Objektorienterade programmeringsspråk

Vad vill vi ha?

Några olika språk

UPPSALA UNIVERSITET

Informationsteknologi

7 steps towards object-based happiness

1. (Object-based modular structure) Systems are modularized on the basis of their data structures.
2. (Data abstraction) Objects should be described as implementations of abstract data types.
3. (Automatic memory management) Unused objects should be deallocated by the underlying language system, without programmer intervention.
4. (Classes) Every non-simple type is a module, and every high-level module is a type.
5. (Inheritance) A class may be defined as an extension or restriction of another.
6. (Polymorphism and dynamic bindings) Program entities should be permitted to refer to objects of more than one class, and operations should be permitted to have different realizations in different classes.
7. (Multiple and repeated inheritance) It should be possible to declare a class as heir to more than one class, and more than once to the same class.

UPPSALA UNIVERSITET

Informationsteknologi

Garbage Collection

- Stor källa till fel i program:
  - Minnesläckage
  - Avreferera felaktiga pekare
- Lösning: automatisk minneshantering. Systemet deallokerar oanvända objekt utan programmerarens inverkan.
- GC minskar den tid och kraft som spenderas på programmering, testning och debugging
- Tidskrävande – ej för realtidssystem (ex spel)

UPPSALA UNIVERSITET

Informationsteknologi

Simula

- en objekt-orienterad utvidgning av Algol 60, ursprungligen avsett för simulering.
- Simulas viktigaste arv till eftervärlden är inte simuleringsprincipen, utan att språket visade värdet av att kunna strukturera objektorienterat
- Introducerade klasser, objekt, arv, dynamisk bindning

Simula: Språket

- Värdetyper (int, real, …) och referenser till objekt
- Skilda operationer för värdetyper och referenser
  - := vs :-, = vs ==, /= vs =/=, etc
- Enkelt arv
- Inkapsling –
  - ursprungligen ingen
  - senare “protected” = osynlig för klienter
- Virtuella metoder
  - Olika implementationer kan ha olika signatur
- Polymorfism
- Statisk bindning – utom för virtuella metoder
- GC

Simula: Klassdefinition

- Parametrar
- Attribut (kan vara klasser)
- Metoder
- Life

Simula - programexempel

```
Class Rectangle (Width, Height); Real Width, Height;
                                  ! Class with two parameters;
 Begin
    Real Area, Perimeter;  ! Attributes;

    Procedure Update;      ! Methods (Can be Virtual);
    Begin
       Area := Width * Height;
       Perimeter := 2*(Width + Height)
    End of Update;

    Boolean Procedure IsSquare;
       IsSquare := Width=Height;

    Update;                ! Life of rectangle started at creation;
    OutText("Rectangle created: ");
    OutFix(Width,2,6);
    OutFix(Height,2,6);
    OutImage
 End of Rectangle;
```

Vad är det för fel på Simula?

- Simula anses vara det första objektorienterade språket men används inte speciellt mycket. Varför?

UPPSALA UNIVERSITET

Informationsteknologi

Vad är det för fel på Simula?

- Generally:
  - Born in a small European country
  - Frozen in 1968
  - Expensive
  - Does not have a modern IDE
  - Too complicated
  - Not enough publications
- Language features:
  - Limited file access facilities (typed files)
  - Missing data types (records, sets)
  - No advanced parallelism and real time support
  - No GUI support
  - Long executable files for short programs
- OOP features:
  - No multiple inheritance
  - No interfaces
- Simulation:
  - No automatic collection of statistics
  - No report generator
  - No useful specialized facilities (resources)

UPPSALA UNIVERSITET

Informationsteknologi

Smalltalk

- Togs fram vid Xerox PARC under 1970- och 1980-talet
- Skapade begreppet objektorienterat
  - Mycket av dagens terminologi
  - (Javas rekommenderade regler för namngivning)
- Avsikten: att bygga ett kraftfullt informationssystem som kunde växa och förändras I takt med användarens idéer och önskemål
- Grunden för många framsteg för datoranvändning, t ex
  - Gränssnittskonstruktion (fönsterhantering, ikoner för val på bildskärm, pop-up menyer)
  - Integrated Development Environmnet (IDE)
  - Plattformsoberoende via användandet av Virtuell Maskin (VM)
  - Just In Time Compiling (JIT) redan 1983
  - Design Patterns
- Används fortfarande inom bla
  - Webteknologi
  - XP

Smalltalk: Språket

- Dynamiskt typat
- Dynamisk bindning
- Allt är objekt – även klasser!
- Enkelt arv – alla klasser ärver
- Attribut dolda, metoder kan exporteras
- Syntax
  - Sex reserverade ord (nil, true, false, self, super, thisContext),
  - Femton specialtecken ([]|{}_^;$#:.-())
  - Tre sätt att skicka meddelanden (unära, binära och nyckelord).

UPPSALA UNIVERSITET

Informationsteknologi

Smalltalk: klassdefinition

```
NameOfSuperclass subclass: #NameOfClass
    instanceVariableNames: 'instVarName1 instVarName2'
    classVariableNames: 'ClassVarName1 ClassVarName2'
    poolDictionaries: ' '
    category: 'name of category'
```

Smalltalk - programexempel

```
Object subclass: #TransientBeing
    instanceVariableNames: 'birthDay '
    classVariableNames: ''
    poolDictionaries: ''
    category: 'SmalltalkIdioms-Being‘

! !TransientBeing methodsFor: 'accessing'!

age
    ^Date today year - self birthDay year!

birthDay
    ^birthDay!
```

C++

- Imperativt programmeringsspråk med stöd för dataabstraktion, objektorienterad programmering och generisk programmering.
- Utvecklades i början på 1980-talet av Bjarne Stroustrup vid Bell Labs.
- Bygger på C med idéer från Algol 68 och Simula
- Är ett av de mest populära programmeringsspråken och används för programmering inom allt från spel till konsumentelektronik.

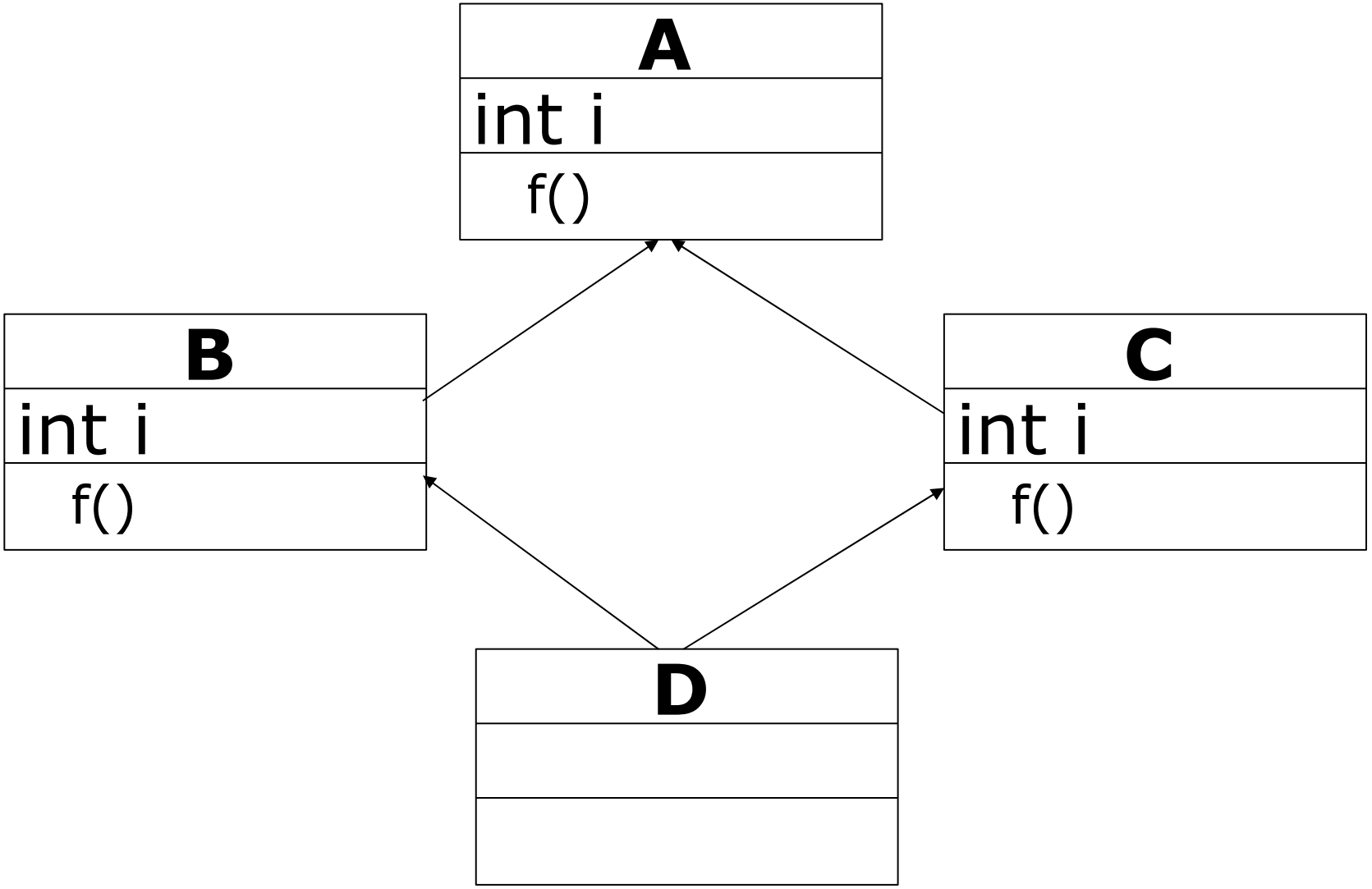
UPPSALA UNIVERSITET

Informationsteknologi

C++: Språket

- Ingen GC
- Explicit call-by-reference
- Överladdade operatorer
- Multipelt arv
- Virtuellt arv
- Templates – generiska funktioner/klasser

The deadly diamond of death



Java

- Hette från början OAK (James Gosling, 1991).
- Utvecklades vid SUN Microsystems.
- Från början avsett för inbyggda system (tex tvättmaskiner, mobiltelefoner, TV/video, spel, mm).
- (Storlek och pålitlighet var viktiga)
- Med WWW fick projektet en annan inriktning
- Java blev ett programspråk för web-applikationer

UPPSALA UNIVERSITET

Informationsteknologi

(Man brukar säga att) Java är...

- enkelt
- “baserat på C/C++”
- objekt-orienterat “från början”
- dynamiskt utbyggbart
- robust, säkert
- oberoende av hårdvaruarkitektur (portabelt)
- web-anpassat: applets, div bibliotek

UPPSALA UNIVERSITET

Informationsteknologi

.NET

- En plattform centrerad kring
  - en Common Language Runtime, CLR (jmf JVM), och
  - en samlig bibliotek
- Kan nyttjas av ett antal språk som kan samköra genom att de kompilerar till ett “intermediate language” (MSIL)

UPPSALA UNIVERSITET

Informationsteknologi

C# (C-sharp)

- utvecklat av Microsoft i samband med skapandet av den nya .NET-plattformen. Standariserat 2001.
- C# är ett (strikt) objektorienterat språk
- bygger på C/C++ med starka influenser av Java.
- Kompileras till MSIL och körs på CLR
- I CLR lever objekten i skyddade miljöer som kallas Application Domains. Kommunikationen mellan objekt i olika Application Domains kan ske väldigt effektivt, i synnerhet om dessa finns i samma process på samma fysiska dator.

C# och Java (≠ C++)

- Kompilerar till maskinoberoende kod som körs i kontrollerad omgivning (abstr. maskin)
- GC tillsammans med borttagande av pekare (I C# finns begränsad användning av pekare i kod som markerats “unsafe”)
- Inga header-filer eller problem med cirkulära beroenden
- Alla klasser ärver från object
- Enkelt arv + multipelt interface arv
- Inga globala funktioner, konstanter

Properties: Getters och Setters

- Java/C++

```
public int getSize() {
   return size;
}
public void setSize (int value) {
   size = value;
}

…

foo.setSize (getSize () + 1);
label.getFont().setBold(true);
```

- C#

```
public int Size {
   get {return size;
   }
   set {size = value;
   }
}

…

foo.size++;
label.font.bold = true;
```

UPPSALA UNIVERSITET

Informationsteknologi

C# - fler nyheter

- Indexers
- Delegates
- Events
- Enum
- Collections + foreach
- Structs
- Primitiva typer som structar
- Operator överladdning
- Virtuella metoder – override
- Interface
- Call-by-reference, “out”-parameter, params
- Switch – även strängar, inget break
- Pekararitmetik i “unsafe”-metoder
- …

C#

- **Finesser:**
  - GC
  - Helt objekt-orienterat, ALLT är objekt - även primitiver
  - Stöder multi-dimensjonella arrayer.
  - Operator överladdning
  - XML kan läggas direkt i en struct istället för i en klass, medför större effektivitet.
  - foreach statement
  - I C# är alla objekt COM objekt.
- **Mindre bra:**
  - Mindre dynamiskt än Java.
  - Använder C++ stil på namn
  - break och continue kan inte ha label.
  - Metoder är inte virtuella default (som i Java)
  - Dålig säkerhet?
  - Ingen template-mekanism

Språk	Simula	Small-talk	Eiffel	C++	Java	C#
GC	**Y**	**Y**	**Y**	**N**	**Y**	**Y**
Poly-morfism	**Y**	**Y**	**Y**	**Y**	**Y**	**Y**
Dynamisk bindning	**Y**	**N***	**Y**	**Y**	**Y**	**Y**
Arv	**Y**	**Y**	**Y**	**Y**	**Y**	**Y**
Multipelt arv	**N**	**N**	**Y**	**Y**	**N****	**N****
Statiskt typat	**Y**	**N**	**Y**	**Y**	**Y**	**Y**

* Dynamisk bindning för virtuella metoder
** Interface ger begränsad form