

RAPPORT RAPPORT

Undersøkelse av virkningene av nytt opplegg for undervisning i objektorientert programvareutvikling

Hallstein Asheim Hansen

Hans Engebretsen



Rapporter fra Høgskolen i Buskerud

Nr. 51

**Undersøkelse av virkningene av nytt opplegg
for undervisning i objektorientert
programvareutvikling**

Av

Hallstein Asheim Hansen og Hans Engebretsen

Kongsberg 2004

HiBus publikasjoner kan kopieres fritt og videreformidles til andre interesserte uten avgift.

En forutsetning er at navn på utgiver og forfatter(e) angis- og angis korrekt. Det må ikke foretas endringer i verket.

ISBN 82-91116-69-5

ISSN 0807-4488

Undersøkelse av virkningene av nytt opplegg for undervisning i objektorientert programvareutvikling.

Høgskolelektor Hallstein Asheim Hansen, hallstein.asheim.hansen@hibu.no

og

Høgskolelektor Hans Engebretsen,
hans.engebretsen@hibu.no

Institutt for teknologi
Høgskolen i Buskerud

Sammendrag

Denne artikkelen omtaler en undersøkelse av hvordan et nytt undervisningsopplegg i faget "Grunnleggende programvareutvikling" fungerer. Opplegget baserer seg på et dataprogram med tilhørende lærebok, som gjør det mulig å lære studentene om objekter fra første dag.

Undersøkelsen omfatter en True/False-test (T/F-test) samt dybdeintervjuer med sju studenter. T/F-testen ble gjennomført på 13 studenter i 1. klasse, og samme test ble utført på 18 studenter i 2. klasse.

Testen viser at 2. klasse har noe bedre grep om meget sentrale begreper innen objektorientering. Andelen av studenter som svarte rett på de mest sentrale spørsmålene var 56 % for 2. klasse mot 46 % for 1. klasse. Denne forskjellen er som forventet.

Dybdeintervjuene viser at studentene jevnt over er godt fornøyd med det nye undervisningsopplegget, og de mener selv at det programmet som benyttes er godt egnet for å få et godt grep om de sentrale begrepene i objektorientering.

Det ble også undersøkt hvor fornøyd studentene er med gruppearbeidet som benyttes, og her var det stor enighet om at dette fungerer bra på det nivå studentene befinner seg på i 1. semester, mens de aller fleste synes at det er greit å slutte med dette opplegget etter ett semester.

1. Introduksjon

De fleste høyskoler og universiteter i Norge som tilbyr studier i Informatikk/Datateknikk gir i første semester et emne hvor hovedfokus er å gi studentene en innføring i å programmere. Det vil si – lage programmer/systemer som for eksempel Internet Explorer og Excel ved å skrive instruksjoner som datamaskinen siden utfører. Ved HiBu (Høgskolen i Buskerud) gis emnet DGPV-1000 - ”Grunnleggende programvareutvikling”.

Når systemer utvikles i industrien er de ofte svært omfattende – så omfattende at de krever en strukturert tilnæringsmåte. Den aksepterte måten (State of the Art) å strukturere programmer på kalles *objektorientering*. Objektorientering går kort fortalt ut på at man representerer objekter – ”ting” – som finnes i virkeligheten, i datasystemer. Et eksempel på objekter kan være ”menneske”, ”avis” og ”bil”.

Studentene i datateknikk har helt siden høsten 1998 hatt Java som programmeringsspråk i faget ”Grunnleggende Programvareutvikling”. I flere år før dette brukte man C++ som programmeringsspråk. Høsten 1998 tok man i bruk Java med tilhørende lærebok [1]. Denne læreboka introduserer objektorientert tankegang i kapittel 4. De senere år har det dukket opp en flom av lærebøker i objektorientert programmering, de fleste basert på programmeringsspråket Java. Felles for de aller fleste av disse er at de reiser en sentral problemstilling:

Hvor tidlig kan vi innføre begrepet objekter ?

Høsten 2002 ble et verktøy kalt BlueJ med tilhørende lærebok [2] lansert. Dette muliggjorde en undervisning i Java, der en begynner å lage objekter fra dag en. På denne måten vil studentene få et godt grep om objektorientert tankegang uten at de må lære seg visse metoder som de ikke har mulighet til å forstå betydningen av på det tidspunkt de lærer dem.

Høsten 2003 ble undervisningsopplegget tilpasset den ny læreboka, for å lære studentene objektorientert tankegang helt fra starten av.

Siden objektorientert tankegang kan være vanskelig å tilegne seg, er vi svært interessert i å se om vi har lykket bedre med dette enn tidligere. Objektorientert tankegang er et omfattende tema og vi har begrenset oss til å undervise følgende grunnleggende temaer:

1. Objekter og klasser

En *klasse* beskriver en spesiell type objekter, og inneholder instruksjoner og egenskaper som gjelder for alle objekter av samme klasse. For eksempel kan et eksempel på en klasse være ”personbil”, og alle personbiler vil da være objekter av

denne klassen. Instruksjonene for "girskifte" og "nedrulling av vindu" spesifiseres kun en gang – når man lager klassen.

2. Egenskaper

En egenskap er en verdi som beskriver et objekt. En egenskap defineres i klassen, og hvert objekt av en enkelt klasse får sin versjon av egenskapen. For eksempel vil klassen for bil ha egenskapene "eier", "merke" og "farge". Et objekt kan ha verdiene "Hallstein", "Ford" og "Grønn", mens et annet har verdiene "Hans", "Audi" og "Sølvgrå".

3. Konstruktørmeter

Når man lager et nytt objekt av en gitt klasse, ønsker man svært ofte å sette opp en del av egenskapene til objektet med en gang – dette gjøres ved hjelp av en samling instruksjoner som kalles for en konstruktørmeter.

4. Modellering av virkeligheten.

Objekter er (blant annet) ment å være abstrakte representasjoner av "ting" fra virkeligheten.

Høsten 2002 endret vi undervisningsformen i emnet. Tradisjonelt har det vært forelest i to timer, fulgt av tre timer med individuelt arbeid med oppgaver knyttet til forelesningen, eller med obligatoriske oppgaver. Da var underviseren som holdt forelesning til stede for å hjelpe til. F.o.m. høsten 2002 ble det holdt en times forelesning, samt en times undervisning i grupper. Hver gruppe var ledet av en underviser som delvis veiledet i oppgaveløsning og delvis utdypet teori fra forelesningene. Oppgaveløsningen ble enten gjennomført ved at gruppa satt sammen rundt en pc, ved at man diskuterte løsninger eller ved at hver student hadde sin pc og underviseren hjalp dem etter tur.

Vi ønsket å finne ut om denne endringen i undervisningsopplegget har hatt en positiv effekt på studentenes læring generelt og objektorientering spesielt.

2. Metoder

Her redegjør vi først for hvordan BlueJ sitt pedagogiske system er lagt opp. Så beskriver vi hvordan vi ved HiBu har gjort bruk av dette. Videre knytter vi dette sammen med den nye undervisningsmetoden vi har introdusert. Til slutt redegjør vi for de undersøkelsene vi har gjort for å få svar på den problemstillingen vi presenterte innledningsvis.

2.1. BlueJ sitt pedagogiske system

BlueJ er et utviklingsmiljø konstruert for å lære studenter å tenke objektorientert. Det er både programmert i og tilbyr utvikling i programmeringsspråket Java – et språk som er:

- objektorientert
- plattformuavhengig
- enkelt å lære
- egnet til såvel små som store programmer

De fleste utviklingsmiljøer er laget for å hjelpe erfarne programmere i utviklingsprosessen, og er ikke nødvendigvis egnet for en fersk student. De tar utgangspunkt i de tekstlige instruksjonene til programmene, noe som gjør at det å se de objektorienterte konseptene blir vanskelig. Et eksempel på brukergrensesnittet til miljøet JCreator (som HiBu bruker i de senere emnene vi tilbyr) finnes i figur 2.1.

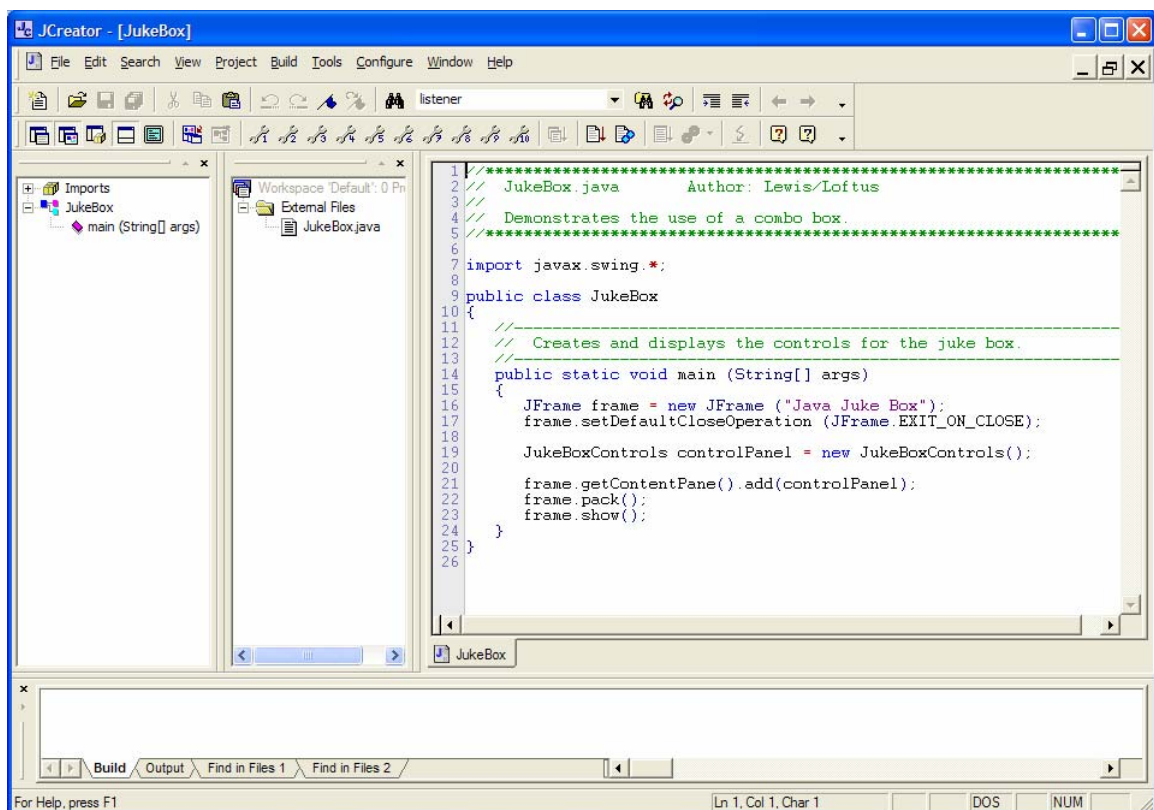
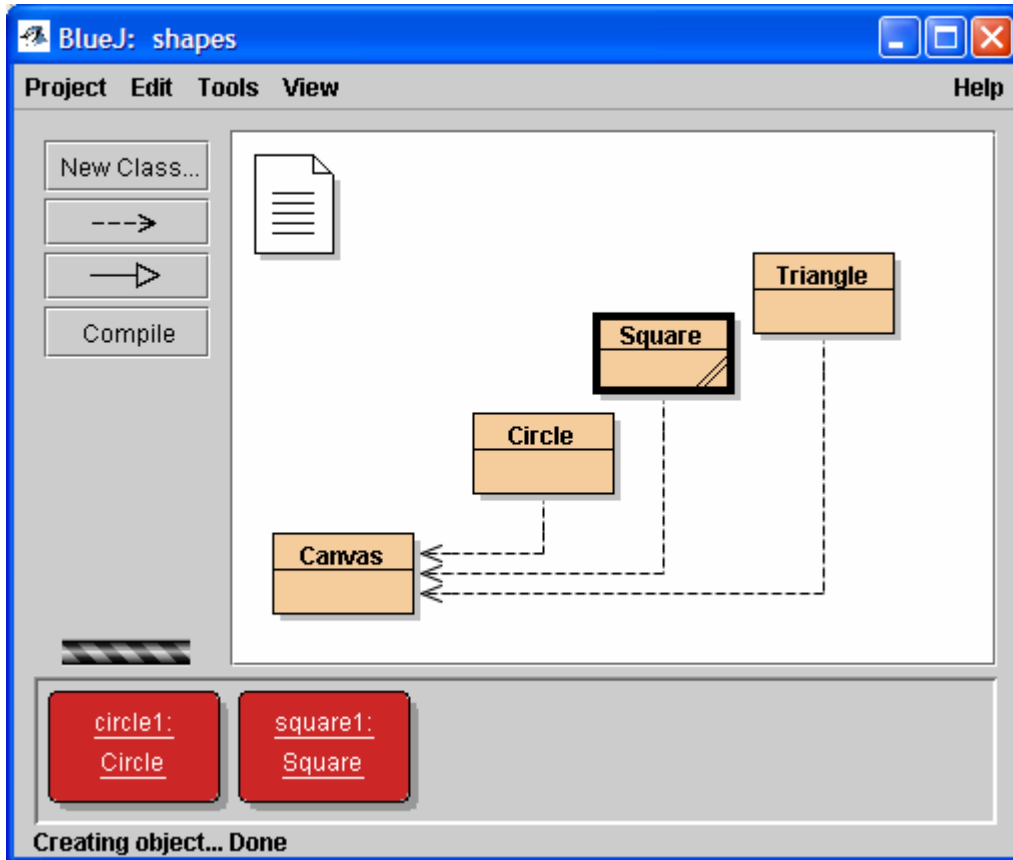


Fig. 2.1 Brukergrensesnitt for JCreator

BlueJ derimot, tilbyr et grensesnitt for direkte manipulasjon av klasser og objekter, se figur 2.2.

BlueJ har et vesentlig enklere brukergrensesnitt med færre knapper og menyer. I det store hvite feltet opp til høyre har vi de forskjellige klassene som finnes i det systemet vi utvikler – de er illustrert som rektangler. De underliggende metodene og

egenskapene er skjult. Hvis man høyreklikker på en klasse kan man lage ett eller flere objekter av denne klassen. Disse finner vi igjen som rektangler med avrundede hjørner i området nederst på skjermen . Dette hjelper studenten å forstå forskjellen på og sammenheng mellom klasser og objekter .



*Fig. 2.2
Brukergrensenitt
for BlueJ*

Når vi høyreklikker på klassen blir objektet instansiert (laget) blir også konstruktørmotoden utført. Måten det gjøres på i BlueJ er at brukeren får opp en dialogboks (figur 2.3) hvor egenskapene som kreves for å instansiere det nye objektet må skrives inn av brukeren.

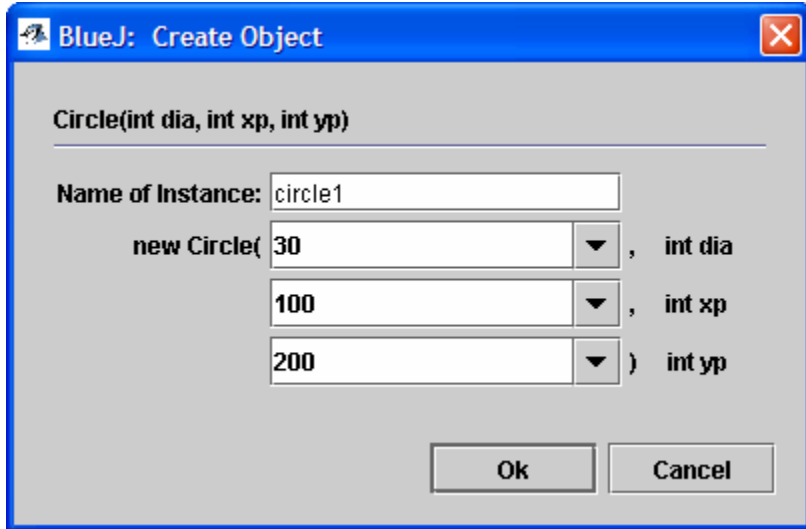


Fig. 2.3 Skjembilde for å lage et nytt objekt i BlueJ

Når det gjelder å modellere virkeligheten viser BlueJ ikke bare et grafisk bilde av klassene, men også relasjoner mellom dem. Etter hvert som undervisningen skrider fram har studentene behov for å endre på koden i eksemplene som følger med BlueJ. Ved å dobbeltklikke på en objektklasse i diagrammet får de opp koden for klassen og kan gjøre ønskelige endringer.

2.2. *Bruk av BlueJ ved HiBu*

Etter at BlueJ dukket opp høsten 2002 ble det raskt klart at fagpersonalet som er og har vært involvert i undervisning av grunnleggende programvareutvikling var samstemte om at dette var noe som måtte taes i bruk så fort som mulig. På fagseminaret ved instituttet ble det derfor bestemt å ta i bruk i faget DGPV-1000. Det å ta i bruk ei ny lærebok fører vanligvis til omfattende omlegginger av leseplanen. Boka passet vårt undervisningsopplegg så bra at vi kunne bruke den kapittel for kapittel, fordi den dekket de viktigste temaene på en god måte. Det eneste ekstra forarbeidet som måtte gjøres var å dele opp kapitlene i passende bolker, da enkelte kapitler var for omfattende til å dekkes på en undervisningsdag. Mange ble delt over to dager, og ett ble delt over tre dager.

2.3. *Undervisningsform*

Høsten 2002 skjedde det en radikal omlegging i undervisningsform for faget ”Grunnleggende programvareutvikling”. Tradisjonelt bestod undervisningen av 2-3 timer forelesning å formiddagen og veiledning med oppgaveløsning på ettermiddagen. Et problem var at mange studenter (som helt klart hadde behov for hjelp) ikke var på skolen da veiledningen startet. Vi la derfor om til kun en time forelesning, der ren teori ble presentert. I tillegg fikk studentene gruppeundervisning.

Gruppene bestod av 4-5 personer med en underviser som veileder. Gruppene var delt inn i tre forskjellige nivåer basert på studentenes egne vurderinger av forkunnskaper i programmering. Studentene fikk imidlertid visse objektive kriterier til hjelp i denne vurderingen. Veiledningstimen varte en time, og det var opp til hver enkelt gruppeveileder hvordan denne ble benyttet. I de gruppene med høyest nivå ble den ofte brukt til å utdype mer avanserte ting enn det som var gjennomgått i forelesningene, mens den for de svakeste gruppene ble brukt til å sikre at et minimum av basiskunnskap i faget ble innøvd.

2.4. *Undersøkelser*

For å undersøke om vi faktisk hadde oppnådd formålet med undervisningsomleggingen, valgte vi å gjøre diverse undersøkelser:

- Bruke egne erfaringer. Begge forfatterne deltok som gruppeveiledere i løpet av gjennomføringen av emnet, mens en av oss (Engebretsen) var ansvarlig for utformingen av det nye undervisningsopplegget samt at han holdt brorparten av forelesningene.
- Test av overflatekunnskap ved hjelp av True/False-test.
- Samme test ble utført for studentene på klassetrinnet over.
- Evaluering av en obligatorisk oppgave studentene leverte i faget DVPV-1020 - "Videregående programvareutvikling". Oppgaven var todelt. I første del skulle man modellere et system og levere en beskrivelse av dette. I andre del senere skulle man implementere systemet og levere inn de ferdige programfilene sammen med en revidert modell av beskrivelsen fra del 1.
- Dybdeintervju med et gruppe studenter. Studentene ble valgt ved at vi trakk ut fire studenter fra hver av de tre nivåene. Noen av disse ble fjernet fordi de ikke lenger deltok aktivt i faget, og ikke leverte inn oppgaven. Et par andre falt av fordi de ikke hadde lyst til å være med. Vi stod derfor igjen med 7 studenter til dette intervjuet

3. Resultater

Vi redegjør her for de observasjonene vi gjorde i forbindelse med undervisningen i emnet, resultatene av True/False-testene vi gjorde. Vi sammenligner resultatet av testene med 2. klassingenes resultater. Så drøfter vi besvarelsene fra den obligatoriske oppgaven. Til slutt forteller vi om resultatene fra dybdeintervjuet som ble gjort med et utvalgt studenter.

3.1. *Egne erfaringer*

Det har vært prøvd med en del forskjellige metoder i gruppeundervisningen. Disse kan kort skisseres som følger:

1. En slags miniforelesning for hele gruppa, der underviser tegner og forteller, men med langt større dialog enn normalt i en forelesning.
2. Utstrakt grad av felles oppgaveløsning der hele gruppe på 3-5 personer sitter ved samme PC og jobber sammen om å løse et problem.
3. Individuell veiledning av det enkelte gruppe medlem ut fra hvilken oppgave dette medlemmet holder å med i øyeblikket.

Vår egen opplevelse av disse alternativene er at alternativ 1 og 2 gir best resultat. Dette framgår også av dybdeintervjuene. Det er viktig at det hele blir en gruppeprosess der veileder i tillegg til det rent faglige også har et stort ansvar for å sørge for at det enkelte gruppe medlem bidrar i prosessen. Dette gjøres bl.a. ved å adressere spørsmål og henvendelser direkte til enkeltmedlemmer:

- Hva tror du om dette Ola?
- Er det lurt det som Kari foreslår her?
- Kan noen av dere komme å andre alternativer?
- Husker du, Jan, hva vi gjennomgikk om dette temaet forrige tirsdag?

Når man benytter en slik undervisningsform vil hver enkelt student få et visst press å vise de andre i gruppa at de henger med. For de som har alvorlige problemer faglig vil disse selvfølgelig også blottstilles for de andre gruppe medlemmene. og dette krever en viss forsiktighet. Det største problemet har vi i så måte hatt med noen gruppe medlemmer som har satt seg opp på høyeste faglige nivå, men ikke forstår selv det mest grunnleggende. Det er imidlertid viktig å understreke at dette dreier seg om kun noen få studenter.

Når studentene sitter og jobber 3-4 ved samme PC er det viktig å sørge for at jobben med å skrive går på omgang, og at man passer på at ingen av deltagerne ”sovner”.

3.2. *Tru/False-test*

Vi lister her spørsmålene, gir en kort forklaring til meningen bak dem, og gir informasjon om hvor sentralt dette spørsmålet er i forståelsen av objektorientering.

1 - ”Vi kan bare lage et objekt av hver objektklasse vi definerer”

En klasse kan brukes til å lage et vilkårlig antall objekter, for eksempel kan klassen bil brukes til å lage et Ford-objekt og et Audi-objekt.
Dette spørsmålet er **meget** sentralt.

2 – ”Vi kan definere en objektklasse som ikke har egenskaper”

Egenskaper finnes i de fleste klasser, men det er ikke noe krav om dette, og det har absolutt mening å gjøre det, selv om det ikke forekommer så ofte..

Dette er ikke så sentralt, men noe man bør kjenne til.

3 – ”Dersom to metoder heter det samme må de ha forskjellig antall parametre”

Dette spørsmålet berører to mekanismer som er sentrale, men som det krever modning å få tak i.: (1)Method Overloading og (2)Method Overriding. For (1) må parametre ha forskjellig type, mens de i tilfelle (2) også kan være av samme type.

4 – ”En objektklasse kan kun ha en konstruktørmetode”

En klasse kan ha et vilkårlig antall konstruktørmetoder, avhengig av hvor mange egenskaper som man ønsker å spesifisere med en gang, og hvordan måte man ønsker å spesifisere disse egenskapene. Dette spørsmålet er meget sentralt, og tema har vært grundig belyst i såvel undervisning som eksamensoppgaver.

5 – ”En mutator-metode er en metode som endrer et objekts tilstand”

To typiske typer metoder er mutatormetoder som forandrer på en egenskap i et objekt, og aksessormetoder som gir oss verdien til en av objektet sine egenskaper. Spørsmålet er sentralt, men man kan godt ha forstått prinsipper med objektorientering uten å kjenne å knytte noe til begrepet mutatormetode.

6 – ”Et objekts egenskaper beskriver objektets tilstand”

Det eneste som skiller to objekter av samme klasse fra hverandre er verdiene til egenskapene. Disse egenskapene utgjør da tilstanden til objektene. Dette spørsmålet er sammen med spørsmål 1 det mest sentrale i testen.

7 – ”En accessormetode er en metode som endrer verdien på en eller flere av objektets egenskaper”.

Dette er et forholdsvis sentralt punkt

8 – ”Konsistens (cohesion) er noe vi gjerne vil oppnå”

Dette er et sentralt begrep i objektorientering og referer til det vi kaller encapsulation. Dette betyr at vi samler det som naturlig hører sammen i et objekt. Altså selve hovedpoenget med objektorientering

9 – ”Kobling (coupling) er noe vi gjerne vil oppnå”

Et viktig prinsipp for objektorientert utvikling er at vi ikke ønsker at ikkerelaterte data og instruksjoner skal kobles for tett sammen. Dette er sentralt og henger sammen med temaet konsistens(spørsmål 8).

På nest side(tabell 3.1) finnes en sammenstilling av resultatene. I denne finner en hver enkelt students svar på hvert enkelt spørsmål. Ved riktig svar gies 1 poeng, ved feil svar - 1 og ved manglende svar 0 poeng. Det er derfor mulig å få en negativ total poengsum. Noe vi ser er tilfelle for en del studenter. De studentene som ble dybdeintervjuet er angitt i tabellen som Student 1, 2 osv.

1. klasse	Spørsmål nr.									Poeng		
	1	2	3	4	5	6	7	8	9	Alle spm.	Spm. 1,2, 3,4 og 6	Mest sentrale spm.(1,4 og 6)
Student 1	1	1	-1	1	1	1	1	1	-1	5	3	3
Student 2	-1	1	1	-1	-1	1	-1	1	1	1	1	-1
Student 3	1	1	1	1	1	1	1	1	1	9	5	3
Student 4	1	1	1	1	1	1	1	1	1	9	5	3
Student 5	1	-1	-1	1	1	0	-1	0	0	0	0	2
Student 6	1	1	-1	-1	1	-1	1	1	1	3	-1	-1
Student 7	1	1	-1	-1	1	1	0	0	0	2	1	1
	1	1	1	1	0	1	1	1	1	8	5	3
	1	1	1	-1	1	-1	1	-1	-1	1	1	-1
	1	-1	-1	-1	1	0	-1	-1	-1	-4	-2	0
	1	1	-1	1	-1	-1	-1	1	1	1	1	1
	1	1	1	1	1	1	1	1	1	9	5	3
	1	1	1	1	1	1	1	1	1	9	5	3
Andel(%) riktig	92	85	54	62	77	62	62	69	62			
									Snitt:	4,08	2,23	1,46
									Snitt(intervjugruppe):	4,14	2,00	1,43
2. klasse												
	1	1	1	-1	-1	-1	1	-1	-1	-1	1	-1
	1	1	1	1	1	1	-1	0	0	5	5	3
	1	1	1	1	1	1	-1	1	1	7	5	3
	1	1	-1	1	1	-1	-1	1	1	3	1	1
	1	1	-1	1	1	-1	1	1	1	5	1	1
	1	1	1	-1	1	1	1	1	1	7	3	1
	1	1	-1	1	1	-1	-1	-1	-1	-1	1	1
	1	1	-1	1	1	1	1	1	-1	5	3	3
	1	1	1	1	0	1	0	1	-1	5	5	3
	1	-1	1	1	1	1	-1	1	1	5	3	3
	1	-1	1	1	1	1	-1	-1	-1	1	3	3
	1	1	-1	1	1	-1	-1	1	1	3	1	1
	1	1	-1	1	1	-1	-1	-1	-1	-1	1	1
	1	1	-1	1	1	1	1	1	-1	5	3	3
	1	1	1	1	1	1	-1	1	-1	5	5	3
	1	1	1	1	1	1	-1	1	1	7	5	3
	1	1	-1	1	1	1	1	1	-1	5	3	3
	1	1	1	1	1	-1	-1	1	1	5	3	1
Andel(%) riktig	100	89	56	89	89	61	33	72	44			
									Snitt:	3,89	2,89	2,23
Andel(%) som har full score på meget sentrale spørsmål:												
1. klasse	46											
2. klasse	56											
Intervjugruppe	43											

Tabell 3.1 Resultater av MC-test

3.3. Sammenlikning med 2.klasse

Den samme True/False-testen ble utført på 2. klasse også, og i tabellen på forrige side er disse også lagt inn. Studentene som går i andre klasse har blant annet hatt et emne som omhandler programmering i et annet objektorientert språk(C++), i tillegg til andre emner. Det var 18 studenter fra 2. klasse som tok testen.

Siden vi forandret opplegget for 1.klasse er visse spørsmål ikke relevant for studentene i 2. klasse. Det gjelder spørsmålene 5, 7, 8 og 9. For de andre spørsmålene ønsker vi å sammenlikne resultatene:

3.4. Evaluering av obligatorisk oppgave

I grove trekk kan vi si at de aller fleste studentene var gode til å utnytte det de hadde lært om objektorientering. Dette var også å forvente siden oppgavene dreide seg om ”fysiske” objekter som biler og trafikklys. Det er vår erfaring at å beskrive denne type objekter faller langt lettere enn å beskrive mer ”abstrakte” objekter som f.eks. vinduer i et vindusbasert brukergransesnitt.

3.5. Dybdeintervju

Det ble foretatt dybdeintervjuer av sju studenter. Vi presenterer først de sju studentene, så går vi igjennom hvilke temaer vi diskuterte med dem. Til slutt lister vi opp svarene vi fikk.

3.6. Temaer

Det første temaet vi ønsket å kartlegge var studentene sine tanker rundt den nye undervisningsformen med gruppeundervisning.

Så ville vi se nærmere på de objektorienterte konseptene vi har prøvd å lære dem: Har de skjønt dem, og hvorfor skjønte de dem?

Til slutt er vi interessert i å finne ut litt om prosessen rundt den obligatoriske oppgaven: Vanskeligheter, hjelp mottatt og innfallsvinkel til løsning.

3.7. *Tilbakemelding om temaene*

Vi presenterer svarene til studentene her. Vi deler dem inn etter kunnskapsnivå, spørsmål om undervisning, spørsmål om OO-kunnskap og spørsmål om den obligatoriske oppgaven.

To av studentene var jenter, to var fremmedspråklige. Fire av studentene var 20 år gamle, den eldste var 27 år gammel.

Student A er en mann på 20 år. Har norsk som morsmål. (B)

Studenten deltok i gruppeundervisningen og satte pris på denne. Likte nivået i gruppeundervisningen. Ønsket ikke å fortsette med denne i det videregående emnet som bygger på DGPV-1000.

Studenten har et godt grep på objektorienterte konsepter, og synes at BlueJ hadde et godt fokus på objekter.

Studenten fikk kun sporadisk hjelp til den obligatoriske oppgaven. Han synes det var spesielt vanskelig å få oversikt over løsningsmetode ut fra kunnskapsnivå.

Student B er en mann på 20 år. Har norsk som morsmål. (B)

Studenten deltok i gruppeundervisningen og satte pris på denne. Han følte at han fikk hjelp til å forstå begreper og programmeringshjelp. Han likte lærerkontakten og ville foretrukket å videreføre denne ordningen på det videregående emnet.

Studenten har et bra grep om objektorienterte begreper. Han mente at kombinasjonen av forelesning, gruppetimer og bruk av BlueJ gjorde at han forsto de objektorienterte konseptene.

Studenten synes at det var de designmessige sidene ved den obligatoriske oppgaven som var vanskeligst. Han hadde en programmeringsteknisk tilnærming til oppgaven.

Student C er en mann på 20 år. Har norsk som morsmål. (A)

Studenten deltok på gruppeundervisningen, men fikk ikke så mye ut av den. Han satte derimot pris på å ha en dedikert lærer. Han foretrekker undervisningsformen fra det videregående emnet.

Studenten kjente en god del av de objektorienterte konseptene fra før, men mente at forelesningene og også BlueJ hjalp ham med å forstå de konseptene han ikke kjente fra før.

Studenten hadde lite problemer med å løse den obligatoriske oppgaven. Han hadde en programmeringsteknisk tilnærming til oppgaven, men utvidet den med en virkelighetsnær/objektorientert tilnærming.

Student D er en mann på 22 år. Har norsk som morsmål. (B)

Studenten deltok på gruppeundervisningen, og synes at den var en grei arbeidsform i begynnelsen, men ikke i det videregående emnet.

Studenten har et middels godt grep om objektorientering. Han synes at forelesninger og egne studier, og i mindre grad BlueJ, var best egnet til å lære de objektorienterte konseptene.

Studenten synes at det var mye vanskelig ved den obligatoriske oppgaven, og fikk en del hjelp av studiekamerater. Han hadde en programmeringsteknisk tilnærming til oppgaven.

Student E er en kvinne på 27 år. Har norsk som fremmedspråk. (D)

Studenten deltok på gruppeundervisningen, og likte den samt det å ha tilgang på lærer. Hun savner gruppetimene og ville gjerne fortsatt med dem.

Studenten har middels kunnskaper om objektorientering og likte BlueJ sin oppbygning, og lærte de objektorienterte konseptene av dette systemet samt av forelesningene og gruppetimene.

Studenten synes det var vanskelig å komme igang med den obligatoriske oppgaven. Hun fikk god hjelp av medstudenter og faglærere. Hun hadde en virkelighetsnær innfallsvinkel til oppgaven.

Student F er en kvinne på 20 år. Har norsk som morsmål. (E)

Studenten jobbet stort sett individuelt, men satte pris på å få hjelp av faglærer. Ønsker ikke å fortsette med ordningen i det videregående emnet.

Studenten har middels kunnskaper om objektorientering og likte BlueJ som et verktøy for å lære seg konseptene. Det var nyttig, i tillegg til forelesning og eget arbeide. Foretrakk å gå over på utviklingsmiljøet JCreator etterhvert.

Det var design som skapte mest problemet for studenten: Hvilke objekter hun skulle bruke. Hun fikk litt hjelp fra andre. Hun hadde en virkelighetsnær innfallsvinkel til oppgaven.

Student G er en mann på 22 år. Har norsk som fremmedspråk. (E)

Studenten deltok på og likte gruppeundervisningen veldig godt. Han ville foretrukket å fortsette med den også i det videregående emnet.

Studenten har lave kunnskaper om objektorientering. Han synes at BlueJ var et godt verktøy for å lære objektorientering, og synes læreboka var god.

Studenten samarbeidet med en annen student om å løse oppgaven. Han synes at det vanskeligste ved å løse oppgaven var å bestemme hvilken strategi han skulle bruke.

4. Diskusjon

True/False- test

Spørsmål 1 er helt sentralt innenfor objektorientering og det er gledelig at bare 1 person har svart feil på dette. Gledelig var det også at kun 2 personer svarte feil på spørsmål 2, ettersom det er uvanlig med objektklasser uten egenskaper. Spørsmålene 3 og 4 hadde 5 og 6 gale svar, noe som tyder på at en stor del av studentene ikke har forstått en del spesielle egenskaper ved metoder. Spørsmål 5 (2 gale svar) og spørsmål 7 (4 gale svar) tyder på at studentene har delvis kontroll over to viktige former for metoder. Det viktige spørsmål 6 om tilstand var det 3 gale svar på. Spørsmål 8 og 9 om utvikling var det henholdsvis 2 og 3 gale svar på.

Resultatene tyder på at vi har lykket bra i å lære studentene noen helt grunnleggende objektorienterte konsepter. Dessverre må vi fokusere mer på metoder og egenskapene til disse.

Sammenlikning med 2.klasse

Mest sentrale spørsmål

Her ligger 2. klasse et stykke foran. lukker vi derimot ut de som har svart riktig på alle disse tre spørsmålene, ser vi at prosentandelen i 2. klasse ligger på 56, mens den i 1. klasse ligger på 46. Dette er en forventet utvikling, fordi de svakeste studentene trenger mer tid på å tilegne seg basiskunnskap.

Hvor var 1. klasse bedre enn 2. klasse?

Av de spørsmål som er relevante for begge klasser er det verdt å merke seg spørsmål 6. Dette viser at 1.klassingene har bedre forståelse av tilstanden til objektene. Det er også betenkelig at 2. klassingene har såpass dårlig grep om spørsmål 9. Begrepet kobling skal de ha vært innom i en del sammenhenger.

5. Dybdeintervju

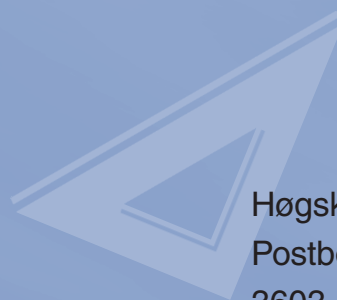
Når det gjelder erfaringene studentene hadde med undervisningsformen var det fem av syv studenter som satte pris på å jobbe i grupper, mens de to siste heller satte pris på å ha en lærer tilgjengelig. Tre av studentene ville fortsette med ordningen videre. Av dette slutter vi at gruppeundervisningen har sin plass i innføringskurset, men ikke i det videregående kurset. Andre ideer kan være å legge til rette for de som ikke vil delta i gruppeundervisningen i innføringskurset, samt å tilby det for de som ønsker det i det videregående kurset.

Når det gjelder objektorientert konsept, så tegner det seg en ganske klart bilde: Studentene satte pris på og lærte av BlueJ verktøyet, men det var i seg selv ikke nok. Det viktigste for studentene var et godt undervisningsopplegg og BlueJ bør være en del av dette.

Det mest interessante resultatet fra det temaet som tok for seg den obligatoriske oppgaven var den kjønnsmessige fordelingen i tilnærmingen til oppgaven. Alle mennene hadde en programmeringsteknisk tilnærming, mens kvinnene hadde en virkelighetsnær/objektorientert tilnærming.

Referanser

1. Lewis and Lofthus: Java Software Solutions. Addison-Wesley. ISBN 0-201-78129-8.
2. Barnes and Kölling: Objects First with Java. Prentice Hall. ISBN-0-13-044929-6.



Høgskolen i Buskerud
Postboks 235
3603 Kongsberg
Telefon: 32 86 95 00
Telefaks: 32 86 98 83
www.hibu.no

ISSN 0807-4488

