

EXAMENSARBETE

Våren 2007

Läroarutbildningen

**En kartläggning av
metaforers bruk i
objektorienterad
programmerings-
undervisning**

Författare

Fredrik Kallin

Handledare

Björn Cronqvist

www.hkr.se

En kartläggning av metaforers bruk i objektorienterad programmeringsundervisning

Abstrakt:

Inom undervisning av Objekt-Orienterad Programmering (OOP) blir metaforer ett allt viktigare redskap. En av anledningarna till detta är bland annat metaforers kapacitet till att underlätta undervisning av ämnesrelaterade koncept. Den här rapporten syftar till att synliggöra och analysera lärares bruk av metaforer som redskap och stöd i undervisning av OOP. För att möjliggöra det här har jag genomfört ett antal intervjuer på ett gymnasium i Skåne. Intervjuerna har skett med lärare som undervisar i OOP-relaterade ämnen. Genom litteratur har jag gjort en generell kartläggning av hur metaforer kan agera som stöd och redskap i undervisning. För att sedan få en djupare kunskap i programmeringsundervisning har jag även undersökt alternativa sätt att använda metaforer i OOP. Med den här teoretiska grunden, samt den informationen som hämtats från intervjuerna har sedan en analys genomförts.

Min slutsats och resultatet av undersökningen visar att metaforer används i OOP undervisning. De är, om än av varierande grad, ett stöd för alla de deltagande lärarna. All samlad fakta i rapporten pekar på att det finns klara fördelar med att använda metaforer i undervisningen. De ökar förståelse av komplexa koncept, är motiverande och praktiska.

Ämnesord

Programmering, program, OOP, metaforer, objektorienterad, språk

Innehållsförteckning	Sida
1. Inledning/Introduktion	6
1.1. Ett historiskt perspektiv	6
1.2. Ett skolcentrerat perspektiv	6
1.3. Vad innebär det att programmera	6
1.4. Modern programmering	7
1.5. Vad är OOP	8
1.6. Syfte	9
1.7. Problem	9
2. Teori	10
2.1. En definition och genomgång av begreppet metafor	10
2.2. Vilket stöd ger metaforer lärare i undervisningssituationer	10
2.2.1 Metaforers verkliga magi - den ökade förståelsen	10
2.2.2 Broarna som byggs	11
2.2.3 Den ökade motivationen och den emotionella tonen	11
2.2.4 Metaforer är praktiska	12
2.3. Metaforer i OOP-undervisning	12
2.4. De fem tillvegagångssätten att se programmering på	13
2.4.1 Kategori 1: Följande	13
2.4.2 Kategori 2: Kodning	14
2.4.3 Kategori 3: Förståelse och Integrering	14
2.4.4 Kategori 4: Problem lösning	14
2.4.5 Kategori 5: 'Deltagande'	14
2.5. Metaforer inom andra ämnen	14
3. Metod	16
3.1. Val av metod	16
3.1. Datainsamlingsteknik	16
3.1.1 Djupintervju	16
3.2. Datainsamlings genomgång	17
3.3. Bortfallet	17
3.4. Metoddiskussion	17
4. Resultat	17
5. Analys & Diskussion	22
5.1 Dessa metaforer använder lärare	22
5.2 Användning av metaforer - en medveten, eller en kanske omedveten, handlig	23
5.3 Användning av metaforer i klassrummet	23
5.3.1 Varför	23
5.3.2 Metaforer en deus ex machina	23
5.4 Vanliga metaforer i klassrummet	23
6. Slutsats	24
7. Förslag till vidare forskning	25
8. Sammanfattning	26
Referenslista	27
Elektroniska referenser	27
Billaga 1	

1. Inledning

1.1 Programmering - Ett historiskt perspektiv

Läser man i uppslagsverket *Encyclopedia Britannica* finner man att året 1954 var ett historiskt år ur flera aspekter. Det var året då vi i Sverige införde tetrapack, och TV-test sändes för första gången, Ernest Hemingway fick Nobelpriset i litteratur och Elvis Presley gjorde skivdebut. Året var även det då ett team, i Texas, USA, under John Backus ledning skapade det första moderna programmeringsspråket. 1954 var året då Fortran föddes. Före Fortrans introduktion i programmeringens värld var programmering en omodern, mödosam och omständlig procedur. Något sådant som en gemensam standard för programmering existerade inte och varje dator kom med sitt eget unika programmeringsspråk. Det här gjorde programmering osmidigt då det resulterade i att datorerna var tvungna att programmeras separat. I och med skapandet av Fortran togs steget från det enskilda programmeringsspråket till ett gemensamt språk, ett som kunde användas på mer än en dator. Före Fortran, vid stora gemensamma systemkonfigureringar, var man nämligen tvungen att, i tur och ordning, göra ändringar på var och en av datorerna. 1954, var det här kanske inget större problem, då det antalet datorer världen över, låg på en relativt hanterbar nivå. Idag, år 2007, med det antal datorer som existerar runt om i världen idag, hade det här dock presenterat ett näst intill olösligt dilemma. För att illustrera det här dilemma tar vi hjälp av det operativsystem som de flesta av dagens moderna persondatorer har installerat: Windows XP. Operativsystemet Windows tros vara skrivet med omkring 35 miljoner rader kod (Sample, 2004). År 2002 ansågs antalet persondatorer ligga på runt en biljon i antal, stigande årligen med ca 130 miljoner (Sample, 2004). Utan att göra några exakta uträkningar gör vi en enkel jämförelse; 35 miljoner rader kod kontra 35 miljoner gånger en biljon rader. Låt oss nu ta en sekund och begrunda att det här exemplet endast inkluderade persondatorer. Persondatorer utgör bara en bråkdel av alla de datoriserade system som använder någon form av moderna programmerings språk. Enligt Sample (2004) så använder de flesta av de existerande kontemporära datorsystemen någon form av modernt språk. Utan den standardiseringen som moderna programmeringsspråk inneburit i och med t.ex. operativsystem som Windows hade det sannolikt inte existerat ens en bråkdel av de datorer och datoriserade system som det gör idag.

1.2 Ett skolcentrerat perspektiv

I takt med att samhället blivit allt mer beroende av datorer och därmed även beroende av programmering har även efterfrågan på kompetenta programmerare ökat. På Skolverkets hemsida är programmering idag en del av de flesta nationella gymnasieprogrammen. Medieprogrammet, Teknikprogrammet, Energiprogrammet och Elprogrammet är bara några exempel på inriktningar där programmering är en valbar del av programmet. Allt fler friskolor öppnas över hela landet. På dessa skolor utgör programmering en allt större del av schemat. Programmering är med andra ord på stark frammarsch i skolundervisningen.

1.3 Vad innebär det att programmera

Vad innebär (begreppet) programmering? Den allmänna uppfattningen kan sammanfattas med att det inte finns någon uppfattning. Få klarar nämligen av att definiera begreppet. De som klarar av att ge en begreppsdefinition jämför programmering med att skriva text på en dator. Det här är inte på något sätt en felaktig beskrivning, då en stor del av att arbeta med programmering består av att just skriva text på en dator. Att programmera innebär dock mycket mer. Här följer en kort, mer detaljerad redovisning av begreppet. För att bättre och lättare illustrera innebörden av att programmera låt oss använda oss av metaforen; *ett programmeringsspråk är som ett mänskligt språk*. Som mänskliga språk, så som Svenska och Engelska kan programmeringsspråk beskrivas som bestående av två framstående delar. Det första är ett vokabulär, och det andra är ett set grammatiska regler för att instruera hur

en specifik uppgift ska utföras. I programmeringsspråk består vokabulären av en uppsättning unika nyckelord (ord som människan kan förstå), och de grammatiska reglerna av en speciell syntax. Så här lyder en av *Yahoo.com* definition av ordet syntax "a systematic, orderly arrangement". Syntaxens uppgift är, liksom de grammatiska regler vi använder oss av när vi skriver mänskligt språk för att organisera de instruktioner som ges. Här följer ett jämförande exempel på hur en syntax kan se ut i programmeringsspråk jämfört med människospråk.

Programmeringsspråk = P.
Människospråk = M

Exempel 1 – Korrekt syntax

P: System.out.println("Kalle mår bra idag");
M: Kalle mår bra idag

Låt oss nu ta och ändra på syntaxen, eller de grammatiska reglerna om man så vill i båda exemplens.

Exempel 2 – Felaktig syntax

P: Out.system.println("Kalle mår bra idag");
M: Mår idag bra Kalle

På grund av att de grammatiska reglerna och syntaxerna inte följs i exempel två, uppstår det här komplikationer. I människospråksexemplet blir konstaterandet "Kalle mår bra idag" istället "Mår idag bra kalle", vilket gör att meningen förlorar den innebörd som den ursprungligen hade. Samma sak sker i programmeringsspråksexemplet. Då en människa däremot möjligen skulle kunna lista ut innebörden att sätta meningen i sin kontext eller att ställa en följdfråga hade ett program upphört att fungera. Det är därför av största vikt att bägge språken följer sina grammatiska regler. Att programmera skulle kunna liknas vid att lära sig ett helt nytt sätt att kommunicera med medföljande grammatik och vokabulär.

1.4 Modern programmering – Ett nutida perspektiv

Det föregående kapitlet kan ha gett intrycket av att programmering är en upplevelse som kräver, för att uppskattas, en förkärlek för att skriva rader och åter rader av kod. Enligt Nguyen (2003) så var det här också det allmänna synsättet av programmering under årtionden. Både i programmeringens kretsar och i skolans värld sågs programmering som lite annat än mödosamt skrivande av långa texter. I takt, dock med att programmering utvecklats så har även sättet att se på programmering gjort det. Den textcentrerade innebörd som tidigare definierade begreppet programmering har i och med skapandet av de objektorienterade språken omstrukturerats till en centrering mot den bakomliggande logiken. Samtidigt som utvecklingen av programmering och sättet att se på programmering i och med OOP gått framåt berättar Nguyen (2003) att undervisningen av ämnet bromsats in. Det här trots att användandet av objektorienterade former av programmeringsspråk inom undervisning numera är standard i de flesta gymnasieinriktningar. Undervisningsmetoder ute i många av skolorna är fortfarande relativt förlegade och en stor del av fokuset i utläring ligger fortfarande inte på att lära elever programmeringens bakomliggande logik. Eleverna får lära sig att programmering endast är en speciell text som skrivs på en dator för att uppnå ett visst resultat. Det här missgynnar dem eftersom det inte utvecklar elevens sätt att se på programmering. Elever behöver lära sig att se på programmering som något mer än skrivande av text på en dator. En utveckling av elevens sätt att se på programmering behöver ske och det är här metaforer kommer in i bilden (Nguyen, 2003).

Om man går djupare in på ämnet objektorienterad programmering (OOP) märker man

snabbt att det är ett ämne fyllt av metaforik. Trots detta har jag under min praktik observerat att metaforers roll i undervisningen av programmering inte är så stor som den kanske borde vara. Det här har konfunderat mig då det, som kommer att påvisas i arbetets teoridel, kan vara ett starkt stöd. Metaforer är nämligen ett bra sätt att träna upp elevers mentala maskinerier och öka förståelsen för den bakomliggande logiken i avancerad programmeringsundervisning.

Ortoney (1993) betonar vikten av att människans mentala maskineri tränas då det har en viktig roll för hjärnans förmåga att förstå och tolka information. Till sin hjälp att förstå och tolka har det mentala maskineriet, till sitt förfogande, ett eget språk. Språket är det som hjärnan använder för att bygga ihop komplexa tankar genom kombination av enkla tankar. Det är delvis så här metaforer skapas, en metafor är ett exempel på en form av komplex tanke som skapats med hjälp av två enklare. Utan hjärnans inre språk hade det inte varit möjligt att vare sig skapa eller förstå metaforer. Metaforer som används för att undervisa OOP består exempelvis två tankar: en tanke som handlar om den kod som skall förklaras, och en tanke som handlar om det objekt som skall skapa förståelse för koden. Skapandet av komplexa tankarna som metaforer är det som hjälper hjärnan att träna sitt mentala maskineri.

Låt oss nu gå djupare in på begreppet Objektorienterad programmering.

1.5 Vad är OOP

Enligt Clark (2002) är OOP utan tvekan en av de mest komplexa programmeringstekniker att förklara. Faktum är att det inte är så mycket en teknik som man ska förklara utan helt nytt sätt att se på programmering och hur program är strukturerade.

Före OOP existerade det egentligen inte något lätt sätt att se på programmering. De program som skapades saknade någon som helst praktisk struktur, och bestod av långa listor av instruktioner för datorn att genomföra. När programmet sedan startades gick listan igenom från första till sista instruktion. Den här enkla strukturen medförde tre stora problem. Det första att programmen helt upphörde att fungera om inte alla instruktioner inkluderade på listan fungerade, ett enda enkelt fel kunde stoppa hela programmet. Det andra att programmen snabbt blev astronomiskt stora. Ett enklare program kunde bestå av allt från tio till hundratals sidor kod. Det tredje var att de båda nyss nämnda problemen gjorde felsökning i programmet till en mödosam procedur. Dessa problem var delvis det som föranledde skapandet av OOP, och med OOP kom ett nytt sätt att se på program och dess struktur.

Enligt Clark (2002) är människans mentala process för förståelse, logik och medvetande av naturen väldigt objektorienterad. Det faktum att våra hjärnor tolkar och behandlar många av de föremål vi kommer i kontakt med som objekt har varit grundläggande i utvecklandet av OOP. Den här vanan att behandla och se det vi stöter på som objekt skulle göra programmering användarvänligare, lättare att utveckla med, samt leda till lättare förståelse av komplexa situationer och procedurer (Clark, 2002).

Clark (2002) berättar att då program tidigare bestod av ett enda stort objekt bygger OOP på samlingar av mindre självständiga objekt som i samverkan utför en uppgift. Var och en av dessa självständiga objekt är helt autonoma och har förmågan att ta emot, behandla och skicka data. Eftersom objekten är autonoma och därför inte beroende av varandra kan delar av programmet nu sluta fungera, eller tas bort utan att resten av programmet upphör att fungera. Låt oss underhålla tanken att människokroppen är ett enda stort program skrivet med OOP, och uppbyggt av en mängd olika objekt, så som armar, ögon och ben. Då alla de här objekten är självständiga så är de också ej nödvändiga för att programmet skall

fungera, människan skall överleva. Går en arm t.ex. förlorad så överlever ändå människan. Hade programmet varit skriven med traditionell programmering så hade dock alla objekten varit livsviktig för programmet. Om en arm eller ett öga gått förlorad hade programmet slutat fungera, människan hade avlidit.

För att enkelt illustrera hur OOP fungerar låt oss åter titta på den mänskliga kroppen som ett program skapat med OOP. Detta program ber vi utföra en uppgift och det är att kasta en boll i en hink som är placerad en meter framför kroppen. För att uppnå programmets mål behöver vi använda oss av tre objekt i vårt program, armen, hjärnan och ögat. Processen börjar med att ögat tar emot visuell information om hur långt avståndet är till hinken. Informationen behandlas och skickas sedan vidare till hjärnan. När hjärnan tagit emot informationen behandlar den i sin tur information, en bedömning görs om hur mycket kraft och i vilken vinkel bollen bör kastas för att träffa hinken. Informationen skickas nu vidare till armen. Efter att armen har mottagit informationen och behandlat den, positionerar sig sedan armen och kastar bollen efter de specifikationer som mottagits. Det som är beskrivet ovan är en illustration av objekt som samverkar, det här är OOP.

I uppsatsen kommer jag att benämna två exempel på OOP språk; Java, C++. Dessa är förmodligen de två vanligaste språken som nämns när OOP diskuteras. Båda språken är helt baserade på ett objektorienterat tänkande och jämförbara i sin utformning. Det som skiljer dem åt, om vi skall göra en grov indelning, är att Java används till utveckling av Internet baserade applikationer, medan C++ används till all övrig utveckling av programvaror.

1.6 Syfte

Syftet med den här studien är att kartlägga hur metaforer brukas som redskap och stöd i OOP - undervisning.

1.7 Problem

Med min undersökning vill jag kartlägga lärares bruk av metaforer som redskap och stöd i sin undervisning i OOP. Frågeställningarna för undersökningen är följande:

- 1 Använder lärare metaforer i klassrummet. Är det en medveten handling?
- 2 Varför använder lärare metaforer?
- 3 Hur används metaforerna i ett klassrum?
- 4 Vilka metaforer är det som används i klassrumssituationer?

2. Teori

Det här kapitlet kommer att inledda med att utveckla och diskutera en definition och beskrivning av begreppet metafor. Kapitlet går sedan vidare till en genomgång om varför metaforer är viktiga i undervisning, följt av en genomgång av metaforer i icke-programmerings relaterade ämnen. Kapitlet avrundas sedan med en genomgång av hur metaforer används i objektorienterad undervisning.

2.1 En definition och genomgång av begreppet metafor

Enligt Goatly (1997) och Ortony (1993) finns metaforer överallt i vår vardag och i vårt språk. Även fast vi kanske inte tänker på dem så använder vi oss av metaforer dagligen, det kan vara att vi läser dem i en tidning, använder dem när vi pratar eller tänker på dem. Metaforer är en stor del av vår kultur. Metaforer och analogier används även idag både medvetet eller omedvetet i de flesta huvudämnena på gymnasiet. Ekonomi och handel, biologi, kemi, fysik och litteratur är bara några av de ämnen där de används (Ortony, 1993). Något enkelt svar på hur man definierar begreppet existerar inte då begreppet är väldigt mångfacetterat. Enligt Goatly (1997) har metaforer ett flertal dimensioner, de kan vara döda, aktiva, mixade, sammansatta m.m. Då dessa dimensioner av metaforer är av en underordnad betydelse i detta arbete kommer jag att avgränsa min definition. Jag kommer ej i arbetet att gå djupare in på de olika typerna. Hur lyder definitionen av begreppet metafor som används i det här arbetet? Innan jag utvecklar den här definitionen låt oss först undersöka hur Goatly (1997) definierar begreppet. För att underlätta förståelsen av definitionen har jag gjort en översättning från det engelska originalet.

"En metafor äger rum när ett samtal om en enhet används till att referera okonventionellt till ett objekt, process eller koncept, eller parallell på ett okonventionellt sätt. Och när den här okonventionella referensakten eller parallell är förstådd på en basis av likhet, matchande eller analogi involverande den konventionellas referens eller parallell av enheten" (Goatly 1997, s. 33)

Här följer arbetets tolkning av den här definitionen. En metafor är ett ord eller en fras som, för att förstå likhet, refererar till en sak som det bokstavligen inte betyder. En annan infallsvinkel är att metaforer identifierar ett objekt eller en idé med en annan, i en eller flera aspekter. Metaforer expanderar sinnet och assisterar i att klargöra innebörden. "Internet är en informationsmotorväg", går det ju t.ex. att påstå. Förhoppningsvis tänker lyssnaren inte när han hör det här att, "Internet är motorväg" utan hellre "Internet transporterar information (a) fram och tillbaka, (b) i hög fart (c) har mycket trafik".

2.2 Vilket stöd ger metaforer lärare i undervisningssituationer?

Detta är en fråga med ett mångfacetterat svar. Det finns många olika sätt som metaforer kan stödja lärare i deras undervisning. Under det här kapitlets gång ska jag dock försöka att redovisa de vanligaste svaren på den här, för arbetet, centrala frågan: Vilket stöd ger metaforer lärare i undervisningssituationer? I de underrubriker som följer kommer jag bland annat att gå igenom de tre vanligaste fördelarna med metaforer, vilka är att de kan skapa och öka förståelse för komplexa ämnen, att de kan agera motiverare och att de är praktiska.

2.2.1 Metaforers verkliga magi - den ökade förståelsen

Studerar man skrifter och artiklar om metaforer, skrivna av bland annat Goatly (1997), Ortony (1993), Thang N. Nguyen (2003) och George Lakoff, Mark Johnssons (1980) så lyfter de alla fram metaforers kapacitet att öka människans förståelse som den främsta. Hur är då detta möjligt? Goatly (1997) och Ortony (1993) teoriserar båda två om att den här förhöjda

förståelsen möjliggörs genom att metaforer har kapaciteten att hjälpa till med strukturella ändringar i människans mentala maskineri.

För att delvis demonstrera den här ändringen i det mentala maskineriet använder vi oss av den enkla, illustrativa metaforen - Idéer är mat för tanken. Metaforen är delvis grundad vid tanken att idéer är objekt som vi kan hämta utifrån, här sker delvis en strukturell ändring, hjärnan förvandlar något abstrakt till något konkret. En koppling etableras mellan hjärnan och kroppen genom att både hjärnan och kroppen kan identifieras som förvaringslådor som vi kan placera saker i. Liksom mat blir metaforer nu ett objekt som kan placeras i ett kärl, som är hjärnan i det här fallet, en metaforiskt skapad likhet mellan idéer och mat har nu uppstått. Genom att den här basen nu är skapad så öppnas en helt ny värld upp för nya metaforer. Metaforen skapar metaforer. Nu kan idéer som mat helt plötsligt sväljas, slukas och smältas. Låter t.ex. en idé allt för oresonlig kan jag kanske inte svälja utan förkastar den. Hade den dock varit resonerlig hade jag kanske i stället svält den utan att tänka. Det här hjälper hjärnan och tankeverksamheten att förstå den psykologiska processen som den tidigare inte haft något direkt eller väl definierat sätt att conceptualisera. För att den här struktureringen skall vara möjlig och eleven skall kunna koppla måste metaforerna som eleven förses med vara både familjära och i kontext.

Trots att metaforer används i de flesta former av undervisning, är dess förekomst, enligt Ortony (1993), mest frekvent i undervisning där koncepten som används är abstrakta nog för att rättfärdiga att de används. Det här är på grund av att abstrakt kunskap för de flesta människor är en ofamiljär form av koncept att lära sig. Den är avlägsen elevernas vardag och därför svåra att ta till sig. Det är därför eleverna förväntas ta till sig den här formen av abstrakta koncept först när de når gymnasie och högskole nivå. Abstrakt, icke-visuell kunskap kan vara kunskap som eleverna inte direkt kan uppleva med sinnena, det kan även vara erfarenheter som de ännu inte upplevt (Ortony, 1993).

2.2.2 Broarna som byggs

För att skapa förståelse för dessa abstrakta, ej erfarna kunskaperna pratar Ortony (1993) och Goatly (1997) om hur man, med metaforers hjälp, kan *bygga* broar. De här broarna *byggs* mellan det abstrakta, ofamiljära, avlägsna och den mottagande eleven. Genom att både läraren och studenten besitter gemensamma kunskaper och erfarenheter så tillåts läraren dra från dessa kunskaper till byggandet av en intellektuell bro. Båda två vet t.ex. vad en bil är, och hur en sådan fungerar, med hjälp av den här delade kunskapen kan läraren lättare förklara hur OOP är uppbyggt. Brons uppgift är att brygga gapet mellan den kunskapen läraren besitter och det eleven saknar. Här demonstreras den verkliga styrkan med metaforer, skapandet av broar, med hjälp av något båda känner till, mellan studenten och läraren (Ortony, 1998).

2.2.3 Den ökade motivationen och den emotionella tonen

Ortony (1993) har noterat att metaforer har en kapacitet att främja lusten och skapa intresse hos studenter för avancerade, ofamiljära och abstrakta koncept. I vissa fall kan även det, i metaforer, existera en så kallad emotionell ton. Använder du dig som lärare av metaforer som involverar specifika ämnen som står eleverna nära hjärtat så kan en emotionell koppling mellan metaforen och eleven uppstå. Eleven har ett stort personligt intresse av ämnet och vill därför lära sig mer om det. I fall där eleven upplever ämnen som irrelevanta kan denna ton vara nödvändig. Ortony(1993) nämner att Best (1984), DiGiovanna (1987), och Marshall (1984) alla, i rapporter de gjort, framhäver vikten av denna emotionella ton.

2.2.4 Metaforer är praktiska

Enligt Nguyen (2003) är metaforer på flera sätt väldigt praktiska. En positiv egenskap som

ofta anses så pass självklar att många inte tänker på den. Metaforer kan hjälpa lärare att spara både tid och ansträngning. Det uppstår nämligen tillfällen och undervisningssituationer då möjligheten finns att vara ordagran men då det kanske inte är praktiskt. Ett sådant exempel kan vara tillfällen då man har långa, teoretiskt tunga lektioner. Här kan det vara praktiskt att använda sig av metaforer för att korta ner dessa lektioner. Ortony (1993) påpekar att metaforer ej bör missbrukas, och att det är lätt att man gör det. Åtskillnaden mellan metaforiskt och ordagrant språk är inte alltid så svart och vitt. De ordagranna förklaringarna bör inte underskattas då dessa ofta är minst lika potenta redskap som metaforer. Ortony (1993) är noga med att betona att om det finns ett ordagrant sätt att få fram en poäng så bör det sättet användas. Metaforer bör t.ex. inte användas för att överföra karaktäristiska drag, hur distinkta de än är, då de lättare kan förklaras med en enkel mening.

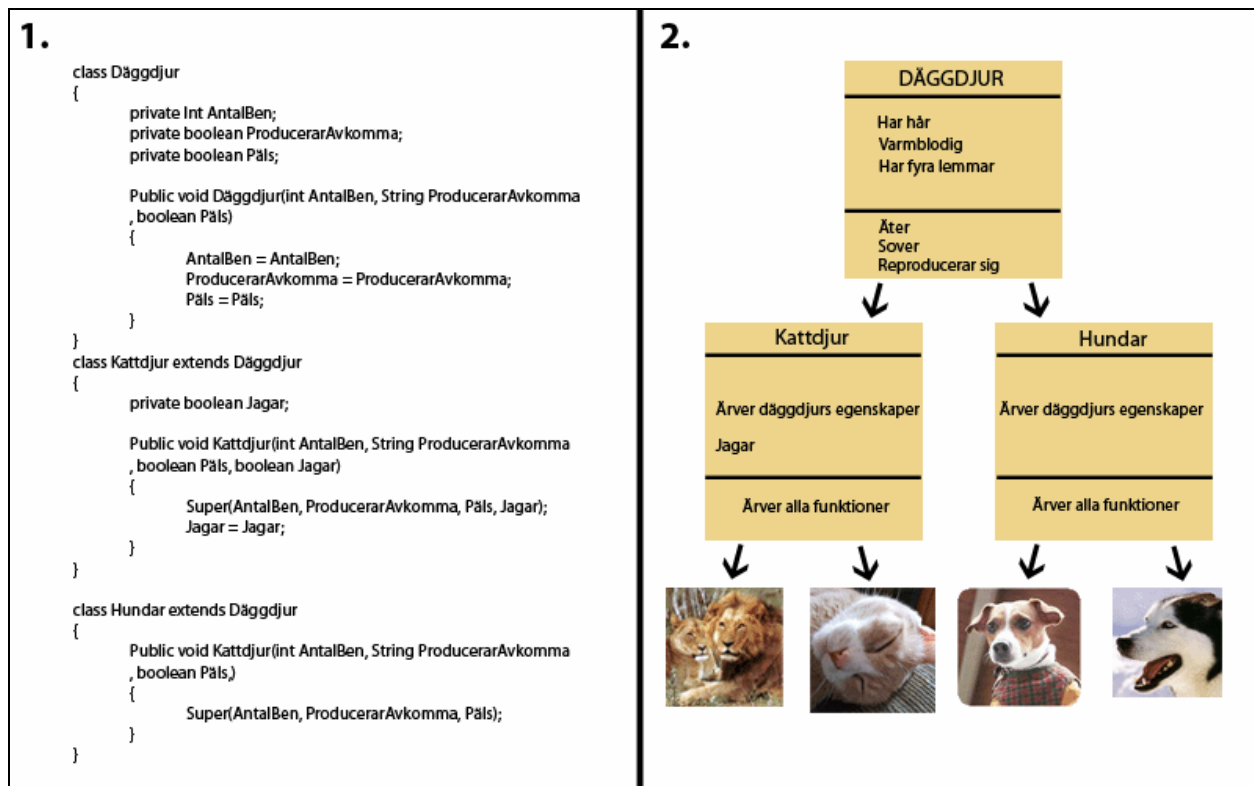
2.3 Metaforer i OOP-undervisning

Som berörts tidigare under kapitel 2.2.1 är kanske det största skälet till användningen av metaforer i undervisning deras förmåga att förstärka förklaring och förståelsen av abstrakta koncept. Tack vare den här egenskapen kommer metaforer väl till användning i undervisning av OOP.

Vid diskussion av undervisningen av OOP kan man enligt Thang (2003) inte undgå att komma in på programmeringsformens fyra mest grundläggande koncept: abstrakthet, inkapsling, arv och polymorfism. Enligt Thang (2003) är de här fyra koncepten överlag de svåraste för en nybörjare i OOP att ta till sig, och de för verkliga undervisningen av OOP till sin spets. Det är nämligen en stor utmaning för undervisande lärare att förmedla de här abstrakta koncepten tillräckligt. På grund av komplexiteten av dessa koncept och att deras innebörd inte har någon egentlig relevans för arbetet kommer jag inte ge någon mer ingående beskrivning av dem.

Låt oss använda oss av det nyss nämnda OOP konceptet – arv - för att demonstrera dilemmat, och hur man med metaforers hjälp kan förmedla förståelse. Arv som koncept bygger på att när en viss objektkategori skapas t.ex. däggdjur eller bilar så kan de underkategorier som skapas från de här klasserna ärva de egenskaper och beteenden huvudklassen har. Detta illustreras i två olika former i Figur 1. Figur 1.1 illustrerar hur arv ser ut skrivet i Java kod, 1.2 innehåller samma kod fast här illustrerat med hjälp av metaforer. Till objektkategorin däggdjur har det skapats två underkategorier vilka innehar namnen Kattdjur och Hundar. Eftersom både kattdjur och hundar är av arten däggdjur får detta ses som logiskt. Dessa båda underkategorier ärver alla de egenskaper huvudklassen däggdjur har. I OOP så är arv en egenskap som representerar en **är en** eller **är ett** relation mellan olika klasser. Till exempel kattdjur och hunddjur **är ett** däggdjur.

Lärarens roll bör vara att få sina elever att öppna ögonen och se allt i omvärlden som objekt. Det är här metaforer träder in i handlingen. Du är en människa, en katt är ett kattdjur, ett datorspel är ett spel. Låt oss säga att bilmärket Porsche är en klass och att fordonskategorin bil är en klass. En Porsche är en bil, därför har Porsche alla de egenskaper alla andra bilar har, dörrar, en kraftkälla, något som driver den framåt. Porsche har här ärvt de här egenskaperna av klassen bilar. Skulle tillverkarna av Porsche skapa ett nytt märke av Porsche skulle den i sin tur ärva av klassen Porsche och bil. Nedanför så visas hur det kan se ut. Figur 1 visar hur en kodbit i OOP kan se ut, i det här fallet så visar den fyra objekt som ärver av varandra.



Figur 1: Grafisk representation av två sätt att se programmerings kod på

2.4 De fem tillvägångssätten att se på programmering.

Utöver de fördelar som alla välutvecklade metaforer potentiellt ger så som ökad förståelse och en motiverande effekt så kan metaforer i OOP potentiellt dra nytta av en extra. Användandet av metaforer i OOP kan även potentiellt hjälpa eleverna att utveckla ett högre och mer sofistikerat sätt att lära sig att programmera på (Bruce, Buckingham, et al 2004). Enligt Bruce, Buckingham et al (2004) har vi alla ett sätt som vi ser programmering på. Har vi ett utvecklat sätt att se på programmering begränsar det oss i vår inläring av detsamma. De flesta elever som börjar programmera upplever programmering som att man skriver en rad syntaxer som magiskt sedan utför en uppgift. Enligt Bruce, Buckingham et al (2004) får elever som inte nått ett högre inläringstillstånd oftast mindre lyckade resultat i att ta till sig programmering. En brasklapp läggs dock in att bara för du är i en specifik kategori så är det ingen garanti för att du nödvändigtvis får bättre resultat t.ex. på en tentamen. Som lärare måste du anpassa dina metoder för att se till att eleverna exponeras till detta sätt att se på programmering. För att skapa kategori 3-eleverna behöver vi basera de övningar vi skapar på element från kategori 3. Endast då kan vi föra eleverna från den nivån av tänkande de är på till en högre nivå av tänkande. Det är här metaforer kan vara användbara (Bruce, Buckingham, et al 2004). Nedanför följer en kort redovisning av de fem tillvägångssätt för en elev att lära sig programmera.

2.4.1 Kategori 1: Följande

Här är elevernas huvudsakliga mål att klara av alla de uppgifter hon/han fått på avsatt tid och huvudsakligt fokus ligger där poäng kan tas. Ämnet programmering jämförs med ett sätt att ta poäng. Strukturen och vilket sätt kursens material presenterats på är därför av stor vikt för eleven. Då eleven har en egen intern bild av dessa kriterier kan eleven uppvisa tecken på frustration om de inte matchar. Ständig feedback krävs även då eleverna har ett beroende av att känna att de är på rätt spår.

2.4.2 Kategori 2: Kodning

I den här kategorin jämföras akten att lära sig programmera med att lära sig koda. Programmeringsspråkets syntax ligger här i fokus och eleverna tror att de behöver kunna den för att programmera. Eleverna kan lätt känna tidspress och frustration då det är en ansevärd mängd syntax som behövs både tränas och läras för att klara av kursen.

2.4.3 Kategori 3: Förståelse och Integrering

I den tredje kategorin ses programmering som ett sätt att lära sig programmera genom integrering och förståelse av de koncept eleven lärt sig. I stället för att lära delarna för sig strävar eleverna att få en helhetsbild. Syntaxen är inte av stor vikt utan de vill utveckla djupare förståelse för att bättre kunna påverka resultatet.

Studenter som lär sig på det här sättet fokuserar både på att förstå uppgiften och för att få förståelse för koncept. Det huvudsakliga fokuset ligger här på att få förståelse för de koncept som ligger till grund för att lära sig att programmera. Den här förståelsen får de genom att göra uppgifter, både de som ingår i kursen och de som de själva sökt upp. Programmering här är att det går upp ett ljus för dem.

2.4.4 Kategori 4: Problemlösning

Här upplevs att lära sig programmera som att göra vad det än krävs för att lösa problemet. Utgångspunkten är ett problem och eleven har som mål att upptäcka vad det innebär att lösa det. Även i den här kategorin så är utvecklingen av en helhetsbild en fundamental del i att lära sig programmera. Programmering och programmet är nämligen bara en del i en större kontext.

Att koda används som en del av inlärningsprocessen men den är bara en del av kontexten att lösa ett problem. Det läggs stor vikt på planering innan du börjar med själva skapandet. Motivationen är själva problemet eleverna skall försöka lösa. Programmera är bara ett redskap för att skapa lösningen på ett problem, lösa en uppgift.

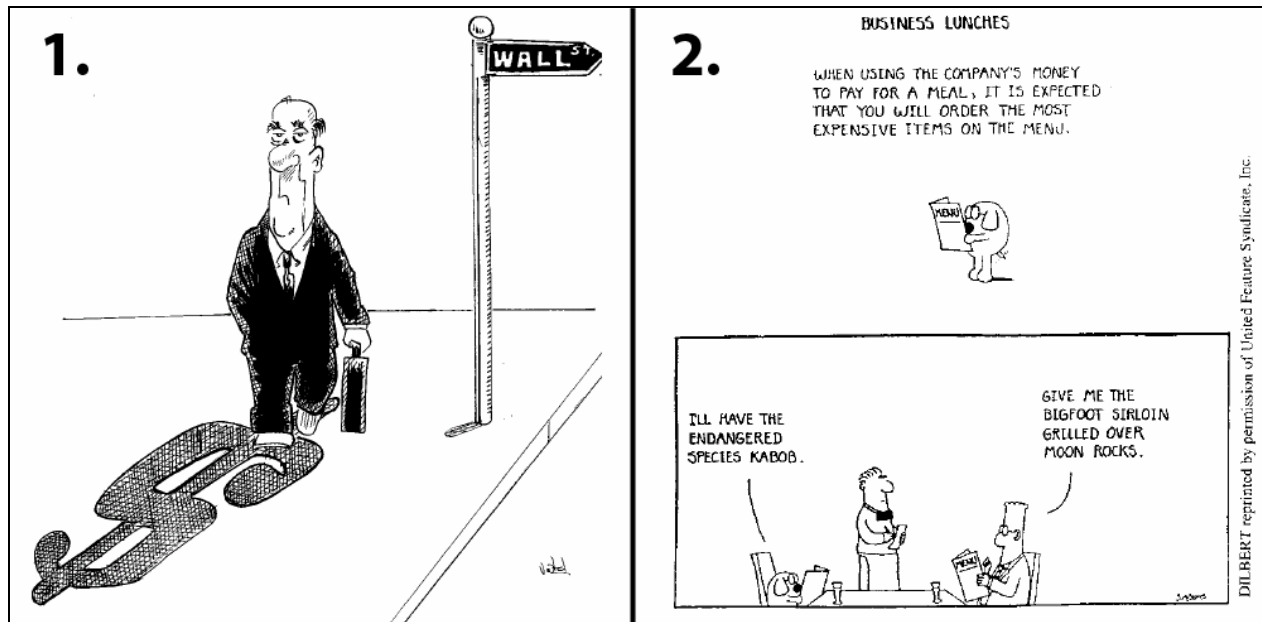
2.4.5 Kategori 5: Deltagande

Här upplevs att lära sig programmera som att lära sig vad som krävs för att bli en del av ett programmeringssamhälle och kulturen. Programmering ses som en kultur, att lära sig programmera är att lära sig kulturen. Att förstå vad det betyder att lära sig att programmera överskuggar hur programmeraren tänker samt vad programmeraren faktiskt gör.

Studenten tittar bortom själva inlärningsituationen, språket och programmen för att förstå vad det innefattar att bli en programmerare.

2.5 Metaforer inom andra ämnen

Så som det nämndes under rubriken "introduktion" så används metaforer i undervisningen av en myriad av huvudämnen, handel, biologi, kemi, fysik och litteratur för att nämna några (Ortony, 1993). Metaforer används bland annat till nybörjare på introduktionskurser i ekonomi. Ardalan (1998) diskuterar om metaforens roll som stöd i att göra ämnet mer attraktivt och roligare. Ett sätt som nämns för att lärare ska kunna åstadkomma detta, menar Ardalan (1998), är genom användandet av underhållande metaforer i sina ekonomikurser. Dessa metaforer konkretiseras genom att illustreras som underhållande, välgenomtänkta bilder.



Figur 2: Underhållande, illustrationer av metaforer (Ardalan, 1998)

I de flesta nybörjarkurser i ekonomi så framställs att öka företags ekonomiska tillväxt som den enskilt viktigaste faktorn för en Financial manager i ett företag. Företaget och dess aktieägare skall alltid komma före Financial managers egna intressen. Några garantier finns dock aldrig för att personen lyckas, ibland så sätter Financial managers sina egenintressen före företagets.

Metaforen i mitt första exempel tar formen av att Financial managers skugga ersatts av formen av ett dollartecken. Det här är tänkt att det ska symbolisera att företagets viktigaste uppgift är att maximera aktievärdet åt aktieägarna. Skuggan symboliserar att var Financial managers än går så krävs det resultat. Den andra metaforen demonstrerar med en stor portion komik hur det kan se ut när Financial managern satt sitt egna bästa före företagets. Att köpa in det dyraste på menyn vid företags luncher får väl nämligen inte anses ligga i företagets bästa.

I båda exemplen så illustreras metaforerna med viss humor. Hypotesen är att nöje och humor ska spela en stor roll som motivatorn i undervisning. Vi återkommer här till det som togs upp under rubriken "Vad bidrar de till med i undervisning" om metaforers roll som en viktig motiverare. Ardalan (1998) teoriserar om hur upphetsande humor kan vara i undervisningssituationer och det faktum att den har en positiv effekt på inläring och minnet. Ett roligare ämne ökar förhoppningsvis även antalet repetitioner eleverna läser av det material de fått. Vad anses då metaforers roll vara i den här processen?

Jo, Ardalan betonar att i stället för att fungera som en grundläggande modell eller grund för det som skall läras, t.ex. inflation, så agerar metaforer, i det här fallet i formen av en rolig bild, som ett stöd som förstärker bilden av inflation. Bilden hjälper sedan eleven att minnas innehållet bättre än vad t.ex. en text skulle göra. Ardalan spekulerar också lätt om att inläringen förstärks genom att informationen kan föreställas av eleverna, vilket leder till en konkretisering för dem.

3. Metod

Det här kapitlet kommer att inledas med en kort beskrivning av min undersökningspopulation och den urvalsprocess jag valt. Vidare går den över till mitt val av metod, samt vilken datainsamlingsteknik jag använt mig av. Under underrubriken djupintervju går jag sedan djupare in på mitt val av datainsamlingsteknik, följt av en genomgång av datainsamlingsprocedur, och eventuella bortfall. Kapitlet avslutas med en kort diskussion om metoden.

Skolan där jag genomfört min undersökning är en gymnasieskola i Skåne. Populationen som undersökningen omfattade bestod av lärare som undervisar i något OOP språk. Förutom att OOP är standardformen av programmering i gymnasieskolans programmeringskurser ingår även programmeringsformen som ett delmoment i form av Director och Flash programmering i multimedia kurserna Multimedia A och B. Inget urval, mer än att alla deltagarna undervisar i OOP, har gjorts av populationen.

3.1 Val av metod

I min undersökning ville jag mer djupgående studera hur metaforer brukas som redskap och stöd i OOP-undervisning. För att få en teoretisk grund att stå på för att göra det här möjligt sökte jag i Halvorsen (1992) och Patel & Davidson (1994) forskningsmetodik böcker. När jag skulle göra mitt val av metod kunde jag, hur jag än vred och vände, inte utesluta vare sig den ena eller den andra metoden. Till slut föll dock mitt val på en kombination av både kvalitativa och kvantitativa metoder. De båda typerna av metod tillförde mitt arbete olika kvalitéer och kompletterar varandra bra. Till insamlingen av empirin valde jag, efter noggrant övervägande, till slut att använda mig av de kvalitativa metoderna. Möjligheterna de gav till löpande analyser, och överlägsen frihet vid presentationen av frågorna var det som vägde över i de kvalitativa metodernas favör. Patel och Davidsson belyser det närmare:

”Fördelen med att göra en löpande analys, t ex direkt efter en intervju eller ett observationspass är att det kan ge idéer om hur vi skall gå vidare. Kanske inser vi att vi i intervjun förbisett något eller att den/de intervjuade uppfattade frågor på ett sätt som vi kanske inte själva tänkt på.” (Patel & Davidsson, 1998, s.101)

De kvantitativa metodernas bidrag till undersökningen dyker först upp i resultatdelen. Här kom de väl tillpass i form av illustrativ statistik för att förstärka förståelsen för dem som vill ta del av resultatet.

3.1 Datainsamlingsteknik

På grund av den tidsbegränsning som det här projektet var under bestämde jag mig tidigt för att använda de intervjuade lärarna som mina informatörer för att uppnå resultat. Halvarsson (2003) beskriver det i sin bok. Till skillnad från exempelvis en enkätundersökning medför de muntliga, kvalitativa metoderna den stora fördelen nämligen möjligheten till flexibla presentationer. Det vill säga möjligheten att både variera sättet och ordningsföljden som frågor ställs till respondenterna, något som inte är möjligt med en enkätundersökning. Eftersom ingen respondent är den andre lik kan det uppstå situationer där det kan finnas ett behov för mig att ändra på ordningsföljden av mina frågor eller till och med omformulering för att underlätta förståelsen av någon fråga (Halvarsson, 2003).

3.1.1 Djupintervjuer

För att lättast kunna uppnå mitt mål bestämde jag mig för att använda mig av muntliga, halvstrukturerade, djupintervjuer (intensivintervjuer). Kunskapen sitter djupt rotat i huvudet på de deltagande lärarna, för mig gäller det att hjälpa de intervjuade att inse att de vet mer

än vad de är medvetna om. Jag ska verka som en katalysator och trigga de intervjuade med rätt frågor – och vilka dessa frågor är kan inte fastställas i förväg – de föds i stunden i den givna situationen. Det här är något som en enkätundersökning kanske inte har kapaciteten till. Hade jag inte haft den personliga kontakt jag har med de deltagande i intervjun hade jag nog dock varit tvungen till att välja en annan metod. Krav för att den här typen av intervju skall bli optimal är nämligen ett förtroligt förhållande mellan intervjuaren och deltagaren. Hade jag varit en okänd person som de aldrig pratat med tidigare hade de kanske inte öppnat upp sig så som de gjorde, vilket hade gjort det otroligt svårt att få fram fakta. Då metoden även, enligt Halvarsson (2003), skapar en ökad förståelse om deltagarens motiv och behov var den verkligen som skapt för min undersökning.

Vid genomförelse av intervjuer finns det alltid en risk att någon av respondenterna oavsiktligt kommer in på en annan fråga samtidigt som diskussionen pågår som intensivast. Här känns det inte alltid rätt att abrupt avbryta respondenten. Det är viktigt för mig att respondenten inte tvingas in i bestämda tankesätt som kan leda till känslan av att vara låst. Det här öppnar upp för fördjupande och öppna svar vilket i sig leder till djupare kunskap och förståelse. Av de här skälen valde jag att göra intervjuerna halvstrukturerade. Det här ökar på flexibiliteten på min undersökning.

3.2 Datainsamlings genomgång

Allt började med att jag förberedde mig väl inför observationstillfället.

Vägen till intervjuerna började med att jag tog kontakt med en lärare på den skolan jag gjorde min undersökning på. Läraren kände jag sedan tidigare då jag handletts av individen under en kort period av Verksamhets Förlagd Utbildning (VFU) tidigare under årets gång. Den här inledande kontakten följde jag upp med ett introduktionsbrev i form av ett e-mail. Väl ute på skolan gick jag igenom reglerna för intervjun, att de var anonyma. Därefter tog jag ut dem till ett grupprum där vi påbörjade intervjun.

3.3 Bortfallet

Bortfallet i undersökningsgruppen blev litet, endast en lärare. Anledningen till detta var att personen fick förhinder för intervjun de tidpunkter jag var och besökte skolan. Då bortfallet var så begränsat som det var hade det en marginell relevans för resultatet av min undersökning.

3.4 Metoddiskussion

I retrospektiv visade sig de flexibla djupintervjuerna vara ett bra val för mig vid min datainsamling. De öppnade verkligen upp för en djup dialog mellan mig och de jag intervjuade. Det faktum att det var halvstrukturerad frågeställning visade sig också vara ett lyckokast. Läs mer om det i resultatkapitlet. Utan att kunna säga med säkerhet så tror jag ett allt för strukturerat sätt att göra intervjun på hade hämmat undersökningen. Under tiden som undersökningen fortlöpte så märktes det hur min teoretiska fördjupning hjälpte mig med datainsamlingen. Genom den teoretiska förankring jag erskaffat mig under skapandet av min teoretiska del kunde jag utveckla intervjuerna från en monolog till en dialog. Ett exempel på detta var när jag bad en av lärarna att ge ett exempel på någon metafor hon använder sig av. Till en början så verkade det näst intill omöjligt för läraren att svara på frågan, men efter en kort dialog där vi allmänt diskuterade metaforer och dess fördelar kom läraren till slut fram till ett svar. Det faktum att jag hade ett bra förhållande till deltagarna hjälpte till att öppna för den här övergången från monolog till dialog. Hade jag inte den kontakten med dem jag hade är det inte alls säkert jag gått tillväga på samma sätt. Mina teoretiska förkunskaper öppnade även upp för de löpande tolkningar och analyser jag gjorde under intervjuerna. Allt det här skapade bra förutsättningar för det resultat jag i

efterföljande kapitel kommer att redovisa och diskutera.

4. Resultat

Inledningsvis kan jag börja med att konstatera att de deltagande i min undersökning alla aktivt använde sig av metaforer i sin undervisning. Dock kan det nämnas att det skedde i en varierande grad. Datalärsamlingen gav mycket information om lärarna och hur de ser på metaforer i sin egen undervisning. Det här kom dock inte som någon överraskning då jag kunde luta mig tillbaka på en öppen och bra relation med merparten av de jag intervjuade. Självklart fanns en risk för tillgjorda svar, av diverse anledningar, under intervjuerna. De deltagande kändes dock genuint intresserade, alla var öppna, och delade mer än gärna med sig av både åsikter och erfarenheter.

I den möjlighet som frågorna tillåter kommer resultatet presenteras först med en enkel tabell som gör en översikt av det som kom fram i intervjun. Efter det gör jag en sammanfattning av resultatet som inkluderar citat och enkel statistik för att underlätta förståelse. Flertalet citat kommer att användas i resultatsammanfattningen då jag anser att lärarnas egna ord ger bäst svar på tal.

Av de sex deltagarna i undersökningen innehar tre tjänsten som medielärare medan de resterande tre undervisar i dataämnen. Datalärarna är de som undervisar i ämnen där OOP är huvudmomentet i de kurser de undervisar i. I medielärarnas undervisning är OOP dock endast ett delmoment.

För att få en så homogen grupp som möjligt har jag försökt få en bra fördelning mellan man och kvinna, och datalärare och medielärare. Följande tabell är skapad för att illustrerar den här fördelningen mellan man och kvinna, samt datalärare och medielärare.

Tabell 1: Fördelningen

	Kvinna	Man
Datalärare	2/6	1/6
Medielärare	0/6	3/6

* = de frågor som används vid intervjuerna

**1. Så vad tycker du om min ide att undersöka metaforers användning i OOP-undervisning?*

Tabell 2: Inställning

Positivt	Negativt
6/6	0/6

Den här inledande frågans främsta syfte var att ge mig en mjuk ingång in i diskussionen/intervjun. Förutom att få den deltagande att lätta på tungan så fungerade den även som barometer för vilken typ av svar man kunde tänkas få i de efterföljande frågorna. Det här tyckte jag var av vikt då det gav mig en möjlighet att anpassa hur jag ställde frågorna till den intervjuade. Frågan uppfyllde väl sina båda syften.

I en jämförelse mellan de två olika sorterna lärare framstod medielärarna som klart verbalare än datalärarna. Medan medielärarna ibland svävade ut i sina redogörelser var datalärarna allt som oftast korta, koncisa och rakt på sak i sina svar. Av de tillfrågade var alla lärarna väldigt positiva i sina bedömningar av intervjuämnet. En av medielärarna

svarade så här:

Medielärare

"Det tycker jag är en bra idé. Det händer ju att man använder sig av metaforer för att koppla till elevers erfarenheter. De har inga erfarenheter av det man ska undervisa i och det är därför man är där då försöker man plocka något de känner igen"

Redan här visade vissa av deltagarna tendenser på att sväva ut och beröra undersökningens övriga frågor. Metaforers effekt att skapa förståelse var bland annat ett av de ämnen som berördes i dessa utsvävningar. Diskussionen spårade dock aldrig ur och jag tillät därför det till viss del. Att skära av dem här tror jag nämligen hade hämmat min intervju. Flexibiliteten i min datainsamlingsteknik kom här väl till användning.

**2. Hur länge har du undervisat i, någon form av, OOP?*

Spännvidden på längden de undervisat, någon form av OOP, varierar kraftigt mellan 1 till 10 år. Av nedstående diagram framgår det hur spritt det var.

Tabell 3: Antal aktiva år som OOP lärare

År	1	2	3	4	5	6	7	8	9	10
	1/6		2/6		1/6		1/6			1/6

Alla, utom en av de tillfrågade lärarna, hade ett år eller mer av erfarenhet inom undervisning av OOP relaterade ämnen. Det bör vägas in då det kan ge en bild av skillnaden mellan hur lärare, som undervisat under en längre tid, undervisar i ämnet kontra de som är nya.

**3. Hur skulle du definiera begreppet metafor?*

Alla deltagarna var säkra i sitt svar och kunde, utan större problem, definiera begreppet. Här följer ett exempel:

Medielärare

"Metafor är ett uttryck för att förklara något som är lite abstrakt på ett lite mer verklighetsnära sätt. Drar en parallell till ett annat skeende som är mer greppbart."

**4. Har du något exempel på någon metafor du använt i dina lektioner?*

Tabell 4: Finns det exemplar

Ja	Nej
6/6	0/6

Fråga 4 verkade vara en svårare fråga att besvara, speciellt för de som undervisar i medieämnen, än de tre tidigare frågorna. Här märktes en skillnad mellan de båda inriktningarna.

Medan de tre datalärarna direkt gav positiva svar, fick två av tre medielärare däremot fundera innan de kom fram till sina svar. I ett av fallen så utbröt en dialog mellan mig och den intervjuade. Dialogen lede till att läraren till slut kom fram till ett svar. 5 av 6 gav dock människo - eller fordons metaforer som man använder för att förklara grundkonceptet med OOP som exempel.

Tabell 5: Vanligaste exemplen

Människan & fordonet	Annan metafor
5/6	1/6

Tystnad och osäkerhet genomsyrade de båda medielärarna medan de funderade på frågan. Efter ett kort meningsutbyte uppdagades det att dessa två lärare inte undervisat i ämnet på ett tag och därför kände sig ringrostiga. Här visade flexibilitet av min datainsamlingsmetod sig användbar. Vi kom överens att de skulle få fundera på frågan medan vi gick igenom och svarade på de resterande frågorna. De båda uttryckte en viss lättnad, samt verkade tacksamma att det gick att ordna. Så här svarade dock den tredje medieläraren:

Medielärare

“Vin och vattenglas har jag använt av. En så kallad behållarmetafor för beskrivning av variabler. Det har nämligen en hel del elever svårt att förstå – data är ju data – och det är ju det men så måste man ha något som beskriver vad innehållet är. Innehållet är vätska och man fyller vattenglasen med den”

Efter jag ställt mina resterande frågor återgick jag sedan tillbaka till den här frågan och fick mina svar.

**5. Har du någon favorit/er som du medvetet eller omedvetet använder dig av?*

Ja	Nej
3/6	3/6

Även här genomsyrades medielärarnas svar av viss osäkerhet. I datalärarnas svar kunde man dock följa en röd tråd. Alla favoriserar nämligen metaforer som brukas för att förklara själva grundkonceptet med OOP så som exempelvis *människan*. Det vill säga hur program skapade i OOP är uppbyggda av autonoma delar som samverkar med varandra. Det här är kanske inte förvånansvärt då det allt som oftast är det som lärare i OOP trycker på. Det här skulle kunna tolkas som att de helt sonika inte hade någon favorit utan nämnde en av standardmetaforerna i deras undervisning. De exempel på metaforer som gavs var människan och fordonet. Här följer ett exempel:

Datalärare

“Människan tycker jag som sagt var är användbar. Alla kan relatera till den och den får eleverna att tänka i andra banor som de kanske inte är vana vid.”

**6. Finns det någon speciell metafor du känner skulle kunna hjälpa elever att förstå OOP? (du behöver inte nödvändigtvis ha använt den själv)*

Tabell 7: Har lärarna speciella metaforer för att förklara OOP

Ja	Nej
3/6	3/6

Datalärare

“Fordons - och människometaforer använder jag ofta, speciellt fordonsmetaforer.”

I 3 av mina 6 intervjuer var jag här tvungen att ändra på min fråga för att förtydliga den. Det blev nämligen här uppenbart att deltagarna hade problem att se skillnad mellan fråga 5 och fråga 6.

Fråga 6 verkade dock för de resterande 3 lärarna vara en relativt lätt fråga att svara på. De 3 nämnde alla "människan" i sina svar.

**7. Har det hänt att du har lagt ner tid på planering av metaforer?*

Tabell 8: Planerar lärarna sina metaforer

Ja	Nej
2/6	4/6

Fyra av de sex intervjuade angav här att de inte planerar sina metaforer, utan de uppstår spontant. De två resterande angav att de lagt ner tid på planerade, dock så var det för att de ingick i lektionsplaneringar. Följande är ett exempel på hur det kunde låta:

Medielärare

"För att jag känner att det kan hjälpa att motivera eleverna och att de ökar medvetandet om den här formen av programmering. Det ger mig även möjligheten att göra ett annars väldigt abstrakt ämne konkret genom användandet av en kombination av bilder och text eller endera."

Den av datalärarna som arbetat längst med undervisning av programmering anmärkte dock att han planerat metaforer i början av sin yrkeskarriär, men att det inte skett nu på flera år. Två av tre medielärares svar kunde nästan tolkas som att de blandat ihop de metaforerna de som använde för programmering undervisning, och de för sina övriga ämnen. En av de två lärarna svarade så här när jag personen frågan:

Medielärare

"När man förklara strukturer och så är det viktigt att man gör det utifrån ett konkret exempel. Annars går det lätt över huvudet för eleverna. Då kan man konstruera en saga i huvudet medan man förklarar."

Inga planeringstillfällen specifikt i syftet att skapa hade ägt rum hos dessa två.

**8. Har det hänt att du skapat metaforer spontant, i ögonblickets stund?*

Tabell 9: Skapar lärarna spontana metaforer

Ja	Nej
6/6	0/6

Som berördes i fråga 7 så uppstod de flesta av metaforerna i deltagarna som skapade i ögonblickets stund.

**9. Varför använder du dig av metaforer?*

Här under har jag sammanställt en enkel tabell med de fyra svaren som angavs på frågan. Alla deltagande gav dock mångfacetterade svar vilket leder till att tabellen innehåller fler markeringar än antal deltagare.

Tabell 10: Skäl till att lärarna använder metaforer

Ökar förståelse	Motivera	Praktiska	Öppnar för individanpassning
6/6	1/6	2/6	2/6

En röd tråd kunde urskönjas genom alla intervjuerna och det var att alla sex nämnde ökad förståelse som den absolut främsta anledningen till brukandet av metaforer.

Även fast de flesta angav multipla skäl så höll alla det här skälet som den främsta. Två av intervjuerna föll även in på vikten av att koppla undervisningen till elevernas erfarenhetsvärld.

Datalärare

"Helt klart för den ökade förståelsens skull."

Medielärare

"Det händer ju att man använder sig av metaforer för att koppla till elevers erfarenheter. De har inga erfarenheter av det man ska undervisa i och det är därför man är där då försöker man plocka något de känner igen"

I det här citatet berörs även en stor del av abstrakthet. Abstraktion nämndes av fyra av sex deltagande som det stora hindret i undervisning av programmering. Här är ett exempel:

Datalärare

"Att ämnet är så pass abstrakt som det är är helt klart ett bekymmer när man skall undervisa nybörjare."

Följande är ett exempel på hur en av lärarna resonerade kring användandet av metaforer för att öka förståelse:

Medielärare

"Handlar om att få medieeleverna att utveckla sitt abstrakta tänkande, vilket är svårt eftersom det här tänkandet hos eleverna oftast saknas, kan jag då kanske exemplifiera med metaforer så kan kanske jag få in dem på rätt tankeespår."

Under flertalet av intervjuerna återkopplas det ett flertal gånger till metaforers roll i att öka förståelse. Följande är ytterligare ett exempel på hur det kan låta:

Medielärare

"Eleverna verkar ha svårt att ta till sig abstrakta koncept som variabler. Kan de inte se något konkret på skärmen så har de otroligt svårt att ta det till sig. Jag försöker därför med metaforernas hjälp skapa något påtagligt för eleverna att se på."

Metaforers roll som motivator togs förvånande nog endast upp ordagrant i en av intervjuerna. En möjlighet är att det var underförstått. Hade det funnits mer tid och man gett möjligheten till. Så här lät det:

Datalärare

"Jag känner att det (Metaforer) kan hjälpa att motivera eleverna"

En annan intressant vinkel, från en av lärarna, att se på OOP var att det med hjälp av metaforer gavs möjligheten att sätta logik och tankesätt i första hand.

Datalärare

"Att man först pratar om logiken i programmering innan man pratar syntaxen, det kan röra till det lite för en typ av elever som skulle kunna bli jättebra i programmering. Idag i programmering så krävs det kanske inte att man älskar att hålla på att mata bitar hit och dit som man gjorde förr."

En lärare fann en stor fördel i metaforer i dess användbarhet till individanpassandet av

lektioner. Speciellt användbart ansåg han det vara till nybörjarkurser i OOP relaterade medie ämnen då de flesta aldrig kommit tidigare haft kontakt med programmering.

Medielärare

"Försöker att undervisa så elevenpassat som möjligt och då gäller det att hitta eleverna, varje enskild elev på den nivån de befinner sig."

En del praktiska skäll angavs också. Två av lärarna tyckte att lektioner ibland, även de allt för abstrakta, kunde bli tunga för eleverna. Här ansåg han att man med hjälp av metaforer kunde kapa en väsentlig del av föreläsningstiden då långdragna förklaringar kunde ersättas av en välplanerad metafor.

Datalärare

"Här är metaforer bra då de hjälper till att korta ner en väldigt tung teoretisk förklaring..."

*10. Känner du att de (metaforer) hjälpt dig i din undervisning av objektorienterad programmering?

Tabell 11: Har metaforer hjälpt lärarna

Ja	Nej
6/6	0/0

Sex deltagare av sex var oreserverat positiva till att metaforer hjälpt dem som stöd i sin undervisning av OOP. De två lärare som nämnt under den föregående frågan att metaforer hjälper dem att spara både tid och ansträngning tog upp det faktumet igen. Tre av tre datalärare nämnde metaforer som en betydande del av sin undervisning. De anser att fördelarna av att utnyttja dem är helt enkelt för många för att de skall kunna ignoreras. Så här beskriver två av datalärarna metaforens roll i deras klassrum:

Datalärare 1

"Det här kanske man inte skall säga men metaforer har alltid varit väldigt praktiska för mig, och har besparat mig mycket tid och ansträngning när jag har förklarat saker"

Datalärare 2

"Metaforer är en nödvändig del i min undervisning. Metaforer har hjälpt mig mycket med något som jag anser är det största problemet med OOP att förklara den bakomliggande logiken. Jag vet inte riktigt om jag klarat att undervisa utan dem "

En av medielärarna kommenterade om att han tidigare haft en tendens att vara övertydlig och prata sönder sina lektioner. Här ansåg han metaforer hjälpt honom vidbehålla elevernas intresse då de tidigare kanske tröttnat.

Medielärare

"Ibland har jag gjort den (lektionen) för teoretisk i min ambition att ge dem så mycket info som möjligt och då tappar man dem i stället."

5 Analys och diskussion

I det kapitlet kommer jag att kort summera resultaten från det ovanstående kapitlet, och lyfta fram några av de mer intressanta trenderna. De här trenderna kommer sedan att användas till att analysera och diskutera arbetets syfte, och de fyra problemfrågorna från kapitel 3 som legat till grund för arbetet. Allt som tas upp här stöds direkt av teoristudien i kapitel 3.

För att underlätta summeringen kommer jag att använda arbetets struktur till min fördel när jag gör referenser till frågor från resultatkapitlet (därför t.ex. refererar 6.1 till kapitel 6 d.v.s. Resultat delen och fråga 1).

En mängd trender kan urskiljas ur resultatet. Här följer en kort summering av dem.

6.1: Den första trenden som urskiljas i resultatdelen är att alla lärarna använder sig av metaforer i undervisningen av programmering.

6.4: Den andra trenden var att de flesta av de tillfrågade favoriserade metaforer som ökade förståelse om själva grundkonceptet med OOP.

6.5: Trend nummer tre var även den att flertalet av de tillfrågade namngav här nämligen metaforer som ökade medvetandet om själva grundkonceptet med OOP.

6.6: Den fjärde trenden var att metaforerna människan och fordonet var de klart populäraste metaforerna. Hälften av de deltagande nämnde dem som sina favoritmetaforer.

6.7: Den femte trenden som framkom var att de metaforer som används sällan planeras. Uppstår eller används metaforer så uppstår de nästan uteslutande spontant.

6.9: Trend nummer sex var att merparten av de deltagande tyckte att metaforers abstrakta kvaliteter var en starkt bidragande orsak till att de använde metaforer.

6.10: Den sjunde var att hälften av lärarna känner att de inte skulle kunna klara sig utan metaforer i sin undervisning. Någon av dem kände till och med så starkt för dem att han skulle få svårt att klara sig utan dem. En annan tar upp att han tycker att de har en motiverande effekt som inte bör underskattas.

Här följer min redogörelse och analys av dessa slutsatser utifrån dessa just nämnda trender.

5.1 Dessa metaforer använder lärare

Följer man trend 6.4, 6.5 och 6.6 ser man tydligt att den dominerande metaforen i undervisning av OOP är metaforer som ökade förståelse om själva grundkonceptet med OOP. Det kommer inte direkt som en överraskning då det är grunden i den mesta OOP relaterade nybörjar undervisningen.

5.2 Användning av metaforer - en medveten, eller en kanske omedveten, handlig

Det här resultatet kan diskuteras. Av resultatet att döma så framgår det tydligt att användandet av metaforer är en medveten handling. Dock framgår det tydligt att det för den delen inte behöver ligga någon direkt planering bakom. Ser man på trend 6.7 så planeras metaforer sällan utan uppstår det så sker det spontant. Det här medför kan diskuteras hur medveten handlingen verkligen är. Med metaforers stöd hanterar man undervisningen av komplexa koncept bättre än då man försöker använda ordagranna förklaringar.

5.3 Användning av metaforer i klassrummet

Går vi tillbaka till trend 6.4, 6.5 och 6.6 kan vi uttyda ett mönster i användningen av metaforer i klassrummet. Av de tre trenderna framstår metaforens roll i undervisningen av OOP relaterade ämnen att först och främst agera som en grund.

Enligt trend 6.5 och teorierna av Goatly (1997), Ortony (1993), Thang N. Nguyen (2003) och George Lakoff, Mark Johnssons (1980) att döma finns det flera positiva samband i mellan metaforer och ökad förståelse.

5.3.1 Varför

En viktig del som inte berörts i analysdelen är varför lärarna använder sig av metaforer i klassrummet. Av både trend 6.9 och teoridelen framgår det att undersökningsgruppen anser

att det stora hindret är att ämnet är så pass abstrakt. Elever tycker helt enkelt att det är tråkigt att skriva 50 rader kod för att få sitt namn uppskrivet 10 gånger. De känner att de varken förstår vad de har skrivit eller varför de borde lära sig det när de inte ser sin framtid inom programutveckling. Metaforers roll som en katalysator för motivation nämns knappt, kanske är de som tidigare konstaterat i intervjuerna underförstådda. Möjligheten finns ju att om studien bestått av fler intervjuer att den motiverande effekten benämnts mer.

Det här stödjer det som Ortony (1993) pratar om i den teoretiska delen. Elever kommer i kontakt med ämnen som de inte kan se och helt saknar erfarenhet av. Som även teorin av Ortony(1993) visat så är det här en stor flaskhals för inläring.

5.3.2 Metaforer en deus ex machina

Trend 6.10 för upp en intressant fråga. Kan man använda metaforer för mycket i sin klassrumsundervisning? En risk kan finnas att lärare relaterar till metaforer som en sorts deus ex machina, en maskin som löser alla bekymmer. Detta känns som en farlig utveckling. Metaforer är nämligen bara ett redskap för lärare att använda för att underlätta i sitt arbetsliv, och som de flesta redskap kan det missbrukas och användas fel. Överflödiga metaforer kan nämligen skapa både förvirring och agera grus i elevernas mentala maskineri. Som Ortony (1993) nämner så finns det en ordagrann betydelse som enklare för fram det som skall föras fram så bör man kanske använda den. För att kunna få ut maximalt av metaforers kvalitéer att förmedla förståelsen tycker jag att man först måste identifiera vilket behov som existerar i klassen för metaforer (faktorer som påverkar och hindrar). De kanske inte alls har något behov av metaforer i sin undervisning utan klarar sig utmärkt utan att man måste blanda in metaforer. En av deltagarna i undersökningen använde sig t.ex. inte av metaforer i fall han inte kände att eleverna var i behov av det. Han elevanpassade sina lektioner.

5.4 Vanliga metaforer i klassrummet

Går man efter trend 6.6 så är människo- och fordons metaforer de vanligaste metaforerna. Det här kommer inte som en överraskning då de används i undervisningen av grunderna i ämnet. De här exemplen berörs även i teorikapitlet som legat till grund för studierna.

6. Slutsats

I det här sista av mina kapitel ger jag mina egna slutsatser av resultatet av mitt arbete. Den här slutsatsen baseras på tre källor: mina erfarenheter från arbetets teorikapitel, resultatkapitel som redovisas ovanför och min egen erfarenhet ifrån skolans värld.

Har jag med min undersökning lyckats uppnå mitt syfte; att kartlägga lärares bruk av metaforer som redskap och stöd i sin undervisning i OOP? Ja, jag känner att mitt syfte har uppfyllts. Då undersökningsgruppens storlek varit relativt begränsad i omfång så får svaret dock tas för vad det är. Baserat på de resultat jag fått fram, den teori som legat till grund för arbete och mina egna erfarenheter så drar jag slutsatsen att metaforer används aktivt i OOP-relaterad undervisning. Användandet sker dock till största del spontant, och graden av användandet varierar mellan de tillfrågade lärarna. Den överväldigande positiva responsen för metaforer var något jag inte helt väntat mig. Tyvärr, så verkar det som att den största delen av användandet av metaforer sker spontant. Endast EN av lärarna hade planerat de metaforer han använde sig av. Spontanitet i undervisningen är inte nödvändigtvis dåligt, men då metaforer är komplexa tankar känns det viktigt att de planeras bra för att kunna uppfylla sina syften. Det verkar finnas ett tydligt positivt samband hos lärare mellan metaforer och de egenskaper teorin tillskriver dem. Den ökade förståelsen som de skapar för avancerade koncept nämns genomgående i resultatet som metaforers största fördel. Då den här fördelen är den som huvudsakligen tas upp i all litteratur i ämnet så var resultaten inte någon överraskning. Vad som var konfunderade var i stället hur få av de övriga fördelarna som nämndes, den motiverande faktorn nämns exempelvis endast en gång. Anledningen till att de inte nämndes kan vara att de ansågs för självklara. Så pass självklara att lärarna helt enkelt inte tänkte på dem. Möjligheter finns att de nämns om de deltagande givits mer tid att svara på att frågorna.

För att få ett mer urtömmande och detaljerat redovisning på svaret på arbetets syfte låt oss återgå till sida fem i kapitel tre för en genomgång av min frågeställning. Här följer nu frågeställningen och dess svar:

1 Använder lärare metaforer i klassrummet. Är det en medveten handling?

Ja. Lärare använder sig av metaforer i klassrummet. Av de sex lärarna i undersökningsgruppen så använder sig alla sex av metaforer på ett eller annat sätt i sin undervisning. Är det då en medveten handling? Ja och nej, lärare har oftast en tanke bakom det men erkänner att användandet ibland sker som en omedveten handling.

1 Varför använder läraren metaforer?

Sett till vad litteraturen säger, och från vad som framgått i min undersökning och de resultat den gav så kan man ge flera anledningar till varför metaforer används. Den vanligaste anledning till dess bruk i undervisningssituationer verkar dock vara att de har potentialen att öka förståelse. Med utgångspunkt från mina observationer känns det dock ibland som att metaforer för en del lärare är ett nödvändigt ont och att de används för att det inte finns något bra sätt att förklara det de vill göra i ord, eller att de inte har något annat sätt att nå fram till eleverna.

1 Hur används metaforer i ett klassrum?

Även fast metaforer ofta används är bruket av dem sällan planerat. När de används så är det i stort sett uteslutande spontant. Lärarna verkar känna att de tillför något till

undervisningen men inte tillräckligt mycket för att spendera tid till att planera dem.

1 Vilka metaforer är det som används i klassrumssituationer?

De vanligaste metaforerna som används verkar helt klart vara de som ökar förståelsen för struktur och de bakomliggande tankarna med OOP.

Vad betyder då dessa frågor? Jag föreslår att användandet av metaforer i OOP borde öka och ges mer utrymme i introduktionskurser i programmering. De får uppmärksamhet men för tillfället känns det för spontant, ingen egentlig tanke verkar ligga bakom användandet. Ett sätt att adressera problemet skulle kunna vara att lärare samverka mer. Lärare behöver lära oss att använda metaforer på ett mer effektivt sätt, vi behöver veta hur vi skall gå tillväga med det här. Genom att läraren lär sig att samtala mer om metaforer så kanske det är möjligt att öka medvetandet om dem.

7 Förslag till vidare forskning

Jag tycker generellt att det behövs mer forskning kring varför lärare inte använder sig mer metodiskt av metaforer. Varför planerar lärare inte de metaforer de använder sig av när de är väl medvetna om deras fördelar? Detta har påpekats som en brist hos lärarnas användande i min undersökning. Därför tycker jag att det här är frågor att fokusera sig på för den som vill göra vidare forskning inom området.

8 Sammanfattning

Syftet med det här arbetet är att ta till kartlägga hur metaforer brukas som redskap och stöd i OOP relaterad undervisning. Utgår man från litteraturen framgår det tydligt att metaforer har en positiv effekt på inläring av ämnen, inte bara i programmerings relaterade ämnen utan även i andra. Metaforer används som stöd i undervisningen av ett flertal olika ämnen så som matematik, historia m.m. Potentiellt har de stor kapacitet som stöd i undervisnings situationer. De kan öka förståelse för svåra koncept, motivera elever till att vilja studera och vara praktiska redskap som kortar ner långa och teoretiskt tunga lektioner. Dess potential som hjälpmedel för att öka förståelse av svåra koncept är möjligtvis dess största fördel. Bland metaforers övriga potentiella fördelar bör dess motiverande effekt heller ej underskattas, motivation är en vital del i all inläring. Ett applikationsområde för metaforer som belyses i arbetet och kanske inte är belyst tidigare är metaforers roll för att ändra elevers sätt att se på programmering. Elevernas sätt att se på programmering kan spela en viktig roll för hur mycket de lär sig i sin programmeringsundervisning.

I intervjuerna deltog sex lärare. Gruppen var homogen och bestod av tre medielärare och tre datalärare. För att få fram det resultat som analyseras och diskuteras i rapporten användes främst en kvantitativ metod. Den kvantitativa metoden som valet föll på var en så kallad djup intervju. Valet av den här formen av metod föll på dess flexibla upplägg och kapacitet att tvinga deltagarna att plocka fram information. Ingen av de intervjuade lärarna behövde dock pressas och de delade hjärtligt med sig av sin kunskap. I resultatet framkom att lärare har stöd för metaforer i sin undervisning av OOP. För merparten av lärarna är användandet av metaforer en medveten handling, få av dem lägger dock ner någon som helst tid för planering av dem. Den ökade förståelsen som metaforer ger i OOP undervisning angavs genomgående av alla de sex tillfrågade som metaforers största fördel. Få berörde dock dess motiverande egenskaperna i intervjuerna.

Då den ökade förståelsen som metaforer ger gavs som det främsta skälet till att de tillfrågade använde dem var det också den typen som hjälpte till med det här populärast bland lärarna. I diskussionen knyter jag ihop resultatet med litteraturen.

Min generella slutsats är att det är bra att metaforer används men att användandet för tillfället är för spontant. Mer tanke behöver läggas ner bakom användandet av metaforer och i framtiden hoppas jag att fler lärare som undervisar i programmering kommer att göra just det.

Referenslista

Barker, Harold, Grant, Phil, Jobling, Chris & Townsend, Peter (1993). *The object-oriented paradigm: a means for revolutionising software development*. Computing & Control Engineering Journal. Vol 4, No.1. ss.10-14.

Bruce, Cristine, Buckingham, Lawrence, Hynd, John, McMahon, Camille, Roggenkamp, Mike & Stoodley, Ian (2004). *Ways of Experiencing the Act of Learning to Program: A Phenomenographic Study of Introductory Programming Students at University*. Journal of Information Technology Education, Vol 3.

Clark, Dan (2002). *An Introduction to Object-Oriented Programming with Visual Basic.NET*. USA: Springer-Verlag New York.

Goatly, Andrew (1997). *The Language of Metaphors*. London/New York: Routledge.

Halvorsen, Knut (1992). *Samhällsvetenskaplig metod*. Lund: Studentlitteratur.

Lakoff, George & Johnsons, Mark (1980). *Metaphors we live by*. Chicago: The University of Chicago.

Nguyen, Thang (2003). *Developing and Using an Integrated Collection of Rich Real-Life Analogies and Metaphors in the Teaching of Complex IT Concepts*. Information System Education Journal, Vol 1, No.5.

Ortony, Andrew (1993). *Metaphor and Thought*. Cambridge: Cambridge University Press.

Patel, Runa & Davidson, Bo (1994). *Forskningsmetodikens grund: Att planera, genomföra och rapportera en undersökning*. Lund: Studentlitteratur.

Ravenscroft, Ian (2005). *Philosophy of mind: A Beginner's Guide*. Oxford: Oxford University press.

Elektroniska referenser

Ardalan, Kavous (1998). On the Use of Entertaining Metaphors in the Introductory Finance Course. Financial Practice and Education.

Tillgänglig: <http://207.36.165.114/JAF/fpess9812.pdf>

[läst 07-05-11]

Encyclopedia Britannica.

Tillgänglig: <http://www.britannica.com/>.

[läst 07-05-15]

Sample, Ian (2004). Constant drive for new computers is taking mega bites out of environment.

Tillgänglig: <http://www.smh.com.au/articles/2004/03/08/1078594301529.html>.

[läst 07-05-07]

Skolverket (2006/2007). Kursinformationssystem för skolan.

Tillgänglig:

<http://www3.skolverket.se/ki03/front.aspx?sprak=SV&ar=0607&infotyp=15&skolform=21&i>

d=10&extrald=0.
[läst 07-05-15]

Intervjufrågor

- 1 Så vad tycker du om min ide att undersöka metaforers användning i objekt-orienterad programmerings undervisning?
- 2 Hur länge har du undervisat i, någon form av, objekt-orienterad programmering?
- 3 Hur skulle du definiera begreppet metafor?
- 4 Har du något exempel på någon metafor du använt i din lektion?
- 5 Har du någon favorit/er som du medvetet eller omedvetet använder dig av?
- 6 Finns det någon speciell metafor du känner skulle kunna hjälpa elever att förstå? (du behöver inte nödvändigtvis ha använt den själv)
- 7 Händer det att du har lagt ner tid på planering av metaforer?
- 8 Händer det du skapat metaforer spontant, i ögonblicket stund?
- 9 Varför använder du dig av metaforer?
- 10 Känner du att de (metaforer) hjälpt dig i din undervisning av objekt-orienterad programmering?