



"Avslutande" föreläsning

- Nyheter i C++
- Notation & Designdiagram
- Designmönster
- Programverktyg
- Att hitta mer info
- Kritik och framtidens C++

400

Department of Computing Science, Umeå University



Nyheter i C++ "971114"

- Modulär programmering
- Standardbibliotek
- Run-time type info (RTTI)
- Småaker
- Några standardundantag

401

Department of Computing Science, Umeå University



Modulär programmering

- Namespaces

```
#include <iostream>
int main()
{
    std::cout << "Hello, world!\n";
}
//ELLER
#include <iostream>
using namespace std; // Lägg alla id i global
int main()
{
    cout << "Hello, world!\n";
}
```

Utan .h

402

Department of Computing Science, Umeå University



Standardbibliotek

- Run-time; allokering, dyn.datatyper
- C standardbiblioteket
- Strings
- I/O-streams
- Container & algoritmer "STL"
- Numeriska beräkningar

403

Department of Computing Science, Umeå University



Småaker i biblioteket

- generella numeriska gränser
- auto_ptr
 - gör delete när den kommer ur scope

```
fkn()
{
    auto_ptr< int > ip = new int;
    *ip = 37;
} // delete ip görs automatiskt
```

404

Department of Computing Science, Umeå University



Header-filer

- #include <stdio.h> läggs i globalt namespace
- #include <cstdio> med namespace std
 - Det **behöver** inte vara en .h-fil
- <X.h> blir <cX>
- Nuförtiden utan .h för standardbiblioteket

405

Department of Computing Science, Umeå University



Run-time type info (RTTI)

- `dynamic_cast<>()`
 - talar om ifall ett objekt är av önskad typ annars 0
- `typeid()`
 - `cout << typeid(*p).name();` // textsträng klassnamn

406

Department of Computing Science, Umeå University



Några cast-exempel

```
double d = 8.1;
int x = static_cast<int>( d ); //standard conv

void test::print() const {
    const_cast< test* > (this) -> number--;}

unsigned x = reinterpret_cast< unsigned >(charP);

pap = dynamic_cast< provarv* >(pp);
```

407

Department of Computing Science, Umeå University



”Små saker”

- bool datatyp, true=1, false=0
- komplexa tal
- valarray, slices - snabba numeriska rutiner
- Vida tecken; `wchar_t wc= L'ab'` - internationellt
- enum kan bara tilldelas definierade värden
- mutable -nyckelord
 - Kan ändra i const-klasser
- `explicit`
 - tillåter inte implicit konvertering

408

Department of Computing Science, Umeå University



Några standardundantag <stdexcept> ”framework”

- `bad_alloc` ↔ `new`
- `bad_cast` ↔ `dynamic_cast`
- `bad_typeid` ↔ `typeid`
- `bad_exception` ↔ `exception spec`
- `out_of_range` ↔ `at()`, `bitset<> []`
- `invalid_argument` ↔ `bitset constructor`
- `overflow_error` ↔ `bitset::to_ulong()`
- `ios_base::failure` ↔ `ios_base::clear`

409

Department of Computing Science, Umeå University



UML

- Unified Modeling Language
- Three amigos
 - Grady Booch
 - James Rumbaugh
 - Ivar Jacobson
- Gav upp deras egna metoder för att uppnå standardisering

410

Department of Computing Science, Umeå University



Användbara modeller är

- riktiga (accurate)
 - beskriver systemet korrekt
- konsekventa (consistent)
 - inga konflikter
- enkla att förmedla till andra
- enkla att ändra
- förståeliga
 - så enkla som möjligt men inte enklare

411

Department of Computing Science, Umeå University



De största problemen med dagens mjukvaruutveckling

- Många projekt börjar programmera för tidigt
- Läger ner för mycket energi på kodningen
- Ledningen saknar förståelse för mjukvaruutv
- Programmerare känner sig mer säkra när de programmerar än när de bygger abstrakta modeller

412

Department of Computing Science, Umeå University



Målen med UML

- Modellera system (inte bara mjukvaran)
- Koppla ihop koncept
- Hantera skalning
- Skapa ett språk som kan användas av både människor och datorer

413

Department of Computing Science, Umeå University



UMLs framtid

- Kommer att bli det dominerande, gemensamma modelleringspråket i industrin
- Brett användningsområde (inkl RealTid)
- Väletablerad, beprövad teknik
- Väldokumenterat
- Fritt (ej företagsägt)

414

Department of Computing Science, Umeå University



Översikt

- Vyer
 - visar systemet utifrån olika aspekter
- Diagram
 - grafer som beskriver innehållet i en vy
- Modellelement
 - representerar OO koncept såsom klasser, metoder och relationer
- Generella mekanismer för kommentarer etc

415

Department of Computing Science, Umeå University



Vyer

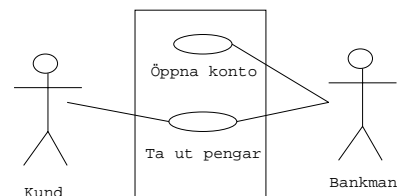
- Användningsfall (use-cases)
 - Systemets funktionalitet som den uppleva av en extern användare (actor)
- Logisk
 - Funktionaliteten inuti; statiska strukturen, dynamiska beteendet
- Komponent
- Parallellitet
- Spridning, Gruppering (deployment)

416

Department of Computing Science, Umeå University



Diagram - Användningsfall

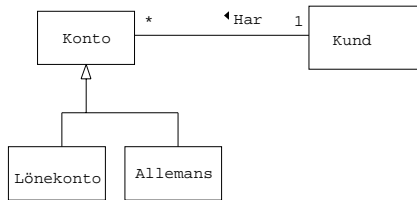


417

Department of Computing Science, Umeå University



Klassdiagram

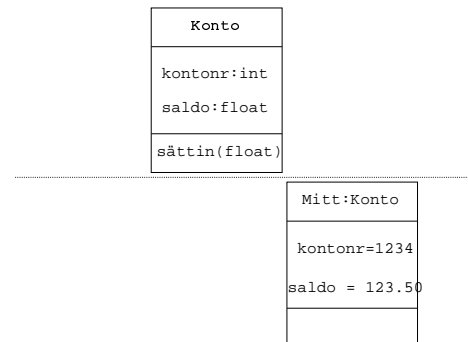


418

Department of Computing Science, Umeå University



Objektdiagram

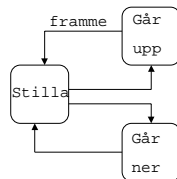


419

Department of Computing Science, Umeå University



Tillståndsdigram

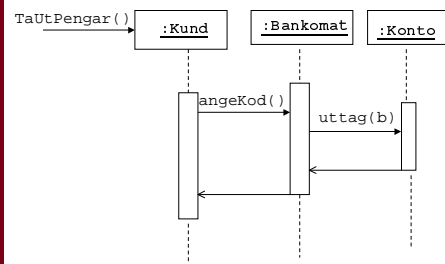


420

Department of Computing Science, Umeå University



Sekvensdiagram

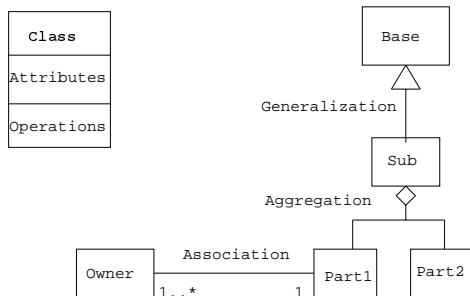


421

Department of Computing Science, Umeå University



Modellelement



422

Department of Computing Science, Umeå University



Designmönster

- Baseras på en bok
 - Design Patterns, Gamma/Helm/Johnson/Vlissides
- Typiska programmonster som dyker upp
- Kan/bör återanvändas
- Ca 25 st i boken
- Ursprungligen från arkitekter
 - Mary Lynn Manns (Patterns)
 - www.cs.unca.edu/~manns

423

Department of Computing Science, Umeå University



Creational Patterns

- Abstract factory
 - Skapar objekt med okänd klass
- Builder
 - Skiljer skapandet från representationen
- Factory method
 - Sparar instantierandet till subklasser
- Prototype
 - Skapar nya objekt från en prototyp
- Singleton
 - Försäkras om enbart ett objekt av viss klass

424

Department of Computing Science, Umeå University



Structural Patterns

- Adapter - Konverterar klassens gränssnitt
- Bridge - Skiljer abstraktion från impl
- Composite - Aggregat som träd
- Decorator - Läger till egenskaper
- Facade - Ger enhetligt gränssnitt
- Flyweight - Hanterar många småobjekt
- Proxy - Surrogat för objekt (accesskontroll)

425

Department of Computing Science, Umeå University



Behavioral Patterns: 1

- Chain of responsibility - Överlämnar objekt
- Command - Kommando som objekt
- Interpreter - Språk ger grammatik&tolk
- Iterator - Accessar element sekventiellt
- Mediator - Kapslar in objektsamarbete
- Memento - Sparar objektets tillstånd

426

Department of Computing Science, Umeå University



Behavioral Patterns: 2

- Observer - Sprider ett objekts ändring till andra
- State - Tillståndsmaskin som byter klass
- Strategy - Kapslar in algoritmer
- Template method - Låter subklassen göra delar av algoritmen
- Visitor - Representerar en operation som ska utföras på en objektstruktur

427

Department of Computing Science, Umeå University



Programverktyg

- Borland Builder
 - En Delphi-miljö för C++
 - Kritikerrosad i USA
- Rational Rose
 - Designverktyg
 - Round-trip engineering
 - C++ / Java / fler
 - Booch/OMT/UML

428

Department of Computing Science, Umeå University



Rational Rose



429

Department of Computing Science, Umeå University



Ett klassdiagram



430

Department of Computing Science, Umeå University



Kritik mot C++

- C++ är inte perfekt
- Är inte bäst till allt
- Stort & komplext språk
- Riskabelt med pekare, indexering, dyn.minne
- Kryptiskt &!~, templates<>
- Ineffektivt för parallellberäkningar
- Varierande implementationer/versioner

431

Department of Computing Science, Umeå University



Bjarne Stroustrups kommentarer

- C++ sucks
 - Använd vad du tror är bättre
 - C++ är mycket användbart; till många applikationer, av många. Avundsjuka
 - Många som hatar C++ har inte provat/lärt sig
- Too complex
 - Många har lärt sig använda/tillämpa C++
 - OOP tröskel (1 år)
 - Det går att skriva oläsligt i alla språk
 - C++ är inte Windows

432

Department of Computing Science, Umeå University



forts

- C++ programmerare är idioter
 - Många (1.000.000?) med blandade nivåer
 - De flesta har inte tid att fördjupa sig
 - Expansivt: Alla är noviser i början
- C++ är som C, fast värre
 - C++ är mer strängt typat
 - Många C/Pascal-programmerare vill inte lära sig
- C++ värdelöst eftersom det saknar feature X
 - Många lyckade projekt (utan X: allt är inte nödv.)
 - Samma personer tycker det är för

433

Department of Computing Science, Umeå University



forts2

- C++ är inte OO
 - Bra support för verkliga applikationer
 - Många borde uppdatera sin kunskap
- Kompilatorerna är buggiga
 - Ja, men de blir bättre (undvik gråa zoner)
 - Många tillbehör; Debuggers, Minneskoll, Browsers
- Standardkommittéen vet inte vad de gör
 - Tro inte allt som står på nätet
 - Hårt arbetande medlemmar (avslår mycket)

434

Department of Computing Science, Umeå University



forts3

- Många C++ katastrofer
 - Språket får skulden men andra orsaker till fel
 - Ex. Denver Bagagehanter (skrivet mest i C/Asm)
 - När många provar icke-triviala nyheter
 - Många fler lyckade projekt
 - » Folk använder C++:s flexibilitet med omdöme
 - » Som all annan ny teknik
- Jag gillar inte C++
 - OK. Det är många saker jag inte gillar.
 - C++ är bäst för många applikationer
 - Skilj tyckande från fakta

435

Department of Computing Science, Umeå University



C++s framtid

- Invänta erfarenheter från bibliotek
- Stabilitet
- Platform ABIs (t.ex Intel IA-64)

● Synpunkter på

- Garbage collection
- Virtual member functions

436

- Nästa revision: 2003

Department of Computing Science, Umeå University



Mer info på Webben

- C++
 - <http://www.research.att.com/~bs/homepage.html>
 - http://nz.com/webnz/robert/cpp_site.html
- UML
 - <http://www.cs.umu.se/tdb/kurser/TDBC18/local/Slides/index.htm> 72-89 (Jürgen Börstler)
 - <http://www.rational.com/uml/index.html>
- Borland
 - <http://www.borland.com/bcppbuilder/>
- Microsoft
 - <http://www.microsoft.com/products/developer.htm>
- Rational Rose designverktyg
 - <http://www.rational.com>

437

Department of Computing Science, Umeå University