4.5/bits/stl_algo.h:5417:2:   instantiated from »void std::stable_sort(_RAIter, _RAIter) [with _RAIter = __gnu_
bad_error_eg.cpp:8:41:   instantiated from here
```

```
/usr/include/c++/4.5/bits/stl_algo.h:5299:4: Fehler: no match for ≫operator<≪ in ≫__first2.__gnu_cxx::__normal_itera
/usr/include/c++/4.5/bits/stl_algo.h: In Funktion ≫_BidirectionalIterator3 std::__merge_backward(_BidirectionalIterat
/usr/include/c++/4.5/bits/stl_algo.h:2847:4:   instantiated from ≫void std::__merge_adaptive(_BidirectionalIterator,
/usr/include/c++/4.5/bits/stl_algo.h:3315:7:   instantiated from ≫void std::__stable_sort_adaptive(_RandomAccessItera
/usr/include/c++/4.5/bits/stl_algo.h:5417:2:   instantiated from ≫void std::stable_sort(_RAIter, _RAIter) [with _RAIt
bad_error_eg.cpp:8:41:   instantiated from here
/usr/include/c++/4.5/bits/stl_algo.h:2740:4: Fehler: no match for ≫operator<≪ in ≫* __last2 < __last1.__gnu_cxx::__n
/usr/include/c++/4.5/bits/stl_algo.h: In Funktion ≫_OIter std::merge(_IIter1, _IIter1, _IIter2, _IIter2, _OIter) [wit
/usr/include/c++/4.5/bits/stl_algo.h:3163:4:   instantiated from ≫void std::__merge_sort_loop(_RandomAccessIterator1,
/usr/include/c++/4.5/bits/stl_algo.h:3261:4:   instantiated from ≫void std::__merge_sort_with_buffer(_RandomAccessIte
/usr/include/c++/4.5/bits/stl_algo.h:3312:4:   instantiated from ≫void std::__stable_sort_adaptive(_RandomAccessItera
/usr/include/c++/4.5/bits/stl_algo.h:5417:2:   instantiated from ≫void std::stable_sort(_RAIter, _RAIter) [with _RAIt
bad_error_eg.cpp:8:41:   instantiated from here
/usr/include/c++/4.5/bits/stl_algo.h:5299:4: Fehler: no match for ≫operator<≪ in ≫__first2.__gnu_cxx::__normal_itera
/usr/include/c++/4.5/bits/stl_algo.h: In Funktion ≫_OIter std::merge(_IIter1, _IIter1, _IIter2, _IIter2, _OIter) [wit
/usr/include/c++/4.5/bits/stl_algo.h:3163:4:   instantiated from ≫void std::__merge_sort_loop(_RandomAccessIterator1,
/usr/include/c++/4.5/bits/stl_algo.h:3263:4:   instantiated from ≫void std::__merge_sort_with_buffer(_RandomAccessIte
/usr/include/c++/4.5/bits/stl_algo.h:3312:4:   instantiated from ≫void std::__stable_sort_adaptive(_RandomAccessItera
/usr/include/c++/4.5/bits/stl_algo.h:5417:2:   instantiated from ≫void std::stable_sort(_RAIter, _RAIter) [with _RAIt
bad_error_eg.cpp:8:41:   instantiated from here
/usr/include/c++/4.5/bits/stl_algo.h:5299:4: Fehler: no match for ≫operator<≪ in ≫* __first2 < * __first1≪
```

1.13 Verbesserte Fehlermeldungen bei Nutzung von StaticAssert

1.13.1 RandomAccessIterator

Zunächst die uneingeschränkt generische Variante:

```
#include <iostream>
#include <vector>
#include <set>
#include <complex>
#include <algorithm>
#include <iterator>
#include <boost/static_assert.hpp>
#include <boost/type_traits.hpp>

template <typename RandomAccessIterator >
RandomAccessIterator foo(RandomAccessIterator from,
                         RandomAccessIterator to)
{
   // this template should only be used with
   // random access iterators...

   //
   // detail goes here...
   reverse(from, to);
   return from;
};
```

```
int main()
{
      std::vector<float> v;
      // std::set<float> v;

      v.insert(v.end(),3.14);
      v.insert(v.end(),15.15);
      foo(v.begin(), v.end());
      copy(v.begin(), v.end(), std::ostream_iterator<float>(std
         ::cout, " "));
      std::cout << std::endl;

}
```

wird einwandfrei übersetzt

```
> make CXXFLAGS="-std=c++0x" ra2b
g++ -std=c++0x -g -I. -I/home/user/include   ra2b.cpp   -o ra2b
```

und funktioniert einwandfrei, für

```
int main()
{
      // std::vector<float> v;
      std::set<float> v;
      //...
```

erscheint jedoch keine vernünftige Fehlermeldung

```
> make CXXFLAGS="-std=c++0x" ra2b
g++ -std=c++0x -g -I. -I/home/buhl/include   ra2b.cpp   -o ra2b
In file included from /usr/include/c++/4.5/bits/char_traits.h:41:0,
                 from /usr/include/c++/4.5/ios:41,
                 from /usr/include/c++/4.5/ostream:40,
                 from /usr/include/c++/4.5/iostream:40,
                 from ra2b.cpp:1:
/usr/include/c++/4.5/bits/stl_algobase.h: In static member function »static void std::__i
/usr/include/c++/4.5/bits/stl_algobase.h:138:7:   instantiated from »void std::iter_swap(
/usr/include/c++/4.5/bits/stl_algo.h:1395:6:   instantiated from »void std::__reverse(_Bi
/usr/include/c++/4.5/bits/stl_algo.h:1441:7:   instantiated from »void std::reverse(_BIte
ra2b.cpp:19:4:   instantiated from »RandomAccessIterator foo(RandomAccessIterator, Random
ra2b.cpp:30:28:   instantiated from here
/usr/include/c++/4.5/bits/stl_algobase.h:89:11: Fehler: Zuweisung der schreibgeschützten
/usr/include/c++/4.5/bits/stl_algobase.h:138:7:   instantiated from »void std::iter_swap(
/usr/include/c++/4.5/bits/stl_algo.h:1395:6:   instantiated from »void std::__reverse(_Bi
/usr/include/c++/4.5/bits/stl_algo.h:1441:7:   instantiated from »void std::reverse(_BIte
```

```
ra2b.cpp:19:4:   instantiated from »RandomAccessIterator foo(RandomAccessIterator
ra2b.cpp:30:28:   instantiated from here
/usr/include/c++/4.5/bits/stl_algobase.h:90:11: Fehler: Zuweisung der schreibgesc
make: *** [ra2b] Fehler 1
```

Durch ein geeignetes statisches Assert der Art

```
#include <iostream>
#include <algorithm>
#include <iterator>
#include <vector>
#include <set>
#include <complex>
#include <boost/static_assert.hpp>
#include <boost/type_traits.hpp>

template <typename RandomAccessIterator >
RandomAccessIterator foo(RandomAccessIterator from,
                         RandomAccessIterator to)
{
   // this template can only be used with
   // random access iterators...
   typedef typename std::iterator_traits<
         RandomAccessIterator >::iterator_category cat;
   static_assert(
      (boost::is_convertible<
         cat,
         const std::random_access_iterator_tag&>::value),
       "no random access iterator");
   //
   // detail goes here...
   reverse(from, to);
   return from;
};

int main()
{
     // std::vector<float> v;
     std::set<float> v;

     v.insert(v.begin(),3.14);
     v.insert(v.begin(),15.15);
     foo(v.begin(), v.end());
     copy(v.begin(), v.end(), std::ostream_iterator<float>(std
        ::cout, " "));
```

```
    std::cout << std::endl;
}
```

wird jedoch der falsche aktuelle generische Parametertyp gemeldet:

```
make CXXFLAGS="-std=c++0x" ra2
g++ -std=c++0x -g -I. -I/home/buhl/include   ra2.cpp   -o ra2
ra2.cpp: In Funktion »RandomAccessIterator foo(RandomAccessIterator, RandomAccessIterator
ra2.cpp:36:28:   instantiated from here
ra2.cpp:18:4: Fehler: statische Behauptung gescheitert: "no random access iterator"
...
```

1.13.2 Nicht instantiierbare Klassen

```
#include <iostream>
#include <algorithm>
#include <boost/static_assert.hpp>
#include <boost/type_traits.hpp>

template <typename T>
struct abstractClass
{
        static_assert(false, "This class may not be
           instantiated!");
        // ...
};

int main()
{
    abstractClass<int> ac;
    std::cout << std::endl;
}
```

1.13.3 Erzwingung gleicher Typen

```
#include <limits>
#include <boost/type_traits.hpp>
#include <boost/static_assert.hpp>

template <class UnsignedInt>
class myclass
{
private:
  static_assert(boost::is_same<UnsignedInt, unsigned int>::
     value,
```

```
                "UnsignedInt isn't unsigned int");
public:
   /* details here */
};

 //myclass<unsigned> m1;
 //myclass<int> m2;
 //myclass<unsigned char> m3;
 myclass<unsigned long> m4;

int main()
{
 return 0;
}
```

1.13.4 Funktionen mit (int/float/...) type promotion

```
#include "promote.h"
template <typename T1, typename T2>
   typename promote_trait<T1,T2>::T_promote
        my_function(T1 x, T2 y)
{
    return  (x + y)/2.0;
}
```

mit `promote.h` ähnlich wie:

```
template <typename T1, typename T2>
struct promote_trait{
    typedef T1 T_promote;
};

template<> struct promote_trait<char, char> {
public:
        typedef int T_promote;
};
//...
```

1.14 Auf Unterklassen eingeschränkte Generizität

```
template <typename ListUnterklasse>
class MyList
{
   static_assert(boost::is_base_of<List,
                                   ListUnterklasse>::value,
                 "ListUnterklasse ist Kindklasse von List");
   // ...
}
```

1.15 Templatefunktionen nur für gewisse generische Argumente: eine Art der Spezialisierung

1.15.1 enable_if-Funktionen

```
#include <iostream>
#include <cmath>
#include <limits>
#include <boost/type_traits.hpp>
#include <boost/utility/enable_if.hpp>

using namespace std;

template <typename ArithmeticLike1, typename ArithmeticLike2>
typename boost::enable_if<boost::is_arithmetic<ArithmeticLike1
   >,
                          double>::type
  geomMittel2(const ArithmeticLike1& a, const ArithmeticLike2&
     b)
{
    return sqrt(abs(a*b));
}

// uebersetze mit -std=c++0x
// oder g++ CXXFLAGS="-std=c++0x" ...

int main()
{
    cout << geomMittel2(3.0, 300.0) << endl;
    cout << geomMittel2(3, 300.0) << endl;
    cout << geomMittel2(-3, 300.0) << endl;
    cout << geomMittel2(-3, 300) << endl;
```

```
    cout << geomMittel2(3.0, 'c') << endl;
    // cout << geomMittel2(3.0, "c") << endl;

    return 0;
}
```

1.15.2 enable_if-Mehrdeutigkeiten

3.2 Overlapping enabler conditions

1.15.3 template class specializations

3.1 Enabling template class specializations

1.16 Ausblick: C++2x eventuell mit eingeschränkter Generizität: Concepts

http://www.generic-programming.org/languages/conceptcpp/tutorial/

```
template<std::CopyConstructible T>
  requires Addable<T>
  T sum(T array[], int n)
  {
    T result = 0;
    for (int i = 0; i < n; ++i)
      result = result + array[i];
    return result;
  }
```

nur für Klassen T mit:

```
auto concept CopyConstructible<typename T> {
  T::T(T const&);
  T::~T();
};

auto concept Addable<typename T, typename U = T> {
  typename result_type;
  result_type operator+(T, U);
};
```

Zu weiteren Concepts vergleiche: http://www.generic-programming.org/languages/conceptcpp/concept_web.p
Eine praktische Anwendung:

```
#include <iostream>
#include <cmath>
#include <vector>
#include <concepts>

using namespace std;

auto concept HasAbs<typename T> {
    typename result_type;
    result_type abs(const T&);
}
auto concept HasPower<typename T>{
    typename result_type;
    result_type pow(const T&, int);
}
auto concept HasPowerd<typename T>{
    requires FloatingPointLike<T>;
    double pow(const T&, const T&);
}

template <int p = 2, InputIterator InputIter, FloatingPointLike T>
requires True<p >= 1>,
    HasAbs<InputIter::value_type>,
    HasPowerd<T>,
    HasPower<HasAbs<InputIter::value_type>::result_type>,
    HasPlusAssign<T,
         HasPower<HasAbs<InputIter::value_type>::result_type>::result_type>
    T pNorm(InputIter first, InputIter last, T init)
{
    for (; first != last; first++)
    {
        init += pow(abs(*first), p);
    };
    return pow((init), (1.0/p));
}
int main()
{
    vector<double> TD (2);
    TD[0] = 200.0;
    TD[1] = 0.0;

    double res = pNorm<3>(TD.begin(), TD.end(), 0.0f);
    cout << res << " sizeof: "
         << sizeof(pNorm(TD.begin(), TD.end(), 0.0f))
```

```
        << endl;

    double TestData[] = {110.0, 10.0, 10.0};
    cout << pNorm(TestData, TestData + 3, 0.0)
         << " sizeof: " << sizeof(pNorm(TestData, TestData + 3, 0.0))
         << endl;

           double TestData2[] = {10.0, 10.0, 10.0};
           cout << pNorm<1>(TestData2, TestData2 + 3, 0.0l)
                << " sizeof: " << sizeof(pNorm<1>(TestData2, TestData2 + 3, 0.0l)
         << endl;


    return 0;
}
```

Link zu conceptg++: http://www.generic-programming.org/software/ConceptGCC/download.php

Zitat: „ConceptC++ makes programming with C++ templates easier, because the compiler can type-check templates when they are defined, so mistakes show up earlier. Real support for Generic Programming also means that many of the template tricks that are needed in standard C++ are no longer necessary, and, yes, it provides much-improved error messages than we get with C++ compilers today.“

1.17 SFINAE

SFINAE als Voraussetzung für `enable_if`

1.18 Assoziierte Typen

Associated Types

Ein Beispiel:

```
template <typename T1, typename T2>
struct promote_trait{
    typedef T1 T_promote;
};
```

value_type, difference_type, ... in iterator_traits

1.19 Checking Concepts without Concepts in C++

static_assert, enable_if, pod-Overloading for copy():

```
template<typename T>
void copy(T const* source,T* dest,unsigned count)
{
    static_assert(std::is_pod<T>::value,"T must be a POD");
    memcpy(dest,source,count*sizeof(T));
}

// ... oder besser:

template<typename T>
typename std::enable_if<std::is_pod<T>::value,void>::type
copy(T const* source,T* dest,unsigned count)
{
    memcpy(dest,source,count*sizeof(T));
}

template<typename T>
typename std::enable_if<!std::is_pod<T>::value,void>::type
copy(T const* source,T* dest,unsigned count)
{
    for(unsigned i=0;i<count;++i)
    {
        *dest++=*source++;
    }
}
```

1.20 POD-Typen

POD Types

1.21 Boost Concept Check Library

Using Concept Checks:

```
#include <boost/concept_check.hpp>

template <class T>
void generic_library_function(T x)
{
  BOOST_CONCEPT_ASSERT((EqualityComparable<T>));
  // ...
};

template <class It>
class generic_library_class
{
  BOOST_CONCEPT_ASSERT((RandomAccessIterator<It>));
  // ...
};
//... BOOST_CONCEPT_ASSERT()
//    only for things seeable in the
//    template function definition
```

oder

```
#include <boost/concept/requires.hpp>
#include <boost/concept_check.hpp>

template<typename RanIter>
BOOST_CONCEPT_REQUIRES(
    ((Mutable_RandomAccessIterator<RanIter>))
    ((LessThanComparable<typename Mutable_RandomAccessIterator<
       RanIter>::value_type>)),
    (void)) // return type
    stable_sort(RanIter,RanIter);
//... otherwise
```

1.21.1 In boost vordefinierte (STL-)Konzepte

Zur Referenz: concept_check.hpp

Basic Concept Checking Classes

Zum Unterschied von `Assignable` und `SGIAssignable`:

```
  BOOST_concept(Assignable,(TT))
  {
    BOOST_CONCEPT_USAGE(Assignable) {
#if !defined(_ITERATOR_) // back_insert_iterator broken for VC
   ++ STL
      a = a;                // require assignment operator
#endif
      const_constraints(a);
    }
   private:
    void const_constraints(const TT& b) {
#if !defined(_ITERATOR_) // back_insert_iterator broken for VC
   ++ STL
      a = b;                // const required for argument to
         assignment
#else
      ignore_unused_variable_warning(b);
#endif
    }
   private:
    TT a;
  };
```

im Vergleich zu:

```
  // The SGI STL version of Assignable requires copy
     constructor and operator=
  BOOST_concept(SGIAssignable,(TT))
  {
    BOOST_CONCEPT_USAGE(SGIAssignable) {
      TT b(a);
#if !defined(_ITERATOR_) // back_insert_iterator broken for VC
   ++ STL
      a = a;                // require assignment operator
#endif
      const_constraints(a);
      ignore_unused_variable_warning(b);
    }
   private:
    void const_constraints(const TT& b) {
      TT c(b);
#if !defined(_ITERATOR_) // back_insert_iterator broken for VC
   ++ STL
      a = b;                // const required for argument to
         assignment
```

```
#endif
      ignore_unused_variable_warning(c);
    }
    TT a;
  };
#if (defined _MSC_VER)
# pragma warning( pop )
#endif
```

Iterator Concept Checking Classes

```
  BOOST_concept(OutputIterator,(TT)(ValueT))
    : Assignable<TT>
  {
    BOOST_CONCEPT_USAGE(OutputIterator) {

      ++i;                    // require preincrement operator
      i++;                    // require postincrement operator
      *i++ = t;               // require postincrement and
          assignment
    }
   private:
    TT i, j;
    ValueT t;
  };
```

Function Object Concept Checking Classes

```
  BOOST_concept(UnaryPredicate,(Func)(Arg))
  {
    BOOST_CONCEPT_USAGE(UnaryPredicate) {
      require_boolean_expr(f(arg)); // require operator()
          returning bool
    }
   private:
    Func f;
    Arg arg;
  };
```

Container Concept Checking Classes

```
  BOOST_concept(Container,(C))
    : Assignable<C>
  {
```

```
    typedef typename C::value_type value_type;
    typedef typename C::difference_type difference_type;
    typedef typename C::size_type size_type;
    typedef typename C::const_reference const_reference;
    typedef typename C::const_pointer const_pointer;
    typedef typename C::const_iterator const_iterator;

      BOOST_CONCEPT_USAGE(Container)
      {
          BOOST_CONCEPT_ASSERT((InputIterator<const_iterator>))
              ;
          const_constraints(c);
      }

    private:
      void const_constraints(const C& cc) {
          i = cc.begin();
          i = cc.end();
          n = cc.size();
          n = cc.max_size();
          b = cc.empty();
      }
      C c;
      bool b;
      const_iterator i;
      size_type n;
  };
```

Concepts: Usage Pattern vs. pseudo-signature
ConceptC++ Publications

1.21.2 Creating Concept Checking Classes

Example

```
template <class X>
struct InputIterator
  : Assignable<X>, EqualityComparable<X>
{
 private:
    typedef std::iterator_traits<X> t;
 public:
    typedef typename t::value_type value_type;
    typedef typename t::difference_type difference_type;
    typedef typename t::reference reference;
    typedef typename t::pointer pointer;
    typedef typename t::iterator_category iterator_category;

    BOOST_CONCEPT_ASSERT((SignedInteger<difference_type>));
    BOOST_CONCEPT_ASSERT((Convertible<iterator_category, std::
       input_iterator_tag>));

    BOOST_CONCEPT_USAGE(InputIterator)
    {
        X j(i);              // require copy construction
        same_type(*i++,v);   // require postincrement-
           dereference returning value_type
        X& x = ++j;          // require preincrement returning X
           &
    }

 private:
    X i;
    value_type v;

    // Type deduction will fail unless the arguments have the
       same type.
    template <typename T>
    void same_type(T const&, T const&);
};
```

Kochrezept:

- „Concept Checking Template Class“ mit Namen des Konzepts (bitte abgekündigte Namensgebung beachten!) erstellen
- Notwendige Vater-Konzepte mittels Vererbung spezifizieren (hier: `Assignable` und `EqualitiyComparable`)
- Nötige assoziierte Typen mittels `typedef` definieren
- Durch `BOOST_CONCEPT_ASSERT()` nötige Einschränkungen an die assoziierten Typen formulieren
- Mittels `BOOST_CONCEPT_USAGE()` die geforderten Eigenschaften (Operationen) der das Konzept erfüllenden Klassen spezifizieren
- Für `BOOST_CONCEPT_USAGE()` nötige Objektdeklarationen inkl. Funktionsdeklarationen vornehmen

What's the difference between C++0x concepts and The Boost Concept Check Library:

- Compiler brauchen Templates bis zur entgültigen Instantiierung nicht zu übersetzen, also auch nicht syntaktisch zu analysieren. Das Auftreten möglicher Fehlermeldungen ist deshalb bis zur Instantiierung aufgeschoben. Um eine vollständige Testabdeckung zu erreichen, benötigt man ein „Urmuster“ des Gebrauchs aller nach Konzept vorgeschriebenen Operationen, die testcompiliert vorhandene fehlende Operationsdefinitionen aufdecken würde: einen **Archetyp** des Konzepts.

  Die ursprünglich in den C++0x geplanten Konzepte hätten Archtypen automatisch erzeugt und benutzt, somit die Templatedefinition automatisch vollständig typgechecked. Das bisherige C++-template-Handling läßt mögliche in der Dokumentation einer Template-Bibliothek unerwähne Requirements eines generischen Objekts lange unentdeckt und frustriert zu unvorhergesehenen Zeiten dessen Benutzer mit einer bis dahin nie aufgetretenen Fehlermeldung(skaskade): Motivierendes BCCL-Beispiel.

  Bei Benutzung der BCCL hat man eigene Konzepte selbst mit Archetypen auszustatten und diese testzucompilieren (siehe Abschmitt 1.21.3). Die in der BCCL vordefinierten für die STL nötigen Konzepte sind in der Datei `boost/concept_archetype.hpp` mit Archetypen ausgestattet.

Zum Beispiel das Konzept `InputIterator` mit den geforderten Operationen `++i, (void)i++, *i++,*i; Defaultkonstruktor, operator=, operator-> (TrivialIterator); Kopierkonstruktor, swap() (Assignable); operator==, operator!= (EqualtityComparable); Defaultkonstruktor (DefaultConstructible)` (vgl. STL InputIterator) und dem folgenden Archetpyp dafür:

```
//
==========================================================================

// Iterator Archetype Classes

template <class T, int I = 0>
class input_iterator_archetype
{
private:
  typedef input_iterator_archetype self;
public:
  typedef std::input_iterator_tag iterator_category;
  typedef T value_type;
  struct reference {
    operator const value_type&() const { return
       static_object<T>::get(); }
  };
  typedef const T* pointer;
  typedef std::ptrdiff_t difference_type;
  self& operator=(const self&) { return *this;  }
  bool operator==(const self&) const { return true; }
  bool operator!=(const self&) const { return true; }
  reference operator*() const { return reference(); }
  self& operator++() { return *this; }
  self operator++(int) { return *this; }
};
```

- BCCL unterstützt die Überprüfung von semantischen Requirements wie z.B. Benutzbarkeit in Multipass-Algorithmen ... **nicht**!
- BCCL unterstützt das Syntaxremappping (temporäres renaming) **nicht**.
- BCCL unterstützt Kontext-basiertes Überladen **nicht**.

1.21.3 Erstellung eines zugehörigen Archetypes

Concept Covering and Archetypes
archetypes
see: 5 Concept Covering

Ein anderes Beispiel:

```
template <class T>
class random_access_iterator_archetype
{
public:
  typedef random_access_iterator_archetype self;
public:
  typedef std::random_access_iterator_tag iterator_category;
  typedef T value_type;
  typedef const T& reference;
  typedef T* pointer;
  typedef std::ptrdiff_t difference_type;
  random_access_iterator_archetype() { }
  self& operator=(const self&) { return *this;  }
  bool operator==(const self&) const { return true; }
  bool operator!=(const self&) const { return true; }
  reference operator*() const { return static_object<T>::get
     (); }
  self& operator++() { return *this; }
  self operator++(int) { return *this; }
  self& operator--() { return *this; }
  self operator--(int) { return *this; }
  reference operator[](difference_type) const
    { return static_object<T>::get(); }
  self& operator+=(difference_type) { return *this; }
  self& operator-=(difference_type) { return *this; }
  difference_type operator-(const self&) const
    { return difference_type(); }
  self operator+(difference_type) const { return *this; }
  self operator-(difference_type) const { return *this; }
  bool operator<(const self&) const { return true; }
  bool operator<=(const self&) const { return true; }
  bool operator>(const self&) const { return true; }
  bool operator>=(const self&) const { return true; }
};
template <class T>
random_access_iterator_archetype<T>
```

```
operator+(typename random_access_iterator_archetype<T>::
    difference_type ,
          const random_access_iterator_archetype<T>& x)
  { return x; }
```

1.22 Programmieren mit Konzepten

Minimiere die Requirements an die Input-Parameter generischer Komponenten, um deren Wiederverwendbarkeit zu steigern.

1.23 Glossar der generischen Programmierung

Konzept A concept contains a set of requirements that describe a family of abstractions, typically data types. Examples of concepts include InputIterator, Graph, and EqualityComparable.

Requirement/Anforderung A requirement is part of a concept that describes the behavior of an abstraction. Requirements tend to be syntactic (e.g., all InputIterators have a dereference operation), semantic (e.g., one can traverse the sequence of values returned from a ForwardIterator muliple times), or performance-related (e.g., incrementing an InputIterator occurs in constant amortized time).

Modell A model is a type or set of types that meets the requirements of a concept. An integer pointer is a model of the InputIterator concept. „Model“ can also be used as a verb to describe the relationship between a type or set of types and a concept, e.g., an adjacency list models the Graph concept.

Archetyp An archetype class is an exact implementation of the interface associated with a particular concept. The run-time behavior of the archetype class is not important, the functions can be left empty. A simple test program can then be compiled with the archetype classes as the inputs to the component. If the program compiles then one can be sure that the concepts cover the component.

assoziierter Typ An associated type is a type that is used to describe the requirements of a concept, but is not actually a parameter to the concept. For instance, the reference type returned when dereferencing an InputIterator is expressed as an associated type. In languages that do not directly support associated types, type parameters can be used instead at some cost to brevity.

konzeptbasiertes Überladen Concept-based overloading selects the most specific algorithm from a set of specializations of a given algorithm.

1.24 Techniques of generic programming in C++

1. Concepts

   Example Concepts: STL, MTL, Ring, Field, Boost Graph Library

   Needed Concepts: for the datatatypes C-XSC Classes and the (generic) algorithms C-XSC algorithms, examples

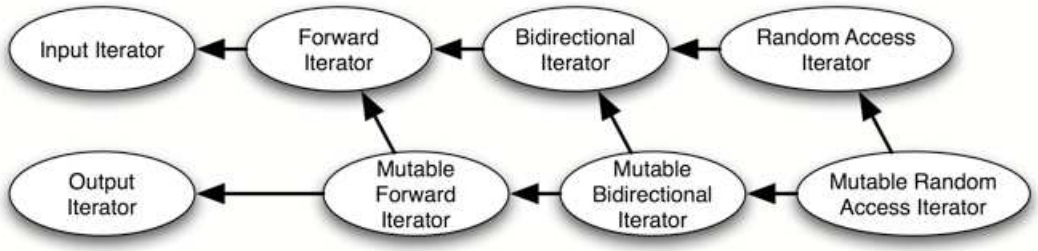
2. Generic Algorithms
3. Type Traits
4. Tag Dispatching
5. Concept-based Overloading
6. Adaptors
7. Concept Checking
8. Archetypes

1.25 Verallgemeinerung von Algorithmen: Lifting

- Iterative finding communality among similar implementations

1. Constructing template version, using default constructor (documenting requirements)
2. Lifting Containers (instead of arrays)
3. Lifting iterators (instead of indexing), avoiding modifying types

- Concepts = organized lists of requirements
- nested requirements
- assoziierte Typen
- hirarchisch verschachtelte Anforderungen
- Beispiel: Concept taxonomy Iterator



1.26 Modelle für Konzepte

http://www.generic-programming.org/about/intro/models.php

1.27 Retroaktive Modellierung (rückwirkend gültig, mit Methoden-Namen-„Umbenennung“ gültig)

http://www.generic-programming.org/about/glossary.php?term=retroactive modeling
http://www.generic-programming.org/faq/?category=paradigms#object-oriented-programming

Temporäres `rename` von Methoden zum Beispiel in Eiffel (hier zur Auflösung von Namenskonflikten bei der Mehrfachvererbung oder zur verbesserten Verbalisierung):

```
class Multiindex inherit
   ARRAY[CARDINAL]rename
                        count as Dimension,
                        clear_all as Null
                     redefine
                        abs,
                        put,
                        make
                     undefine
                        has
                     end;
feature
   abs: CARDINAL;
   put(v:like item; i:INTEGER)
        - - replace i-th entry, if in index interval, by v
        ⋮
end - - class Multiindex
```

Temporäres Umbenennen (map) von Methoden eines Datentyps zur Modellierung von Konzepten in ConceptC++:
http://www.generic-programming.org/languages/conceptcpp/tutorial/#adapting
„Concept maps: show *how* a set of types meets the requirements of a concept.“
(aus: Taming C++ Templates with Concepts)

```
concept DreieckLike<typename D>{
   typename length_type;
   requires std::FloatingPointLike<length_type>;

   length_type getSide(const D&);
   length_type getHeight(const D&);
```

```
    //...
}

concept ParallelogrammLike<typename P>{
    typename length_type;
    requires std::FloatingPointLike<length_type>;

    length_type getSide(const P&);
    length_type getHeight(const P&);
    //...
}

class Dreieck{
    public:
     float a; float b; float c;
     float h_a; float h_b; float h_c;
};

class Parallelogramm{
    public:
      double a; double b;
      double h_a; double h_b;
};

concept_map DreieckLike<Dreieck>{
    typedef float length_type;
    float getSide(const Dreieck& d){ return d.a; };
    float getHeight(const Dreieck& d){ return d.h_a; };
};

concept_map ParallelogrammLike<Parallelogramm>{
    typedef double length_type;
    double getSide(const Parallelogramm& p){ return p.b; };
    double getHeight(const Parallelogramm& p){ return p.h_b; };
};
```

Mixins:
http://en.wikipedia.org/wiki/Mixin
http://stackoverflow.com/questions/505686/are-there-scala-like-mixins-for-c
Mixin based programming in C++
Language Comparison:
A Comparative Study of Language Support for Generic Programming

2 Metaprogrammierung

C++-Metaprogrammierung
Typelists

```
template<class List1, class List2>
struct TypeListAppend
{
    typedef TypeList<typename List1::Head, typename
        TypeListAppend<typename List1::Tail, List2>::Result>
        Result;
};
template<class List2>
struct TypeListAppend<NullType, List2>
{
    typedef List2 Result;
};
// Auf die Implementierung von TypeListBeforePivot und
   TypeListAfterPivot soll hier verzichtet werden
template<class List, template<typename A, typename B> class
   Comparator>
struct TypeListSort
{
    typedef typename TypeListAppend<
                        typename TypeListSort<
                            typename TypeListBeforePivot<
                                typename List::Tail,
                                typename List::Head,
                                Comparator>::Result,
                            Comparator>::Result,
                        TypeList<
                            typename List::Head,
                            typename TypeListSort<
                                typename TypeListAfterPivot<
                                    typename List::Tail,
                                    typename List::Head,
                                    Comparator>::Result,
                                Comparator>::Result
                            >
                        >::Result Result;
};
```

Metaprogramming
Template metaprogramming
Meta Controll Structures (see C++ Template Metaprogramming):

```
// IF

template <bool condition, class Then, class Else>
struct IF
{
    typedef Then RET;
};

template <class Then, class Else>
struct IF<false, Then, Else>
{
    typedef Else RET;
};


// if sizeof(int) < sizeof(long) then use long else use int
IF< sizeof(int)<sizeof(long), long, int>::RET  i;

//...

//                n!
// C(k,n) = ------------
//          k! (n-k)!

template <int k, int n>
struct Combinations
{
    enum { RET = Factorial<n>::RET/(Factorial<k>::RET*Factorial
        <n-k>::RET) };
};

cout << Combinations<2,4>::RET << endl;
```

Expression templates
Template template parameters
Mixins
Traits
C++-Metaprogrammierung(Seite 27: Nachteile der Metaprogrammierung in C++)
The Boost C++ Metaprogramming Library

3 Policy-basiertes Klassendesign

3.1 Policies

Policy based Design

```
template < typename output_policy , typename language_policy >
class HelloWorld : public output_policy , public language_policy
{
    using output_policy::Print;
    using language_policy::Message;

public:
    //behaviour method
    void Run()
    {
        //two policy methods
        Print( Message() );
    }
};

#include <iostream>

class HelloWorld_OutputPolicy_WriteToCout
{
 protected:
    template< typename message_type >
    void Print( message_type message )
    {
        std::cout << message << std::endl;
    }
};

#include <string>

class HelloWorld_LanguagePolicy_English
{
 protected:
    std::string Message()
```

```
    {
        return "Hello, World!";
    }
};

class HelloWorld_LanguagePolicy_German
{
 protected:
    std::string Message()
    {
        return "Hallo Welt!";
    }
};

int main()
{
    /* example 1 */
    typedef HelloWorld<HelloWorld_OutputPolicy_WriteToCout,
       HelloWorld_LanguagePolicy_English> my_hello_world_type;

    my_hello_world_type hello_world;
    hello_world.Run(); // Prints "Hello, World!"

   /* example 2
    * does the same but uses another policy, the language has
       changed
    */
    typedef HelloWorld< HelloWorld_OutputPolicy_WriteToCout,
       HelloWorld_LanguagePolicy_German >
       my_other_hello_world_type;

    my_other_hello_world_type hello_world2;
    hello_world2.Run(); // Prints "Hallo Welt!"
}
```

Policy based design: Überblick
Seite 8: Policies
Generic Pool Design
policy-based class design
Boost Numeric Conversion Library: Policies

3.2 Entwurfsmuster Strategie

C++ Design Pattern: What is a Design Pattern?
Einführung in Design Patterns: 4.4 Das Strategy Pattern
Strategy pattern
The Strategy design motif
Implementing the Strategy Pattern
Design patterns

3.3 Orthogonale Policy-Dimensionen

A Case for Orthogonality in Design
Orthogonality

3.4 Policies (Fortsetzung)

Policy-based Design

3.5 Aspekte komplexer Unternehmensanwendungen

Code Scattering der Wirkungsstellen einzelner Policies/Belange/Anforderungen
AOP = Aspect Oriented Programming
Dynamische Einbau neuer Anforderungen in Geschäftsapplikationen
An Introduction to AOP
I want my AOP!

Aspect, Join Point, Advice, Pointcut
AspectC++
AspectC++ Eclipse Plugin

AspectC++ Quick Reference

Nachteile:
Schlechte Unterstützung beim Debuggen, Profilen, ...
(mögliche) Codeexplosion beim Aspekt-Einweben
Setzt AOP-Begriffe und -Ideologien als Bekannt voraus
Erforder Pattern-Matching-Erfahrungen (Filterdefinition)
Erfordert Recompilation der in der Regel riesigen Unternehmensapplikationen
Probleme der Abhängigkeit von der Reihenfolge des Einwebens(?)
evtl. schlecht lesbarer neu entstehender Code
Sind wirklich alle relevanten Codestellen mit Advices geändert worden? (fehlende

direkte Sprachkonstuckte von C++, z.B. Annotationen mit Aspekt-Bezug, ...)

Vorteile:
Schnell und einfach aufzusetzen
selektiv einsetzbar
keine Modifikation der Originalquellen nötig
leicht entfernbar
gute Performance

Entwicklungsstand der aspektorientierten Programmierung