



OFFLINE NR.3 · 2011







Offline er et tidsskrift for linjeforeningen Online.

REDAKTØR:

Lorents Odin Gravås  
lorentso@online.ntnu.no

MARKED:

René Olavi Räisänen  
raisanen@online.ntnu.no

ØKONOMI:

John-Erik Johansen

FORSIDEFOTO:

Kai Torgeir Dragland

GRAFISK PROFIL:

Beate Hay Sandmo

LAYOUT:

Beate Hay Sandmo  
René Olavi Räisänen  
Lorents Odin Gravås  
Beate Baier Biribakken  
Rikard Eide  
Susanne Gustavsen  
Tina Juni Størø  
Sverre Johann Bjørke

TRYKK:

Øien & Indergaard  
Opplag: 425

KONTAKT:

Redaksjonen, proKom  
redaksjonen@online.ntnu.no

Sem Sælunds vei 7-9  
7034 TRONDHEIM

<http://online.ntnu.no>



## // REDAKTØRENS Nullindeksering

Du leser nå historiens andre (tredje) Offline, hurra!

FOTO: KAI TORGEIR DRAGLAND

De som var på Online sitt nullindekserte 25-årsjubileum fikk antageligvis nok av den typen vitser iløpet av kvelden. Vi frisker derfor opp alle de andre minnene dine fra jubileumet, med en jubileumsspesial uten overnevnte vits.

Vi har gjort et forsøk på å oppsummere Online sin mystiske historie så langt, og det er godt å se at det en gang så hullete lappeteppe som er Online sin historie nå begynner å holde bedre på varmen.

### Fra hånd til munn

Våre kapitalistiske venner i bedKom har sørget for at vi nå har en ordentlig bærekraftig økonomi. For å gi bedKom nye utfordringer svarer vi i redaksjonen med å øke både sidetallet og opplaget betraktelig, og kan nå by på 36 sider (!) kvalitetsstoff og et nesten dobbelt opplag.

### Espen utforsker virkeligheten

En høst har nesten gått siden forrige utgave, og masterstudentene har vært på den tradisjonelle IT-ekskursjonen for å besøke bedrifter. Vår Espen var der såklart utelukkende

for å dekke dette. Silicon Valley er også et sted som huser mange bedrifter relevant for informatikere, så vi sendte like greit Espen dit også.

### Alternativt pensum

Vi i Offline tar ansvar. Med en høst som snart er omme følger eksamenslesing. Men siden du tydeligvis leser dette i stedet, har vi ikke glemt å lure inn litt stoff som kan være eksamensrelevant: Vi tar utgavens pattern til et nytt nivå ved å ta for oss fem prinsipper som gjør programvareutvikling til noe mer enn en sort magi: SOLID. Sist gang i vår serie om hacking var temaet Cross-Site Scripting (XSS), og det er derfor naturlig å denne gangen ta for seg en veldig relatert svakhet innen webprogrammering: Cross-site request forgery (CSRF).

Når du til slutt sitter der på eksamen og skrivekrampa inntar begge hender, vit at Offline også da tenker på deg. Online sin kampanje for PC på programmeringseksamen har allerede fått blæst både i Under Dusken og andre medier, og vår egen Beate Baier Biribakken har stått i spissen for dette. I denne utgaven av Offline utdyper hun saken ytterligere.

### 05 Busbuddy

Offline har snakket med teamet bak suksessen Busbuddy

### 08 PC på eksamen

Drakampen for å få tillatelse for PC på eksamen er i gang

### 11 Servermiljøer

Christian gir oss et kræsjkurs i hvordan serverrommet skal se ut

### 13 Silicon Valley

Herr Olsen dro til IT-vuggen og besøkte garasjer

### 15 Mysteriet Online

Nullindeksering og mer måtte til når vi gravde i Onlines historie

### 18 Jubileet i bilder

Online feiret et kvart århundre med et brak. Vi har bildebevisene på at det skjedde!

### 22 Mastertur til Oslo

Den årlige it-ekskursjonen gikk av stabelen tilbake i september

### 25 Pattern: SOLID

Dag Olav forklarer oss prinsippene bak SOLID

### 30 Gogogadget: Kindle

Amazon har lansert en ny kindle, vi tar en liten kikk på hva som gjør den verdt pengene

### 32 Hacking: CSRF

Dag Erik følger opp serien med en beslektet metode til forrige blads XSS

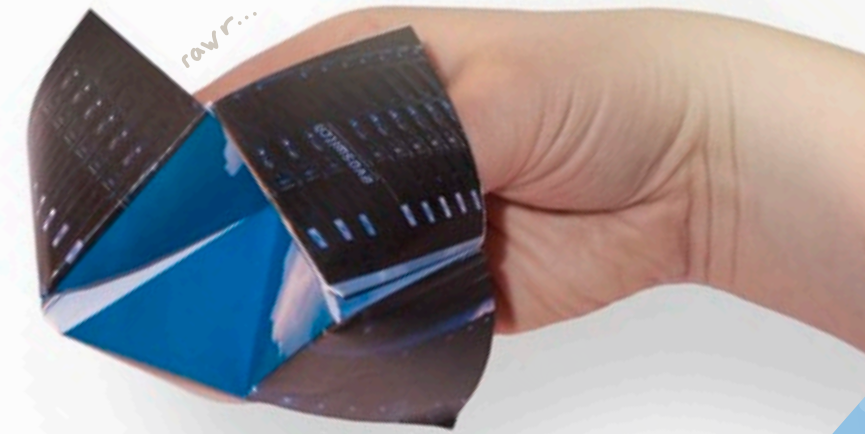
### 34 Nerdeoppskrift

Hvor mange nerder trengs det for å lage middag?

### 36 Refresh

Vi har funnet frem litt oppdateringer fra NTNU

s.11 >



s.18

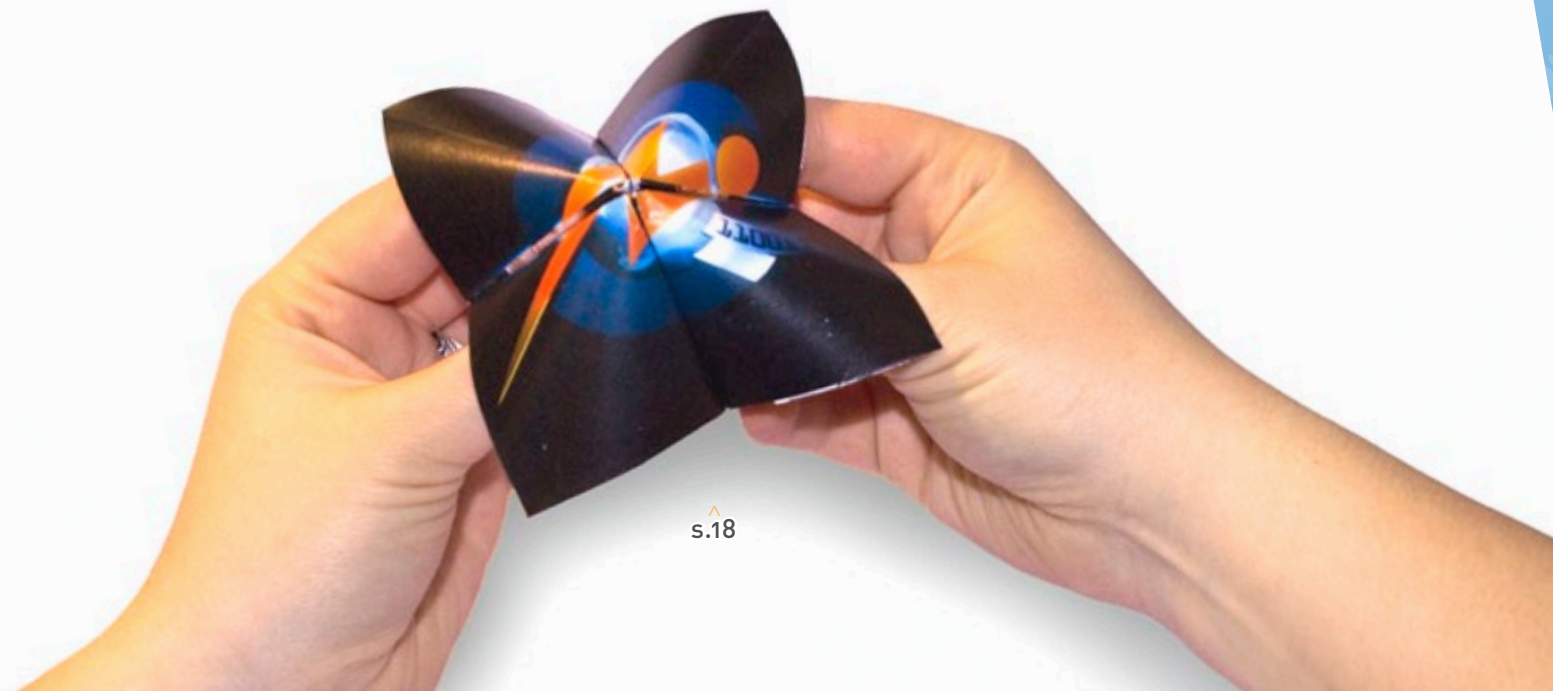






FOTO: KAI TORGEIR DRAGLAND

Håper du også fikk oppleve Onlines første jubileum. Vi avholdt nylig linjeforeningens 25-årsjubileum med kjempesuksess. Hovedstyret er stolte over all den innsatsen våre frivillige la ned i denne festen. Dette er linjeforeningens første jubileum. Måtte det bli det første i en rekke av mange, mange jubileer i Online!

I forbindelse med vårt første jubileum oppdaget vi tidlig mangelen på historisk informasjon om Online. Jubileumskomiteen (jubKom) og Hovedstyret har gravd og undersøkt i lang tid for å finne gode, nevneverdige historier fra vår fortid. Omsider fikk vi kontakt med Onlines første formann, Ole A. Brevik. Han satte i gang innsamling av historisk materiale sammen med noen av de første Onlinerne. Etter kort tid fikk vi tilsendt en historiesamling som i detalj beskriver hendelser fra de første årene linjeforeningen eksisterte. Blant dokumentene i denne skattekisten av historier var referatet fra årsmøtet i 1987, der «Terje Rydland sa seg villeg til å gi ei orientering om datafag-studier». Historien kan leses i sin helhet på Online sin etterhvert omfattende wiki.

Når man graver etter historie oppdager man fort hvor enkelt vi har det i dag. Det skal ikke mer til enn et Googlesøk og en tur innom Wikipedia før man finner mer informasjon enn man kunne drømme om. Dessverre gjelder dette kun informasjon som faktisk ligger på åpne nettsider - ikke informasjon som er gjemt vekk på ødelagte harddisker, på gamle tapemaskiner på Dragvoll, i Google Docs-samlingen til utgåtte styremedlemmer og så videre.

En av de beste kildene til historie og informasjon er møtereferater. Disse referatene gemmer seg ofte på noen av de ovennevnte kildene. Arkivene av referater i Online har tradisjonelt bestått av samlinger med .PDF-filer som er tidkrevende å gå gjennom. Denne tradisjonen måtte snus!

## // STYREORD

Mens Online feirer 25 år åpner Hovedstyret sine hittil skjulte referater som detaljerer linjeforeningens utvikling.

## ONLINES INDRE VIRKE

Den 28. september vedtok Hovedstyret å slutte med LaTeX-maler og .PDF-dokumenter. I stedet skulle vi legge innkallinger og referater på HS-wikien. Referatene som ligger der er enkle å gå gjennom, de er søkbare, - og viktigst av alt: De ligger på Hovedstyrets wiki der fremtidige styremedlemmer enkelt kan finne dem.

På det samme møtet tok vi opp en sak om åpenhet. Vi hadde nå et godt arkivsystem for fremtiden, men hva med den nåværende generasjonen? Noen komiteemedlemmer har ytret et ønske om å få vite mer av hva som foregår i linjeforeningen. Disse medlemmene fikk alt de ba om idet Hovedstyret kopierte ut referatene sine til den åpne wikien. Her kan alle som måtte være interessert åpne våre referater og finne ut hvilke jern Online har i ilden om dagen. Personinformasjon, bedriftsdiskusjoner og overraskelser er riktignok fjernet fra de åpne referatene.

Det nærmer seg semesterslutt og juleribbe. Over nyttår skal Hovedstyret sørge for at informasjon Online har i diverse Google Docs-samlinger skal flyttes over til wikien. I tillegg skal vi sette i gang arbeidet med å lage en hoveddekskursjon for 3. klasse. Det første trinnet av informatikere som vil oppleve denne turen er de som nå går i 2. klasse. Er du i 2. klasse og vil være med på planleggingen av turen? Ta kontakt!

God lesning!

Med vennlig hilsen

Leder, Linjeforeningen Online  
leder@online.ntnu.no

AtB satt på sanntidsdata, men et åpent API eksisterte ikke. Gjengen bak busbuddy.no tok til slutt oppgaven i egne hender.



# Slapp sanntidsdataene fri



TEKST: LORENTS ODIN GRAVÅS  
FOTO: RENÉ OLAVI RÄISÄNEN

Helt siden AtB tok over de store bussrutene i byen har ivrige applikasjonsutviklere siklet etter tilgang til sanntidssystemet. Mange har tatt kontakt med AtB og spurt om tilgang, men hos AtB har det vært lite svar å få. Uten overnaturlige evner var det lenge umulig å vite om det var vits i å løpe til bussen.

Dette skulle endre seg drastisk den dagen ryktene om at det fantes en applikasjon som, på mystisk vis, hadde fått tilgang til sanntidsdataene til AtB, nådde informatikeren Roy Sindre Norangshol.

- Jeg dro rett hjem fra deltidsjobben for å teste denne applikasjonen. Jeg fikk da bekreftet at her var det sanntidsdata – dette var veldig spennende!

Roy Sindre har stått i spissen for prosjektet busbuddy.no, som per dags dato leverer et enkelt, fritt og åpent JSON-basert API for AtB sitt sanntidssystem. Det man stusser litt over er hvordan Busbuddy kunne tilby dette lenge før AtB selv bestemte seg for å slippe disse dataene fri.

### Stjålet tjenestenøkkel

Gjengen bak Busbuddy-prosjektet ble nemlig lei av å vente på AtB, og bestemte seg heller for

å levere dataene selv. Denne éne applikasjonen med tilgang skulle vise seg å være nøkkelen som åpnet døren for en flom av applikasjoner og webtjenester som i dag benytter sanntidsdataene fra AtB.

Det sa seg selv at applikasjonen måtte få dataene levert fra et sted. Ved hjelp av et såkalt *man in the middle*-angrep loggførte Roy Sindre all trafikk fra- og til telefonen sin, og kunne dermed avsløre hvor applikasjonen fikk sanntidsdataene fra.

- Det tok ikke lang tid før det tikket inn data. Vi fant videre ut hvor applikasjonen gjorde kall mot, og at dette var en SOAP-tjeneste som kjører hos AtB på et domene vi aldri hadde hørt om, forteller han ivrig.

### Orario di partenza?

Når man har tilgang til dataen som går over nettverket er det stort sett en enkel sak å finne ut hvordan en applikasjon autentiserer seg, og dermed kunne utgi seg for å være den applikasjonen. På dette tidspunktet hadde Roy Sindre samlet en liten gjeng interesserte via IRC, og ved å utgi seg for å være appen kunne de utforske denne tjenesten videre.

Det som skulle gjøre oppgaven litt ekstra interessant var at AtB sitt API slettes ikke var på engelsk eller norsk. Utviklerene hadde heller latt seg inspirere av Andrea Bocelli og navngitt alle metoder på italiensk.

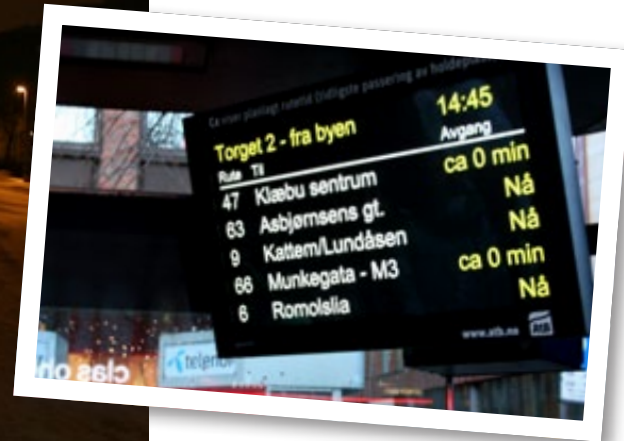
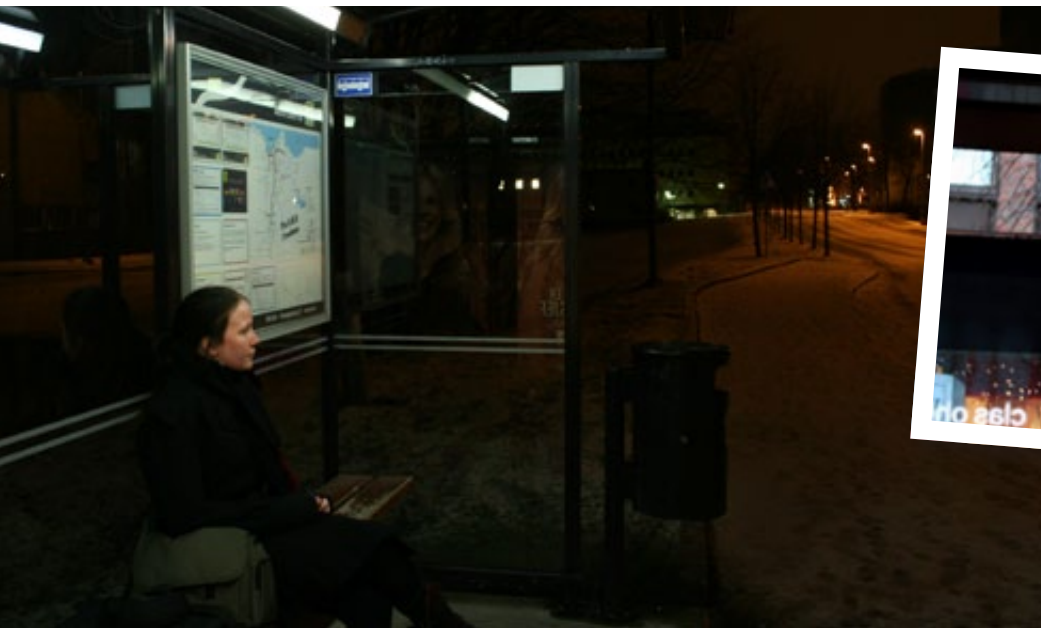
### Et enklere API

Det tok derfor ikke lang tid å finne ut at AtB sitt API ikke var noe å jobbe med:

- Det første som slo oss litt var at det hadde litt rar oppbygging. Og ikke minst at det var på italiensk. Interessen vår var dessuten å lage en mobilapplikasjon, og å programmere mot SOAP-grensesnitt for mobiltelefon er et herkuten like. Det var ønskelig å levere et mye enklere format, forteller Roy Sindre.

Dette førte til at gjengen utviklet en tjeneste som omstrukturerte og oversatte sanntidsdataene, og leverte alt i form av lettvektig JSON. Den første applikasjonen som tok i bruk API-et var en applikasjon for iPhone, utviklet av eks-informatikeren Magnus Romnes samme kveld. Bare dager senere fulgte Roy Sindre selv med en applikasjon for Android.





**BOMMET:** Med sanntidssystemet er det stort sett bare ens egen arbeidslyst som gjør at man bommer på bussen.

### Sanntidsdata til folket

Etter noen uker og mye finpusning ble tjenesten gjort fritt tilgjengelig. De som ønsker det kan nå få sin egen API-nøkkel ved å følge instruksjonene på [api.busbuddy.no](http://api.busbuddy.no). Kildekoden for selve tjenesten ble også gjort open source og er nå tilgjengelig på Github. Roy sindre forklarer:

– Selvfølgelig kunne jeg vært dust og sagt at ingen andre får tilgang, men min applikasjon er jo langt fra den beste. Den fungerer og gjør arbeidet sitt, men den er bare et hobbyprosjekt. Det er alltid noen andre som er mer kreative, har mer tid, og kanskje lager en bedre løsning. Det er jo poenget med at slik data burde være fri – det er alltid kreative sjeler der ute som kan lage noe genialt.

### Mottagelsen

Roy Sindre og Magnus ble langt fra de eneste brukerne av API-et; Alf's bybuss, Bartebuss og mange andre fulgte etter. Fra starten av hadde tjenesten opp mot 500 kall per dag, og dette steg senere til et gjennomsnitt på 2500 per dag. Tallene gikk riktig nok litt ned igjen etter applikasjonen Bartebuss nylig gikk over til en annen løsning.

### AtB reagerer

Selv om intensjonene var gode, utleverte busbuddy data som ikke var offentlig, uten eierens viten eller samtykke, ved hjelp av en stjålet API-nøkkel. Det var likevel ikke busbuddy.no som vekket reaksjoner hos AtB, men heller Magnus sin iPhone-applikasjon. Like etter

den ble sluppet tok AtB kontakt med Magnus via en blogg, med beskjed om å ta kontakt snarest. Magnus sendte svar hvor han forklarte at han bare forholdt seg til tjenesten levert av busbuddy, mens Roy Sindre sendte en offisiell mail til AtB fra busbuddy:

– Jeg sendte en oppfølging og presenterte oss og formålet vårt, at vi hadde ingen onde intensjoner og at vi håpet de ikke skulle stenge oss ute.

Roy sindre fikk ikke noe svar, og spekulerer i om AtB i det hele tatt forsto hva dette var snakk om i utgangspunktet.

*«Det er alltid kreative sjeler der ute som kan lage noe genialt»*

### Utestengelsen

Til slutt ble API-nøkkelen som busbuddy så lenge hadde misbrukt bannlyst fra AtB sitt system. Men da hadde allerede gjengen bak busbuddy.no funnet en annen vei inn i systemet, og tjenesten forble uberørt.

– Etter at vi hadde begynt å grave litt i systemene deres, fant vi et annet, helt ubeskyttet, API-kall som ga de samme dataene. AtB kunne kanskje trengt en sikkerhetskonsulent, smiler Roy Sindre lurt.

### AtB åpner opp

Magnus fikk videre svar med klar beskjed om at han brukte data som ikke var offentlig, men

at AtB hadde planer om å legge dem ut senere. Like før den offentlige lanseringen fikk til og med Roy Sindre svar på mailen sin, med beskjed om at det sannelig var på tide at busbuddy fikk sin egen API-nøkkel.

– Jeg tror de så det litt an og forsto det positive i det vi gjorde. I senere tid stod det jo også på plakaten på bussene at du kunne «laste ned en applikasjon for din telefon» forteller Roy Sindre.

### Fortsetter på Busbuddy

I dag er det mulig for alle å sende en e-post til AtB og skaffe seg tilgang til det offisielle API-et og dermed sanntidsdata. Roy Sindre vil likevel anbefale at folk bruker og hjelper til på busbuddy-prosjektet, da det tilbyr et API litt mer behagelig å jobbe med enn et inkonsistent, italiensk, SOAP-grensesnitt.

– Jeg tror egentlig ikke systemet til AtB er helt ferdig selv om de har gjort det åpent, og kjappgangen der kan kanskje ha hatt noe med det vi gjorde i sommer



## En sommer utenom det vanlige

Kristina Nessel Kjerstad intervjuet en våre Summer Interns, 22-årige Dag Einar Monsen, som studerer datateknikk på tredje året ved NTNU. Denne sommeren tilbragte han imidlertid et helt annet sted enn Trondheim – nemlig Opera Softwares kontor i Tokyo.

### Hva jobbet du med hos Opera Software?

Den første uka jobbet jeg med *Open The Web*. Arbeidet gikk ut på å kontrollere at Operas nettleser fungerer som den skal på forskjellige kjente nettsteder med PC, mobil og tablets, og så videre. Dersom det ikke fungerte normalt, undersøkte jeg grunnene til dette. Etterhvert startet jeg på det som jeg jobbet med resten av sommeren, nemlig å utvikle innhold til den nylig annonserte *Opera TV Store*. I hovedsak gikk det ut på å modifisere widgets som allerede var utviklet for Opera Desktop og få dem til å fungere på TV.

### Hadde du noen spesielt interessante prosjekt mens du jobbet hos Opera?

Det høres kanskje litt diplomatisk ut, men jeg synes ærlig talt at alt jeg jobbet med var svært interessant. Det var kanskje litt mer spennende å samarbeide om å lage en ny widget fra scratch enn å modifisere de som allerede var utviklet. Da fikk jeg også muligheten til å lære litt om git og litt mer om avansert javascripting.

### Hvorfor valgte du å reise til Tokyo, og hvordan opplevde du det å jobbe der?

Alternativene jeg fikk var Tokyo, Beijing, Polen, Sverige og Oslo. For meg sto det mellom Tokyo og Beijing, og valget falt egentlig ganske fort på Tokyo. Jeg følte at Japan var mer

ulikt Norge, og jeg var ute etter å få nye inntrykk; og det fikk jeg. Noe som overrasket meg spesielt positivt var maten. Jeg var jo innstilt på at jeg kom til å innta en del spesielle retter, men bortsett fra noen veldig seige kamskjell likte jeg alt jeg ble servert svært godt! For øvrig var det jo helt utrolig mange mennesker der. Alt var bare enormt. Tokyo er så ufattelig mye større enn man klarer å forestille seg.

### Hvordan opplevde du Opera som arbeidsplass? Noen overraskelser?

Jeg hadde jo på forhånd hørt litt om bedriftskulturen i Japan, med dresskode og 16-timers arbeidsdager. Det var en positiv overraskelse å se at det ikke var sånn hos Opera. Det var et veldig behagelig arbeidsmiljø hos Opera, og de opererer med flexitid så man kan styre dagen litt på egenhånd.

### Hva tenker du om fremtidige jobbmuligheter og ønsker?

Jeg er absolutt ikke fremmed for en karriere i Opera. Etter denne sommeren har jeg fått litt mersmak for å reise, og med muligheter som Operas *Bridge The World* er jo det veldig realiserbart. Å få jobbe i San Mateo i Silicon Valley eller Melbourne, Australia hadde absolutt ikke vært dumt.

### SØK NÅ!

<http://www.opera.com/company/jobs/internship/>

### LES MER!

<http://my.opera.com/internshipatopera/blog/>



# Lei av å kode med penn og papir?



**Dagen har kommet, vi skal testes. Metodene sitter inne, hendene kjenner hver eneste tastetrykk, hver eneste snarvei. Tiden er inne: Jeg er klar for å snakke datamaskinens språk. Og dét skal gjøres på papir.**

TEKST: BEATE BAIER BIRIBAKKEN  
FOTO: ERIK LOTHE

Det er ikke dette undervisningen har innstilt meg på. Dette samsvarer ikke med hvordan jeg tenker og arbeider. I dagesvis har jeg måttet forberede meg, ikke i så stor grad på å lære stoffet, men på å kode på papir. Når dagen kommer vil jeg sitte og knote: Skulle ønske jeg kunne smette inn en metode her, bytte ut variabelnavnet der, rydde siden der alt ser ut som en gedigen krusedull. Hvorfor kan vi ikke bare ta eksamen på pc? Den tanken var jeg ikke den første som tenkte.

Den 5. mai i år vedtok Studenttinget ved NTNU at alle studenter som tar skriftlig eksamen skal få muligheten til å ta denne på pc. Ved valget til Studenttinget påfølgende semester var det også mange av kandidatene som satte dette som sin hjertesak.

## Et sted å begynne

Undertegnede ble først involvert i denne saken på årets generalforsamling, hvor Michael Johansen la frem sine kampsaker for kommende studieår. En av hjertesakene hans var muligheten til å kunne ta i bruk pc på eksamen uten å måtte søke om spesiell tilretteleggelse for det.

Etter sommeren som fulgte traff jeg Michael på Onlinekontoret, hvor vi kom inn på diskusjonen om digital eksamen. Noen burde gjøre noe. Michael gav meg en epostadresse og den informasjonen han allerede hadde gravet frem. Så var det i gang.

## Onlines underskriftskampanje

Det var et stort skritt mot fremtiden og datamaskin på eksamen da Studenttinget vedtok at studentene skulle ha muligheten til å kunne ta sin skriftlige eksamen på pc. Men selv om mange av studentene så behovet for pc på eksamen, var det noen av dem som mente at det å innføre pc i alle fag ville bli for ambisiøst dersom det skulle være noe håp om å få gjennomslag for saken. Dette var bakgrunnen for at undertegnede, med mye hjelp og støtte fra Online, den 21. oktober lanserte en underskriftskampanje som fremmet et krav om pc på programmeringseksamen.

## I rampelyset

I forrige utgave av Under Dusken fikk underskriftskampanjen sin første mediadekning. Den 16. november holdt også Studenttinget, i samarbeid med Studentparlamentene ved UiT, UiO og UiB, sin kampanje «Pc på eksamen», og diskusjonen fikk ytterligere fokus i media.

Dette førte til at flere universitet og høyskoler tok kontakt med meg slik at deres studenter også kunne delta i underskriftskampanjen.

## Hvorfor programmeringseksamen?

For studenter som ikke studerer it-relaterte emner kan det kanskje virke urettferdig at muligheten til å bruke pc på eksamen innføres eksklusivt i programmeringsfag. Dette er ikke helt tilfeldig: I programmering handler alt om struktur. Skriver du en eksamen i norsk eller noe tilsvarende vil teksten fremdeles gi mening om to avsnitt har byttet plass, men i programmering kan det føre til at hele koden er ubrukelig.

Det er ikke til å legge skjul på at det endelige målet med underskriftskampanjen er at alle studenter skal få muligheten til å bruke pc på samtlige eksamener, men det er urealistisk å tro at den ordningen vil bli innført for 20.000 studenter med det første. Dette betyr ikke at det er umulig, men at «alle skal få» - det er et vanskelig løfte å holde.

Pc er allerede godt integrert i samfunnet, men akkurat eksamenssituasjonen, den skiller seg ut. Det bør ringe noen bjeller blant utdanningsinstitusjonene når studenter på landsbasis krever å få bruke pc som skriveverktøy på eksamen for å føle seg rettferdig vurdert.

De fleste utdanningsinstitusjoner er enige i at pc på eksamen er fremtiden, likevel tviholder de på penn og papir. Dette begrunner de med høy risiko for juks og for mange praktiske utfordringer knyttet til pc på eksamen for alle studenter.

## Utfordringer

Selv om pc på eksamen er en drømmesituasjon for mange, er det naivt å tro at det kan gjennomføres uten modifikasjoner. Derfor er det bra, og ikke minst forståelig, at studenter også har møtt kampanjen med skepsis og trukket frem utfordringer som er knyttet til bruk av pc på eksamen.

Som det ble nevnt i intervjuet i UD, har det ennå ikke blitt gjort klart en løsning for hvordan man kan gjennomføre en eksamen på pc innen høyere utdanning. Noen av utfordringene som nevnes er antallet studenter, kostnader, mangel på lokaler, manglende strømtilførsel, datatap og sikkerhetsspørsmål. Selv etter å ha hørt lovnader fra mange hold er det fremdeles et spørsmål som for mange står ubesvart: Hvordan i all verden er dette mulig å gjennomføre i praksis?

## Skreddersydd operativsystem

Per i dag er det to norske universitet som har prøveordninger med pc på eksamen: Universitetet i Agder (UiA) og Universitetet i Bergen (UiB). På UiA stiller universitetet med tynnklienter slik at studentene ikke trenger å ha med pc selv. På UiB kan studentene stille med egne datamaskiner og får tildelt minnepinner med et operativsystem (OS) som overkjører det OS-et som allerede finnes på studentenes pc.

Dette OS-et booter direkte fra minnepinnen uten behov for installering og skal verken gi tilgang på internett, egne filer eller avansert programvare - kun en enkel tekstbehandler. Den sistnevnte modellen er sannsynligvis den best fungerende løsningen, i hovedsak på grunn av antallet studenter.

Underskriftskampanjen har fokusert på den sistnevnte modellen, da det ikke foreligger et ønske om endringer i selve eksamensgjennomføringen - man skal fortsatt kunne gjennomføre eksamen på papir dersom man ønsker det. Studentene fortjener en mulighet til å rydde opp i koden, bytte om - og legge til nye variabler og metoder fortløpende. Systemet som er forespeilet vil mest sannsynlig være en Linux-distribusjon og må kunne fungere på alle typer datamaskiner. Dette systemet er ikke laget ennå, men er teknisk sett fullt mulig å gjennomføre.

## Bak kulissene

Selv om undertegnede har engasjert seg mye i saken, har også Onlines Drift- og utviklingskomité (dotKom) bidratt mye ved å utvikle et skreddersydd, generisk system for underskriftskampanjer, som blant annet er blitt brukt til å føre denne kampanjen. Underskriftskampanjen for pc på programmeringseksamen finnes på: [online.ntnu.no/petition](http://online.ntnu.no/petition)



## Klar for nye utfordringer?

Nets er et av Nord-Europas ledende selskap innen betalingsløsninger, informasjonsformidling og digitale sikkerhetsløsninger, med en av de mest omfattende og innovative produktporteføljer i Europa. Selskapet er resultatet av en sammenslåing mellom BBS, Teller og PBS i Danmark.

Nets har 2.200 ansatte, har hovedkontor i København samt kompetanse- og forretningsentre i Oslo, Stockholm, Helsinki og Tallin.

- ✓ Opptil 20 % av arbeidstiden til selvinitiert innovasjon og læring.
- ✓ Systemer med stor samfunnsmessig betydning.
- ✓ Over 80 utviklere med fokus på åpne løsninger.

### Nets Academy

- program for nyutdannede utviklere

- ✓ Rullering mellom områder
- ✓ Sosiale turer
- ✓ Dedikert mentor
- ✓ Besøke nordiske kontorer
- ✓ Søknadsfrist 1. april 2012 på [nets.eu](http://nets.eu)
- ✓ Oppstart 1. september 2012

### Kontaktpersoner:

- ✓ Fredrik Andersen | HR | Tlf. 908 21 574
- ✓ Thomas Andre Lund | IT | Tlf. 997 10 161



De fleste har hørt om serverrom og datasentere med kabelklyser, blinkende lys, gigantiske kjøleanlegg, surrende disker og høytgående vifter. Men hva skjuler seg egentlig bak dørene på rack-skapene?

# Servermiljøer – en introduksjon

TEKST: CHRISTIAN STRAND YOUNG  
FOTO: STOCKFOTO

En server, eller en tjener, har som formål å levere ressurser eller tjenester til klienter. Dette kan være alt fra et filområde til en nettside som én eller flere vil ha tak i. En annen server kan også være en klient, uten at vi går nærmere inn på akkurat det.

En fysisk server på et enterprise-nivå er som regel i formen *rack*. Dette er et standardisert format på rundt 48cm i bredden. En boks kan dog ha variabel høyde som måles i *enheter* eller *unit* – der en U er 4.5cm. Vanlige servere er som regel på 1 til 2U. Større bokser som f.eks. filservere kan strekke seg til over 10U. Et standard skap huser som regel mellom 40-50U.

Hver enkelt boks monteres inn i skapet ved hjelp av skinner festet på sidene av boksen, og blir deretter skrudd fast i skapet. Dette gjør det enkelt å dra ut én enkelt boks for å gjøre vedlikehold, uten å demontere hele skapet.

Så, hvorfor er disse serverene store og flate

bokser med rare standardiseringer? Kostnad er et sentralt punkt her. Ikke nødvendigvis kostnad i form av penger, men plass. Det er mye mer effektivt å plassere 40 bokser oppå hverandre i høyden, enn å ha mange ulike kabinetter på rad og rekke. Forestill deg 100 pc-er stablet oppå hverandre på et 30kvm stort rom. Problemet er ikke å finne frem til boksen, men heller å forflytte seg i rommet uten å rive ned noe.

### Mer enn bare sære bokser

Det er likevel ikke utseendet som gjør servere spennende. Innmaten i en server skiller seg betydelig fra en vanlig hjemme-pc. Blant annet er serverhardware vesentlig dyrere. Hvorfor er det slik? Servere trenger ikke å kjøre det nyeste COD- spillet på maks grafikk, men heller levere sikre og raske tjenester. Grafikkortet er integrert og byttet ut til fordel for mer CPU-kraft og flere minnespor. Mange servere har gjerne mer enn én CPU som deler minnet. Raidkontrollere og flere strømforsyninger er heller ikke ukjent, men dette kommer vi tilbake til senere.

### Pålitelige systemer

Essensen i serverhardware er at det er laget for å holde. Det skal kunne stå og surre i flere år, under belastning og uten pauser. Byggekvantiteten og materialvalget er derfor ofte av en dyrere sort. I et servermiljø er det også nulltoleranse for feil. En bitfeil kan ikke oppstå når onkel i Amerika skal sende deg 3 millioner i nettbanken. ECC-RAM – Error Correcting Code, er noe man sjelden finner i en hjemme-pc, men de fleste servere spiser faktisk ikke noe annet. CPU-er i en server har stort sett rikelig med cache og kjerner til å ta seg av all last som er forventet. Når man da putter flere CPU-er sammen med et delt, raskt og (til dels) feilfritt minne, har du oppskriften på kraft.

IO-tunge tjenester som for eksempel mail krever ytelse også lengre ned i lagerhierarkiet. Serverdisker (SCSI/SAS f.eks.) er også forskjellige fra vanlige disker. SAS-disker kjører over et SATA-interface og er gjerne mer grundig testet fra fabrikk. Følgelig har de lavere feilprosent. SAS-disker kan spinne nesten tre ganger så raskt som tradisjonelle disker.



## Redundans

Som nevnt tidligere er det lite rom for feil. Om en strømforsyning går dukken, eller en kurs mister strøm, kan ikke systemet gå ned av den grunn. Når man lager en server vil man derfor ha flere lag redundans gjennom hardwaren. Kritiske bokser har derfor gjerne to eller flere strømforsyninger på forskjellige kurser som tar over dersom noe feiler. Det er også vanlig å beskytte seg videre mot strømbrudd i form av aggregater og UPS (Uninterruptible Power Supply).

Tap av data er noe av det verste som kan skje. Redundans på permanente lagringsmedier er derfor uhyre viktig. Vanligvis sikres dette på to måter. RAID (Redundant Array of Independent Disks) og backup. RAID I (speiling) eller RAID 5/6 (paritetsdata) er de vanligste måtene å sikre seg mot en disk-krasj. Eksempel: Du har to like diskene og setter opp RAID I. All data du skriver til disk blir da speilet likt på begge diskene. Om den ene disken skulle feile, tar den andre over. RAID kan også brukes til å forbedre ytelse (RAID 0 - striping). Her sprer man data over flere disk slik at man lese i parallell.

## I verste fall

Backup er også et must. Om en utvikler klarer å ødelegge et skript, skrive over en database (jada, det skjer) eller noe lignende bør man ha muligheten til å få tilbake data. Har man plassproblemer kan inkrementell backup være en fin løsning. Her tar man én full backup, for å så bare ta backup av nye filer og filer som er endret. Som storskala systemadministrator bør du også ta høyde for at det umulige kan skje. Noen kan fysisk stjele diskene dine eller brenne ned serverrommet ditt. Da er det lurt å ha kopier på en helt annen lokasjon. Dette blir vanligvis gjort med tape-kassetter man flytter fysisk, eller over et datanettverk som internett.

**PROTIP:** Et software-raid yter som regel bedre enn en billig hardware raid-kontroller.

## Clusterfuck

Når flere datamaskiner samarbeider om en oppgave kalles dette et cluster. Et cluster ser ofte ut som en enkelt maskin utenfra, men bak kulissene fordeles arbeidet mellom flere maskiner. Det finnes grovt sett tre grunner til å ha et cluster:

- » Du skal gjøre tunge beregninger.
- » Du har store mengder data.
- » Du ønsker å unngå nedetid eller datatap.

På NTNU har vi et eget tungregneanlegg. Her jobber massevis av datamaskiner sammen for å gjøre ressurskrevende beregninger. Værmeldingen er et eksempel på et problem som er for stort til en enkelt datamaskin, og krever et tungregnecluster.

Søkemotorer har en indeks som skal romme hele Internett, og dette tar en god del plass. For at søkemotoren skal kunne ha god nok ytelse er man avhengig av at alt får plass i RAM, så man slipper å lese og skrive til disk hele tiden. For å få til dette er man avhengig av å fordele indeksen over flere maskiner, slik at hver maskin bare får en andel av indeksen som får plass i RAM. Når du søker etter et ord, sendes forespørselen til den maskinen som har den delen av indeksen med ditt søkeord i.

For en stor søkemotor er det imidlertid ikke nok å bare dele opp indeksen. Hver enkelt del av indeksen får også så mange forespørsler

at en enkelt maskin ikke kan håndtere alle. Derfor dupliseres alle delene av indeksen over flere maskiner, slik at hver maskin ikke får flere forespørsler enn de tåler.

En annen grunn til å gjøre duplisering er for å bedre tilgjengeligheten. For eksempel kan man ha en tjeneste som kun krever en enkelt maskin for å fungere, men er avhengig av 100% oppetid. Da kan man lage et cluster av to eller flere maskiner, slik at tjenesten fortsetter å svare selv om en maskin går ned. For at et dupliseringscluster skal kunne brukes til å bedre oppetiden er det viktig at man har nok ekstra kraft til å gjøre jobben selv om en av maskinene i clusteret forsvinner.

## Safety on board

En av de største utfordringene en systemadministrator har, er sikkerhet. Har du sikkerhets-hull kan uautoriserte få adgang til systemene dine, og det vil du helst ikke. På tampen skal jeg nå gi et par kjappe tips.

Tjenester du ikke skal bruke, trenger du heller ikke. Fjern dem! Du bør også sikre hver enkelt tjeneste istedet for å stole på at brannmuren tar alt. Sikkerheten er aldri sterkere enn det svakeste leddet. Windowsmaskina ingen har rørt på tre år kan potensielt gi adgang til resten av boksene dine. Softwareoppdateringer er et must! Sist, men ikke minst, aldri gi brukerne dine mer tilgang enn de faktisk trenger.

Silicon Valley, rett utenfor San Francisco, er kjent som IT-bransjens midtpunkt, vugge og fødested. Det virket derfor som et naturlig sted for Offline å utforske. Undertegnede tok på seg oppdraget og reiste til Silicon Valley.

## OFFLINE UTFORSKER

# SILICON VALLEY

TEKST OG FOTO: ESPEN SKARSBØ OLSEN

Noe av det første som slår en når man kommer til Silicon Valley er garasjene. I disse fantastiske garasjene har flere større bedrifter startet sin ferd mot storhet. Bill Hewlett og Dave Packard startet HP i en garasje i Palo Alto i 1939, Steve Wozniak og Steve Jobs startet Apple i en garasje i Mountain View i 1975 og Larry Page og Sergey Brin startet Google i en garasje i Palo Alto. Felles for alle var de lå i Silicon Valley, og har i ettertid blitt store suksessbedrifter.

## En dinosaur i alder og størrelse

Mitt første bedriftsbesøk i Silicon Valley er hos IBM Almaden. Dette er en forskningslab som IBM har liggende i åssiden med utsikt over Silicon Valley. IBM Almaden er fødestedet til blant annet harddisken,

SQL og relasjonsdatabaser og det var spesielt å gå rundt i gangene der så mye av grunnlaget for moderne datamaskiner var blitt lagt. IBM startet i IT-bransjen med elektroniske hullkortmaskiner i 1890-årene og er dermed hva vi kan kalle en dinosaur innen databransjen, men langt ifra død. IBM har i dag over 550 000 ansatte verden over, og er fortsatt i front når det gjelder fremtidsforskning på IT. Da vi var på besøk fikk vi en innføring i IBMs fokus på å etterligne den energieffektive prosessoren vi bruker hver dag, hjernen, for å energieffektivisere datamaskiner. Hjernen har ca 1680 Ghz regnekapasitet, og bruker bare 20-40 watt i timen. Etter presentasjonen fikk vi en liten omvisning der vi blant annet fikk se verdens første harddisk.

## En katalysator for Silicon Valley

Stanford University ligger midt i Silicon Valley, har cirka 15 000 studenter, en campus som er

halvannen gang større enn Trondheim sentrum og har bidratt til oppstarten av over 3000 bedrifter, som HP, Cisco, VMware, Yahoo!, Google, Nike, Logitech og Sun. Stanford har vært en essensiell bidragsyter når det gjelder å skape kultur for at studenter skal starte bedrifter i løpet av studietiden, eller når de er ferdige. På Stanford har man en forventning om at alle studenter skal prøve å starte sin egen bedrift, og spør ikke studentene «Hvor skal du jobbe når du er ferdig?», men «Hva skal du grunnlegge når du er ferdig?». Har du startet en bedrift, og feilet slik som mange gjør, så har du fortsatt en god sjanse på å få jobb i et av de etablerte firmaene. Det er en kultur som aksepterer at man prøver og feiler. På Stanford fikk vi en god innføring i hvordan de tilbyr fjernundervisning i samarbeid med Sverige og Singapore, en omvisning på campus der alle byggene stod i stil med hverandre, man hadde sitt eget busselskap som





gikk gratis rundt på campus og en innføring i noe av den nyeste forskningen på Stanford. Vi fikk blant annet sett en virtuell virkelighetslab som forsket på hvordan representasjon i en virtuell virkelighet kunne påvirke personligheten i virkeligheten og et fungerende distribuert sosialt nettverk.

#### Google – Ikke kontorer, men campus

Google har vokst fra to personer i en garasje i Palo Alto til en bedrift med 10 000 ansatte i Mountain View i løpet av 15 år. I over førti forskjellige bygninger på Googleplex sitter de ansatte. Der har de fri tilgang til over tjue forskjellige kafeer og restauranter, med frukt, godteri og brus. På campus finnes det også fasiliteter for fritidsaktiviteter som svømmebasseng og volleyballbaner. Tjenester som lege og fysioterapeut er alltid tilgjengelig og en frisør er innom ukentlig.

Offline var på besøk på Googleween, deres Halloweenfest, og campus var fylt med neder som var kledd ut som alt fra Nyan Cat til Waldo til personer i Top Gun-uniformer.

Google tror på en flat struktur i firmaet sitt, og derfor har Larry Paige og Sergey Brin

et åpent møte hver fredag klokken 15:30 der alle de ansatte kan spørre dem om hva som helst, fra hvilken retning firmaet beveger seg i, til hvorfor det ikke er Mountain Dew utenfor kontor x.

Vi fikk en dag med foredrag om sikkerheten rundt deres skyløsninger, og fikk en påminnelse om at skyteknologi ikke er nytt, med referanse til Hotmail, og en liten video over hvor sikre datasentrene deres var. I tillegg fikk vi et foredrag fra Android-sjefen om hvordan de jobbet med flaggskipet sitt, Nexus-serien, og fikk en liten intro til Galaxy Nexus. Etter foredragene fikk vi en rask omvisning i de fire bygningene som opprinnelig hadde plass til hele Google, men i dag kun er fire av over førti bygninger. Lokalene var åpne, lyse, fulle av farger og det var tilgang på mineralvann, frukt og snacks på nesten hvert hjørne.

#### Happy Hour!

Det siste stoppet Offline rakk denne runden i Silicon Valley var Facebook. Det er et ganske ungt firma, som leier sine lokaler bare noen få hundre meter fra campusen til Stanford. Når man kommer til bygningen er det

ingenting som tilsier at Facebook har kontorer der, før man ser «H A C K» skrevet på dørene til bygget. Offline var invitert til Facebook av en tidligere NTNU-student som hadde fått jobb der for fire uker siden. Han gav oss en omvisning i kontorene som bar sterkt preg av at det var et lagerbygg. Ledningene som hang i taket var synlige, og det var et totalt åpent landskap med små stillerom langs veggene som kunne brukes ved behov. Selv Mark Zuckerberg satt på en vanlig pult midt i kontorlandskapet. Hver fredag avsluttes arbeidsdagen klokken 17:00 med Happy Hour bestående øl, vin og snackmat i kantina. Her er det likt for alle, og Offline observerte at Mark måtte stå i kø som alle andre.

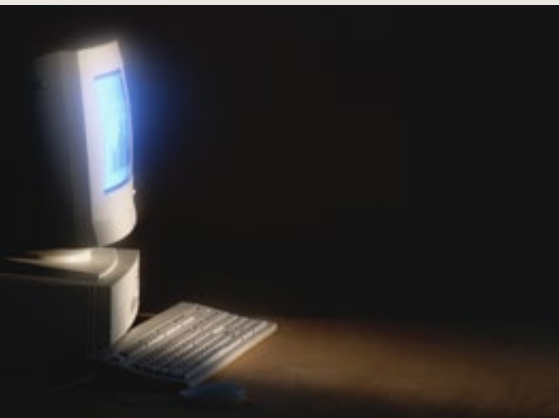
Etter en tur på byen og en natt senere var den alt for korte turen til Silicon Valley over, og Offline måtte dessverre dra hjem. Vi skal komme oss tilbake!

Det har vært mange myter spunnet rundt Onlines fortid. Vi har rett og slett ikke visst hvor vi kommer fra, hvem vi egentlig er. Det er på tide å kaste lys over identiteten vår.



**MYSTERIET ONLINE**





*I denne meget primitive fortiden snakket man ennå om at «nå er Norge kommet i dataalderen»*

OLE A. BREVIK

File name: ONLINE.AA7;9  
Document title: Oversikt over datafag-studiet  
Author initials: LAA  
Last edited: 5-NOV-87

Informatikkstudiet ved AVH (Den allmenntvitenskapelige høyskolen).

- DA 1 Grunnkurs i EDB og programmering
- DA 2 Videregående programmering
- DA 3 EDB i skolen
- DA 5 Grafisk databehandling
- DA 6 Datamaskinens virkemåte
- DA 7 Datastøttet læring
- DA 8 Sanntidsprogrammering og prosessilknytning
- DA 9 Systemering I
- DA 10A Innføring i FORTRAN
- DA 10B Innføring i COBOL
- DA 11 Systemering II
- DA 12 Nyere programmeringsverktøy
- DA 13 Billedbehandling
- DA 14 Ekspertsystemer
- DA 15 Innføring i PROLOG
- DA 16 Basis programvare i datamaskinsystemer
- DA 17 Billedanalyse
- DA 18 Semesteroppgave i informasjonsteknologi



MS-DOS 1.0  
Lanseres i 1982



Mac System Software 1.0  
Lanseres i 1984

## 1985

**on-Line stiftes**  
Gorbatsjov snakker om glasnost og perestrojka, Bobbysocks vinner Melodi Grand Prix og on-line, linjeforeningen for informatikk, blir stiftet ved AVH på Lade.



Jubileet 12.11.11: Ole Brevik, linjeforeningens første leder, viser når on-line ble opprettet.

## 1986:

Michael Johansen, vår nåværende leder, blir født

## 1987

**Sosialkomiteen opprettes**  
Blir etter hvert til dagens arrKom

TEKST: KATHRINE STEFFENSEN

FOTO: KAI T. DRAGLAND, STOCKFOTO, ONLINES ARKIV

ILLUSTRASJON: BEATE HAY SANDMO

I forbindelse med det vi trodde var vårt 25-årsjubileum bestemte Jubileumskomiteen at de skulle ta et dykk i arkivet for å se hva de kunne finne av historie. Dette hadde de jobbet med siden 2009. Da vår nåværende leder, Michael Johansen, endelig kom i kontakt med Onlines første leder, Ole André Brevik, skulle vi lære mye om Onlines spede barndom. Brevik kunne blant annet opplyse oss om at i 1986 var feil årstall for stiftelsen av Online. Linjeforeningen ble nemlig stiftet året før, i 1985, og 25-årsjubileet hadde gått oss hus forbi. Vi forsto da at det var nødvendig å få et bedre grep om historien vår.

### Stiftelsen

I begynnelsen, da datafag (oldtidens informatikk) var en relativt ny samling fag, var studiet en del av Den Allmenntvitenskapelige Høgskole (AVH) på Lade. Brevik forteller at noen av studentene på datafag følte et

sterkt behov for å bedre miljøet for datafagstudentene. Det hadde seg nemlig slik at regjeringen bevilget penger til datafagstudiet på AVH, og resultatet ble en styrtflom av studenter på Lade.

*Da satt vi der, et år etterpå, og innså at nå har vi jo nesten tatt alle fagene som fantes. Hva skal vi gjøre nå?*

Et av argumentene for å starte en linjeforening var å skape et samlingspunkt for de mange studentene. Det faglige miljøet trengte også en oppvikker, forteller Brevik. Den gang fantes det ikke noe rent datafagstudium. Derfor var det mye fokus på å etablere et hovedstudium i informatikk, slik at fagmiljøet kunne bli tettere og større. I talen sin på jubileet sa han: «Da satt vi der, et år etterpå, og innså at nå har vi jo nesten tatt alle fagene som fantes. Hva skal vi gjøre nå?»

## 1992:

The Gathering arrangeres for første gang

## 1993:



## 1997:

Begrepet «googling» er ennå ukjent, men dette forandrer seg

## 1997-98

**Bedriftskontakten opprettes**  
Kjent som bedKom fra oktober 2004



**Terje Rydland**  
Allerede på 90-tallet en entusiastisk mac-bruker

## 90-tallet



**Ballmer-peak**  
Microsofts myteomspunne teori blir testet ut ved AVH

### De tidligste årene

Det nystiftede on-line avholdt en spørreundersøkelse blant datafagstudentene for å kartlegge faglig interesse for blant annet hovedfag i informatikk. Denne ble senere brukt for å vise at det var behov for et informatikk-institutt. On-line var tidlig ute med forslag for ny fagplan da institutt for informatikk endelig ble etablert to år etter linjeforeningen, høsten 1987. On-line jobbet stadig videre for at informatikk skulle kunne tilby studier fra grunnleggende nivå til doktorgrad.

Det var ikke bare på en faglig arena at on-line var frempå: Senere ble Ladebekken volleyballklubb etablert uavhengig av AVHI

(nå NTNUI). Sosialkomiteen ble også opprettet i 1987 for å sikre gode sosiale vekstvilkår for studentene på informatikk. Denne komiteen ble til det vi i dag kjenner som Arrangementskomiteen, eller arrKom.

### Hverdagen på Lade

On-line fikk ikke eget kontor før på 90-tallet, så studentene måtte finne andre steder å være under daglig sosialisering. Dette foregikk i all hovedsak på stud.ass-kontoret, men det foregikk også overnattinger på mikrolabben eller terminalrommet for de aller ivrigste medlemmene. Av faste arrangementer som ble startet opp på denne tiden, og som fremdeles lever den dag i dag, kan vi nevne Blåtur

og x-fest. Den gang ble x-festen arrangert sammen med Delta, linjeforeningen for matematikk og fysikk, og ikke Abakus som det er i dag. On-line arrangerte for øvrig mange fester sammen med Delta, ettersom de også hadde tilholdssted på Lade.

### Et lite hull i historien

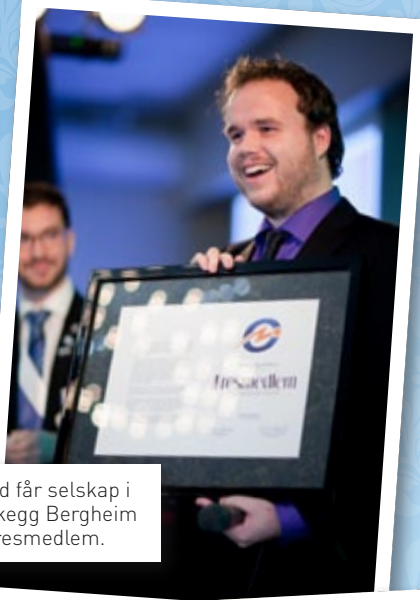
Nå har det seg slik at Brevik ikke gikk på informatikk for alltid, trass i utvidet fagtilbud. Derfor måtte vi etter hvert finne nye kilder til informasjon. Første halvdel av 90-tallet (92-95) er oppsøksvekkende lite opplyst. Et diplom fra Toddygaloppen i 1993 vitner likevel om vår eksistens på denne tiden.

**Arvid Holme**  
Blar i sine sagnomsuste foiler



Online feiret sine tjuefem år 12. november på Prinsen Hotell.

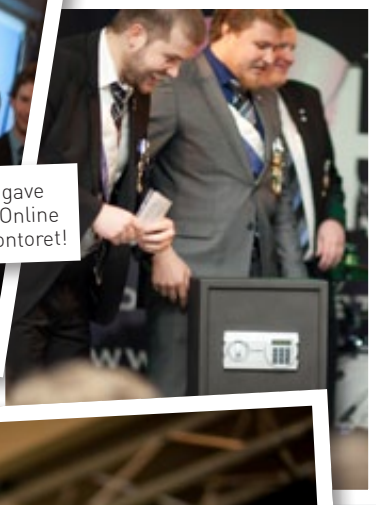
FOTO: Kai Torgeir Dragland



Steinar og Arvid får selskap i det Erlend Klakegg Bergheim utnevnes til æresmedlem.



Overrekkelse av gave fra Eldsterådet. Online fikk ny safe til kontoret!



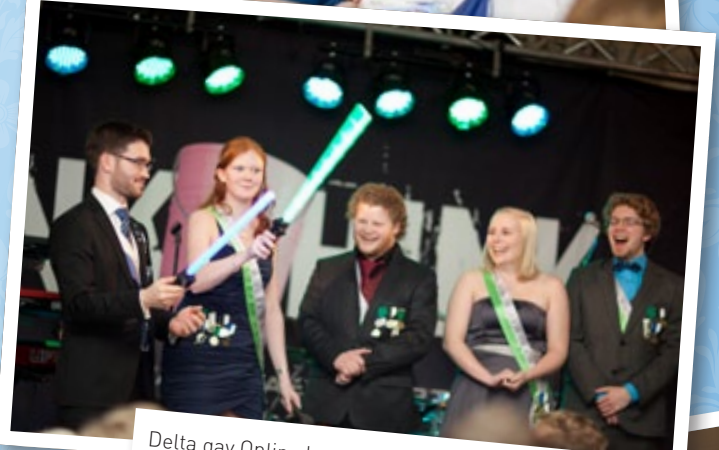
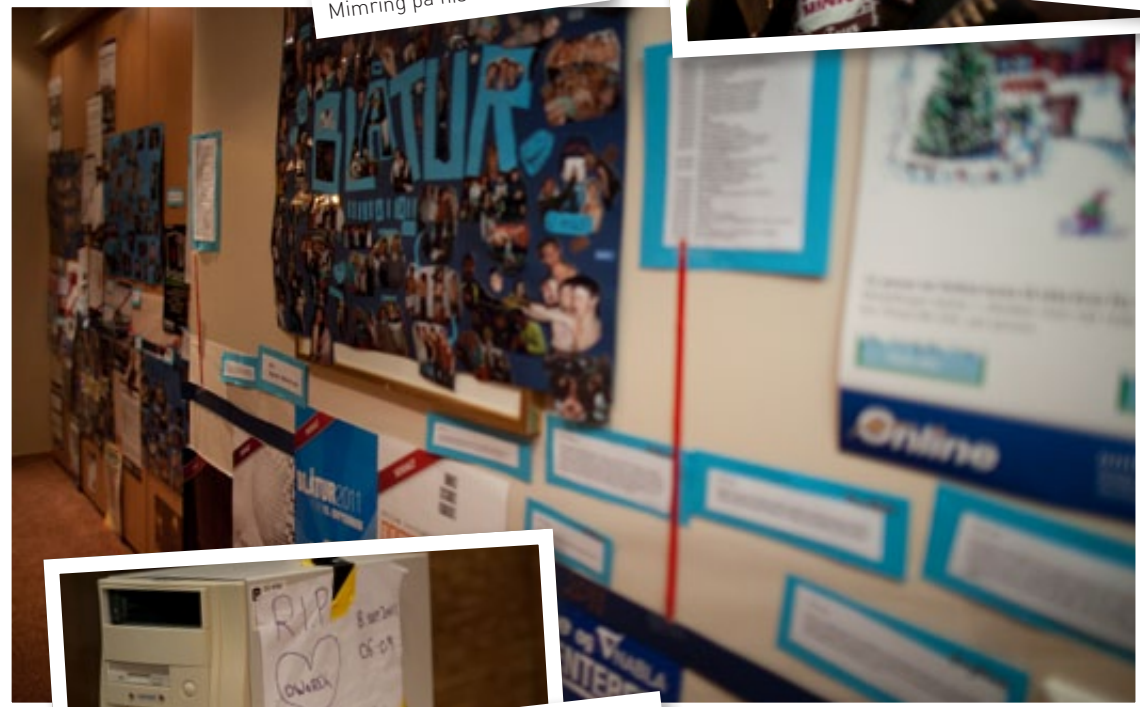
# Subileet i bilder



Mimring på historierommet

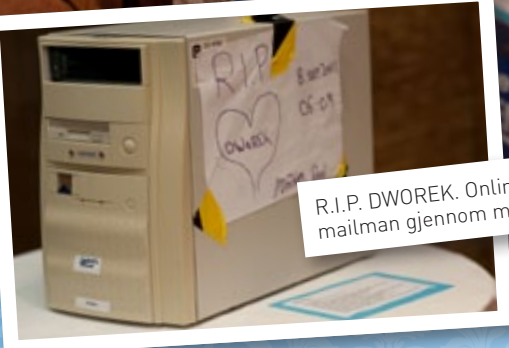


Pink Hink kom helt fra Åre for å spille.



Delta gav Online lasersverd i 25-årgave. Peow-peow!

Disco-Erik viser gamle kunster.



R.I.P. DWOREK. Onlines mailman gjennom mange år.



Gaveoverrekkelse fra Smørekoppen





2000

**.com-bobla sprekker**  
Informatikk mister halvparten av studentene over natten



2002

**Velkommen til kontoret!**  
Online får kapret kontor på det nybygde P15



2005:  
YouTube lanseres og viral marketing er et faktum

2006

**fagKom opprettes**  
Onlinere får innblikk i spennende teknologi og lærer nye ting utenfor forelesningene



2011

**proKom vedtas på Genfors**  
Brått får Online snasne plakater og sitt eget tidsskrift

2001

**Kjelleren åpner**  
Online finner kjemien med Volvox

I 1995 er det mer informasjon å oppdrive, som for eksempel at det ble arrangert både fest på studenterhytta og vaffelsteking for medlemmene. På denne tiden var internett blitt et mer allment fenomen, og det var på denne tiden Telenor stjal det som burde vært Onlines domene.

IT-bransjen var i oppsving, og Online fikk mer kontakt med bedrifter som ønsket å samarbeide med linjeforeningen. Komiteen Bedrift ble derfor opprettet i 97-98 for å sikre et godt samarbeid mellom studenter og fremtidige arbeidsgivere. Som en slags forløper til it-ekskursjonen reiste informatikkstudentene på tur til Oracle Software i 1997. Dessverre brøt bussen sammen halvveis, og kun tre studenter som hadde kjørt bil kom frem.

### Gløshaugen neste

Etter hvert som Online nærmet seg tusenårs-skiftet skjedde det store omveltninger, særlig på grunn av dot com-boblen som sprakk like etter årtusenskiftet. Dette resulterte i at informatikk mistet en stor andel av studentene sine, og linjeforeningen kun satt igjen med et styre. Årene som fulgte bar mye preg av omorganisering. Den største grunnen til dette var

2004

**dotKom opprettes**  
Online får sine egne kodeaper

at informatikkstudentene flyttet fra Lade til IT-syd på Gløshaugen. Det var mye oppstyr rundt innflyttingen, særlig fordi IT-syd var nedslitt og led av plassmangel. Etter hvert som tiden gikk stabiliserte forholdene seg, og Online fikk på nytt komiteer og et hovedstyre. Som et fullverdig medlem av Gløshaugen var det på sin plass at Online fikk en kjeller på Moholt, og i januar 2000 fikk vi tildelt Realfagskjelleren sammen med Volvox og Alkymisten. Kjelleren ble offisielt åpnet 16. mai 2001 med brask og bram, og senere ble det også Delta og Spanskrøret være med.

### Velkommen til kontoret!

I 2002 sto et nytt bygg med datasaler ferdig på parkeringsplass 15. Etter et par år på slitne

IT-syd ønsket Online seg et nytt sted å være, og vi søkte dermed om kontorplass. Det var vi ikke alene om. Også Abakus var interesserte i kontoret. Takket være en tidlig søknad var det likevel Online som kunne flytte inn i tredje etasje på P15, og kontoret er i dag åsted for mye kaffedrikking og generell kos.

Årene som fulgte bar preg av mange sosiale arrangementer, bedriftspresentasjoner og flere komiteer. I 2004 ble Drifts- og utviklingskomiteen (dotKom) startet opp for å holde styr på det tekniske, blant annet drift av hjemmesiden. To år senere ble det klart at det var behov for å opprette en komité som kunne arrangere kurs. Dette var i kjølvannet av «Google speaker series», som hadde holdt foredrag i samarbeid med Online og

2010

**økonomiKom vedtas på Genfors**  
Online får som første forening på haugen en hel komité som skal ta hånd om økonomien

Abakus. Resultatet ble fagKom, som i dag har ansvaret for å «stimulere studentenes faglige interesser» (fagKoms wiki). I praksis vil dette si å arrangere kurs i regi av bedrifter eller andre aktører.

### Veien videre

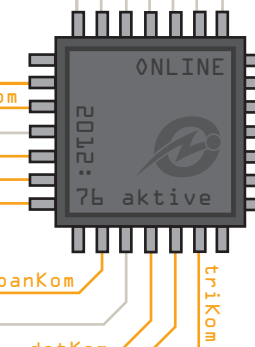
Lille Online har blitt voksen. Økonomien har blitt større, medlemstallet på medlemmer generelt og i komiteer har økt, og vi har markert oss som en seriøs aktør på mange felt her på Gløshaugen. I dag har Online hele 76 personer i komiteer og styre, det høyeste antallet noensinne.

For ikke lenge siden, i 2010, ble proGruppen opprettet for å møte et økende behov for plakater og medaljer for diverse arrangementer.

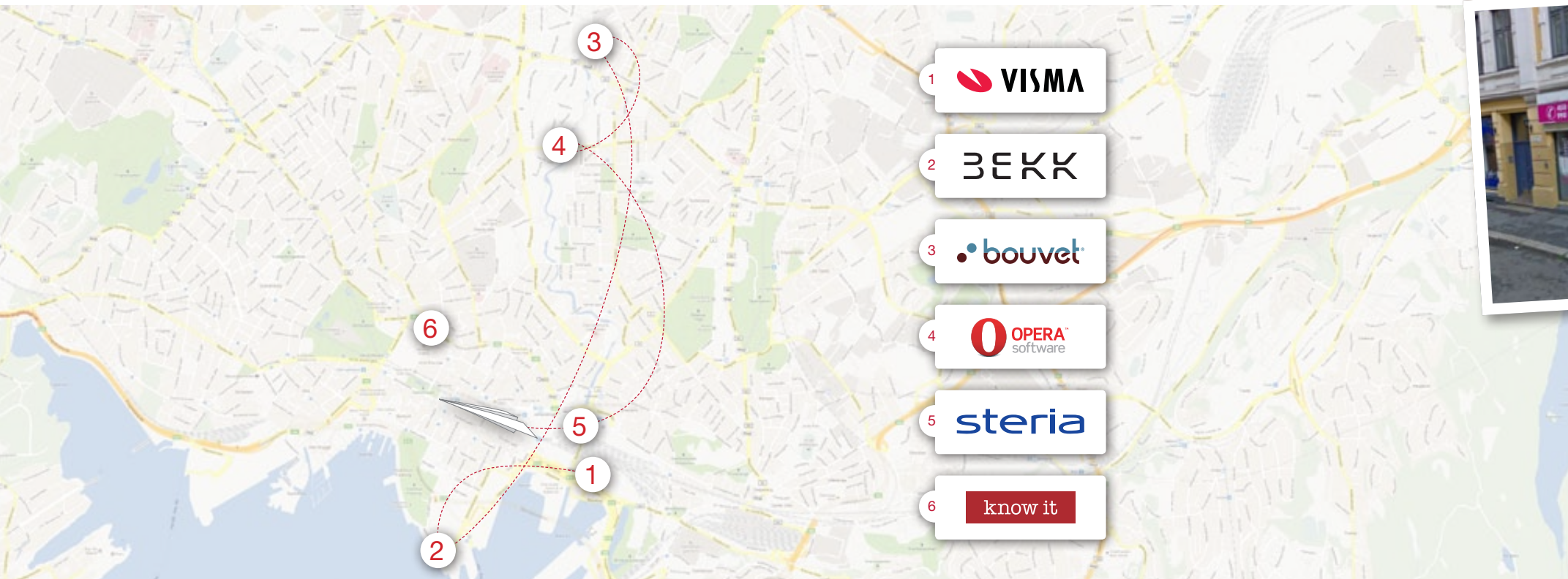
ProGruppen ble vedtekstsfestet mai 2011, og skiftet da navn til Profil- og aviskomiteen, proKom. Ikke lenge etter oppstarten sørget proKom også for at Online endelig kunne skilte med et eget tidsskrift. Høsten 2011 kom det siste tilskuddet til komité-samlingen, nemlig Trivselskomiteen (triKom). De har ansvaret for å gjøre det enda triveligere på kontoret og generelt gjøre hverdagen hyggeligere for informatikere. Det skal nok en gang arrangeres vaffelsteking i regi av Online, for å nevne noe.

2011

**triKom opprettes**  
Kontorsjefen får seg undersätter







**Linjeforeningsleiar med fleire talent:**  
Vår kjære leiar, Michael, også kjent som MrClean på IRC har visst starta seg eit renseri i Oslos gater

## BTW BedKom besøker BEDRIFTENE

**Online er no halvveis til femti (pluss litt til) og det vert berre viktigare med fokus på kvalitet. BedKom tok turen til Oslo for å sørge for ein god dialog med industrien.**

TEKST OG FOTO: RENÉ OLAVI RÄISÄNEN

Som ein oppfylgjar til suksessen i fjor valgte vi i bedKom å ta ein tur til Oslo for å pleie det gode samarbeidet vi har med nokre bedrifter, samt skape ein open og triveleg dialog med potensielt nye samarbeidspartnarar.

Morgenkaffi, lunsjkaffi, møtekaffi og middagskaffi. I løpet av dei tre dagane vi var i Oslo fekk magen køyrd seg. Dei to timane vi hadde berekna per bedrift viste seg å vere høgst overkalkulert. For so vidt ein god ting, men vi gjekk noko lei av kaffi i lengda.

### Høgt rekrutteringspress

Hovudfokuset for turen var å høyre kva industrien har behov for og undersøke om vi kan eller bør forbetre det vi tilbyr. Eit av temaa som gjekk igjen var dette med eit vanvittig høgt press på rekruttering og stadig forskyving av datoar. Dette er noko vi har diskutert litt i bedKom allereie og vi trur det lønar seg, både for studentar og bedrifter, å roe ned litt. Kanskje kan det vere ynskjeleg med ein intensjonsavtale for felles søknadsfrister?

Blant dei nye bedriftene vi møtte var det litt interesse for å høyre om linjeforeningskulturen som eit heile. NTNU stiller jo ekstra sterkt i denne kategorien. Vi la rivalisering og kniving til side og promoterte kulturen so godt det lot seg gjere og tipsa om andre linjeforeningar der dette følte naturleg. Ærlighet varar lengst – er det ikkje det ordtaket sei?

Online var små ein gang i tida og vi har lært mykje frå dei som har gått stien før oss og håpar at vi også kan vere til hjelp for andre i framtida.

### Opplæring av nye

Utover å diskutere dei produkta vi sel er det også ein god måte å lære opp nye medlem i bedKom. Som i fjor reiste vi sørover med ein gammel og ein ny. Vilde som er ny av året var ganske stille dei tre fyrste møta, men tok nesten heilt over på dei siste møta. To intensive dagar med bedriftsmøter gjorde visst susen.

Dagane var som nemnt ikkje særleg stressande. Vi fekk god tid til å vandre innom diverse kaffibarar og butikkar; og i jakta på ein rimeleg lunsj fann vi jammen også eit renseri med same namn som linjeforeningsleiaren vår, MrClean.

### Verdien av eit ansikt

Som ein del av bedKom er ein stadig i kontakt med bedriftsrepresentantar og sender eit utal med epostar fram og tilbake. Ein sak eg har lagt meg på minnet gjennom mine år i bedKom er kor verdifullt det er å ha møtt kontaktpersonen sin ansikt til ansikt. Det gjer det veldig mykje enklare å føre ein fornuftig dialog og eg følar meg tryggare i stillinga mi.

Eit besøk rundt om i Trondheim er under planlegging, og kanskje blir det også ein ny oslotur på vårparten.

# ITEx: Mastertur til Oslo

**Arbeidsmarkedet for informatikere er utrolig godt om dagen, og i den sammenheng drar masterstudentene på en ukes tur til Oslo hver høst for å besøke bedrifter.**

TEKST: ESPEN SKARSBØ OLSEN  
ILLUSTRASJON: RENÉ OLAVI RÄISÄNEN

I september var vi trettini spente studenter som la ut sammen på en fire-dagers tur til Oslo. Ekskursjonen ble som tidligere år dekket av bedriftene som ønsket besøk; både reise, hotell og daglige måltider var inkludert. Hver bedrift hadde satt av god tid til å si litt om seg selv og mingle sammen med oss i etterkant for å gjøre seg litt bedre kjent.

### Ankomstdagen

Turen startet hos Visma Sirius. Der fikk vi informasjon om bedriften, og konkurrerte i å produsere papirfly som skulle traversere en graf med ulike papirfly som kunne forsere hver sin vekta kant (sic). Vi måtte bruke Scrum som utviklingsmetode, og kjørte to sprinter. Etter at en vinner var kåret, fikk alle som ønsket det, intervju for sommerjobb eller fast jobb. To personer endte opp

med tilbud i hånden samme dag og Visma ordnet tapas som bespising og tok oss med på en tur på byen senere.

### B-dagen

Neste dag startet vi med å besøke et av Norges mest attraktive IT-selskap, Bekk, hvor vi fikk en innføring i hva de driver med og en omvisning rundt i bygget. Forvaltnings- og Management Consulting-avdelingen tok godt i mot oss og forklarte litt om hva de jobbet med og presenterte et spennende utviklingsprosjekt for Windows Mobile. Besøket ble avsluttet med et kurs i en alternativ måte på inputvalidering og lunsj sammen med Bekk.

Bouvet sto så for tur. De startet med å presentere noen av kundene sine, der de viste oss de respektive kunde-prosjektene og fulgte opp med en innføring i Bouvet sitt daglige virke. Det ble arrangert speedintervju og ordnet med pizza for alle. Kvelden benyttet vi, i kjent studentstil, for å utforske «tigerstadens» fantastiske uteliv.



BOUVET: Studentene følger nøye med under bedriftspresentasjonen

### En tur til Opera og Steria

Dag 3 startet med et besøk hos et av Norges beste teknologiselskaper, Opera Software. De fylte besøket sitt med foredrag om alt fra markedsføring til et kurs om HTML5 og CSS3 av Håkon Wium Lie. Etter en runde med tapas til lunsj avsluttet Opera med en omvisning i serverrommet før vi dro videre.

Konsulentbransjen er utrolig aktive innen rekruttering om dagen, og etter Opera, var ekskursjonens femte konsulentbedrift

klare for å ta i mot oss – Steria. Som de fleste store IT-selskapene har de mange konsulenter engasjert i NAV-prosjektet og en av disse ga oss en innføring i pensjon i tillegg til den vanlige bedriftspresentasjonen. Utover kvelden fikk vi spise og mingle med Steria i deres lokaler og vi fikk muligheten til å nyte utsikten fra taket på postgirobygget.

### Avreisedagen

Know IT var den siste bedriften vi besøkte. Som sjettemann ut skiller de seg fra de andre med at de fleste som jobber «inhouse» meste-parten av tiden, mens man i andre selskaper ofte er stasjonert hos kunden.

Know IT hadde, som de andre, en presentasjon av seg selv og en omvisning der vi fikk se hva folk jobbet med, for eksempel en 3D-printer noen av dem hadde laget for fremvisning på JavaZone. Vi fikk også et foredrag om optimalisering av heisalgoritmer, som dessverre ikke ville ha fungert på heisene på PI5, da algoritmen krever fire heiser (og gjerne fungerende, red. anm.)

Etter fire dager og seks bedrifter var turen over, og vi vendte snuten nordover. Noen har signert jobb, flere har fått tilbud. IT-ekskursjonen er et flott tiltak for oss studenter.

Takk til Visma, Bekk, Bouvet, Opera, Steria og Know IT for en fantastisk tur!



Kjapp i toppen  
og løs i snippen!

Vi er på jakt etter løse snipper som mener at verden fremdeles har til gode å se de beste løsningene innen informasjons- og kommunikasjonsteknologi.

Hvis du synes et uformelt miljø med høyt faglig fokus, kort vei til sjefen og kultur for entreprenørskap høres ut som en drøm, vil vi svært gjerne at du sender en mail til [hei@knowit.no](mailto:hei@knowit.no).

*KnowIT er et av Skandinavias ledende miljøer innen informasjons- og kommunikasjonsteknologi. Vi jobber med ledende merkevarer innen offentlig og privat sektor, og tar langsiktig ansvar for våre kunders verdiskapende prosesser.*

[knowit.no](http://knowit.no)

know it

## UTGAVENS PATTERN

Kjennskap til forskjellige design patterns er nyttig og kan spare oss for mye arbeid i det lange løp. Noen patterns er så abstrakte at de gjerne overskrider grenser til å bli filosofier eller prinsipper.

# S.O.L.I.D.

I denne utgaven skal vi ta for oss SOLID-prinsippene som nesten må sies å være obligatorisk kunnskap for enhver programmerer med ønske om å produsere god kode og systemer som er enkle å vedlikeholde og utvide.

TEKST: DAG OLAV PRESTEGÅRDEN

ILLUSTRASJON: LORENTS ODIN GRAVÅS M.FL.

SOLID er et akronym bestående av forbokstavene til fem viktige prinsipper for programmering av god objektorientert kode. Bak de ulike bokstavene ligger prinsippene «Single Responsibility Principle», «Open/Closed Principle», «Liskov Substitution Principle», «Interface Segregation Principle» og «Dependency Inversion Principle».

Brukt sammen vil disse prinsippene gjøre at du skriver kode som er lettere å utvide, lettere å vedlikeholde og som er mindre utsatt for såkalte «code smells» - symptomer i koden som tyder på dypere designproblemer i kodebasen. Disse kan være duplisert kode, metoder og klasser som er for omfattende eller biter av kode som benytter seg av implementasjonsdetaljer i klasser de helst ikke bør vite for mye om. Om det finnes mange mer eller mindre

vonde dufter i kodebasen eller du bare ønsker å vite mer om god design, burde du lese videre. La oss ta for oss alle prinsippene, ett for ett:

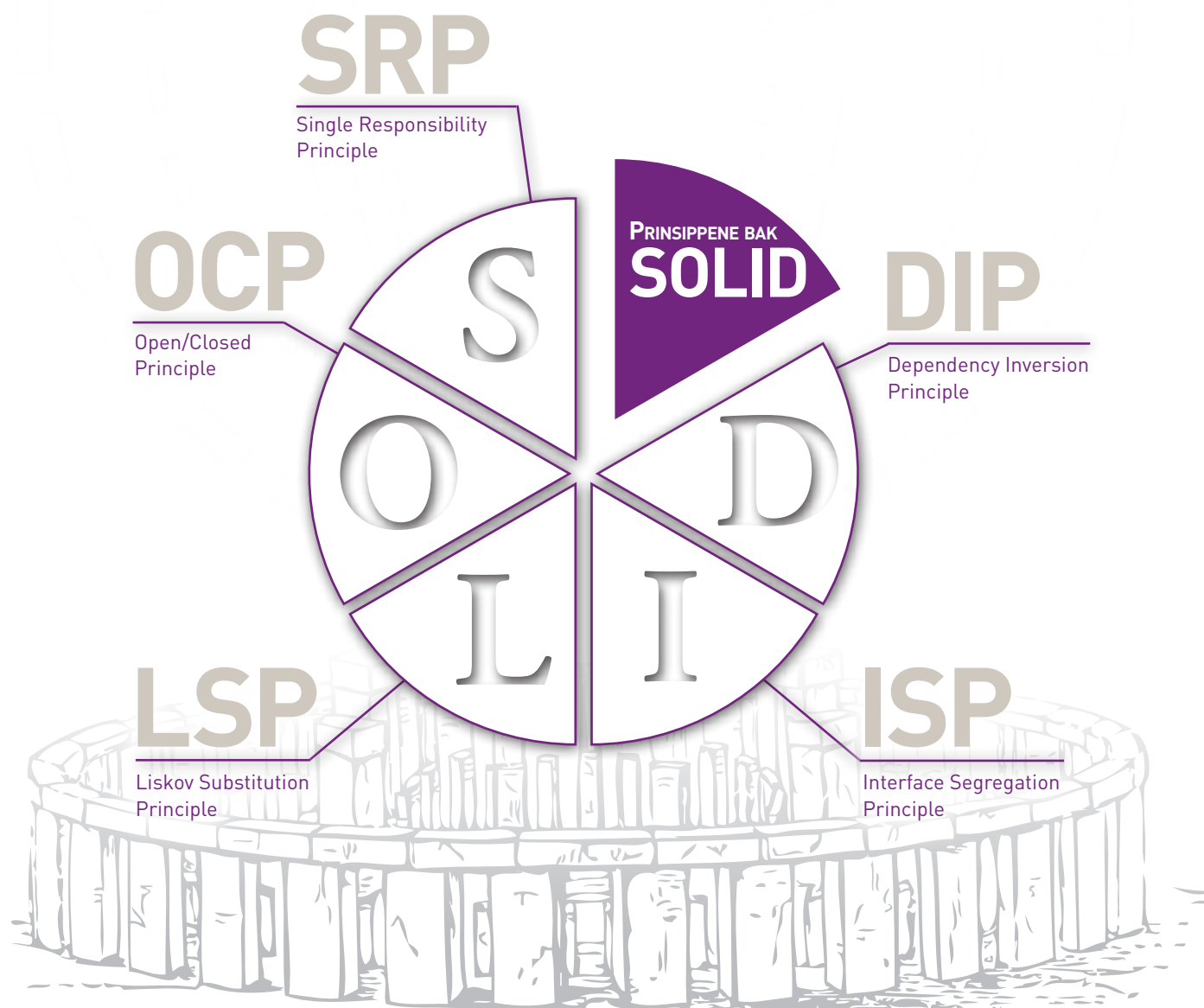
### Single Responsibility Principle

**SRP:** SRP dikterer at en klasse eller et objekt skal ha ett enkelt ansvarsområde. Dette er tett knyttet til konseptet om «kohesjon», altså hvorvidt forskjellig funksjonalitet innad i en modul hører sammen. Høy kohesjon er fordelaktig da det sikrer at en klasse ikke gjør for mye og gjør det lettere å skifte ut deler av kodebasen. Ved lav kohesjon vil det ofte bli høy kobling (mange avhengigheter) mellom klasser og dette fører til flere bieffekter ved endringer av enkeltdeler. Gjenbruk gjøres også vanskeligere, siden flere avhengigheter følger med hver modul. Et konkret eksempel her kan være at en klasse SALG trolig bør ha ansvar for å regne ut totalsum for alle produkter i salget, mens det å hente ut disse dataene fra databasen bør tilhøre en annen klasse.

SRP gjør også kodebasen vår mer robust for endringer. Dersom samme klasse for eksempel gjennomfører både utregning av data og tegning av en webside vil det være risiko for at vi ødelegger kode som skriver ut deler av websiden når vi skal endre måten dataene blir regnet ut på. Ved å splitte disse i klasser med hvert sitt ansvarsområde sikrer vi at data som flyter mellom dem forblir lik og at de dermed ikke påvirker hverandre.

Prinsippet kan - og bør - med fordel også brukes for metoder og funksjoner, og ikke bare klasser. Selv innad i SALG-klassen vi nevnte tidligere kan det skje mange ting som av natur er forskjellige og ikke hører direkte sammen. Vi kan tenke oss at en del av å regne ut totalsummen består av å beregne kvantumsrabatter for et «ta tre betal for to»-tilbud. Slik logikk kan med fordel skilles ut i egne metoder. Hadde vi lagt denne logikken





«Every object should have a single responsibility, and all its services should be narrowly aligned with that responsibility»

- OSCAR WILDE OM OBJEKTORIENTERT DESIGN



→ inn i et par for-løkker med noen lite beskrivende variabler i en allerede lang og kludrete prisutregningsmetode vil det være vanskelig å forstå hva som skjer. I tillegg kan vi ikke være sikre på om en endring av akkurat disse kodelinjene også påvirker andre deler av utregningen. Skilles det ut i en BEREKNKvantumSRABATT()-metode forstår vi med en gang hva koden gjør, og vi har en helt separat blokk med kode vi kan endre uten frykt for å ødelegge resten av programmet.

### Open/Closed Principle

**OCP:** OCP dikterer at en klasse, modul, funksjon eller lignende skal være «åpen for utvidelse, men lukket for endring». Det finnes flere måter å gjennomføre dette på, og tradisjonelt har det vært to betydninger av prinsippet:

Den opprinnelige tanken i prinsippet var at en klasse, når ferdigskrevet, kun skal endres for å reparere feil. Dersom man ønsker ny eller endret funksjonalitet lager man en ny klasse som kan, om ønskelig, gjenbruke funksjonalitet fra den originale ved hjelp av arv. Den trengte uansett ikke nødvendigvis å ha det samme grensesnittet som den opprinnelige klassen.

På 90-tallet endret definisjonen seg populært over til bruken av abstrakte grensesnitt. Her tenkes det at grensesnittet er fastsatt, og forskjellige implementasjoner kan lages og benyttes om hverandre basert på dette grensesnittet.

Selve grunnfilosofien i prinsippet sier altså at design av programvare bør gjøres på en slik måte at det å legge til ny funksjonalitet medfører minst mulig endring i eksisterende

kode. Vi ser allerede her at OCP og SRP jobber godt sammen. Ved å lage mange små klasser med hvert sitt ansvar kan vi konstruere mer avanserte strukturer gjennom å aggregere disse klassene sammen. Vi kan også fint bytte ut enkelte deler med ny og bedre funksjonalitet siden alle klasser følger fastsatte grensesnitt og dermed lover å gjøre den samme jobben for koden som benytter dem.

### Liskov Substitution Principle

**LSP:** LSP dikterer at hvis en klasse B er en subklasse av A, skal man kunne bytte ut instanser av A med instanser av B uten å endre ønskelig oppførsel i programmet, altså korrektheten av utregninger, oppgaver som blir gjennomført og så videre.

LSP er ikke bare et syntaktisk prinsipp, men enda viktigere et semantisk prinsipp.

Dette betyr at klasser i et hierarki skal fungere med samme semantikk - ikke bare gjennom samme grensesnitt.

For å få en bedre forståelse av konseptet kan vi se på et typisk brudd. Vi kan tenke oss en REKTANGEL-klasse med mulighet for å endre henholdsvis lengde og bredde. Om vi så subklasser denne klassen til en KVADRAT-klasse får vi et semantisk problem.

Det gir ikke lenger mening å kunne sette både bredde og lengde forskjellig. Kanskje vil vi da gjennomføre denne implementasjonen slik at dersom man setter lengde blir også bredde satt, og om man setter bredde blir også lengde satt. Dette betyr at vi ikke uten videre kan bytte ut en instans av REKTANGEL med en instans av KVADRAT da dette vil kunne gi oss feil oppførsel dersom vi prøver å sette forskjellig lengde på sidene, som vi naturligvis forventer at skal fungere for et generelt rektangel.

Eksempelet med REKTANGEL og KVADRAT er selve hovedeksempelet som blir brukt når LSP forklares, men hvordan man skal løse implementeringen av disse to klassene i tråd med LSP strides de lærde (og mindre lærde) om. I hovedsak kan man tenke seg to løsninger. En mulig løsning er å ha et grensesnitt der man ikke kan endre sidene, men kun sette dem ved opprettelse. Altså gjøre klassene «immutable» eller «umulige å endre». Et annet, og kanskje bedre alternativ er å spørre seg om man virkelig trenger å behandle kvadratet som en egen klasse. Så lenge brukende kode tar ansvar for å sette og sjekke at begge sider er like er det et kvadrat, og det gir kanskje ikke mening å fastsette dette i en egen datatype.

### Interface Segregation Principle

**ISP:** ISP søker å oppnå en kodebase med lav kobling, altså der deler av systemet er minst mulig avhengig av andre deler de ikke i utgangspunktet har noe å gjøre med.

Prinsippet gjør dette ved å diktere at grensesnitt skal splittes i mindre deler når det begynner å bli for stort. Dette skal gjøre at ingen brukere av grensesnittet skal være avhengig av operasjoner de ikke bruker.

Prinsippet stammer i utgangspunktet fra en jobb Uncle Bob (se faktaboks) gjorde for Xerox, der en modul for printerstyring ble så stor at den tok timer å kompilere hver gang man gjorde en liten endring. Istedenfor å ha grensesnitt for printing, stifting av papir, sending av fax osv. hadde de et stort grensesnitt som kunne gjøre alt printeren kunne. Dette betydde til slutt at enkel kode som kun ønsket å printe var avhengig av et grensesnitt som kunne gjøre et utall andre avanserte funksjoner.

Dette betydde også til slutt at man ikke kunne sette inn en enkel printer uten stiftet eller faksfunksjon da grensesnittet var avhengig av dette. Selvsagt kunne man laget en implementasjon der disse funksjonene ikke fungerte, men det ville i så fall krevd at koden som brukte implementasjonen måtte passe på å kun bruke de delene som faktisk

fungerte. Det å vite hvilke deler av koden som bare brukte printerfunksjonen og hvilke som brukte fax ville også være vanskelig. I tillegg kunne slike større endringer i kodebasen gi katastrofale følger ettersom kode man hadde oversatt brukte mer fra grensesnittet enn det som ble tilbudt.

Ved å splitte et slikt grensesnitt til tre forskjellige kan man bytte ut printerimplementasjonen for deler av koden som brukte denne, og fax for deler som trengte dette. Plutselig ser vi at det blir mulig å bytte ut en stor Xerox-maskin med en enkel printer og en enkel faxmaskin.

### Dependency Inversion Principle

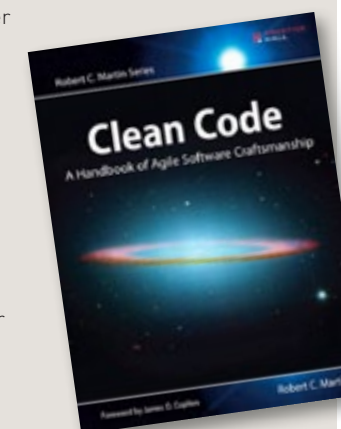
**DIP:** Når vi lager programvare bygger vi typisk et større system fra flere små lavnivå-komponenter (merk at med lavnivå mener vi kode som mer konkret jobber med detaljer, ikke nødvendigvis Assembly-kode eller lignende). Dette er gjerne bygget opp slik at høynivå-koden avhenger direkte av lavnivå-komponentene - både i form av

### MANNEN BAK SOLID

Akronymet ble først brukt av en nåtidens store personligheter innen utvikling: Robert C. Martin, kanskje bedre kjent som «Uncle Bob»

Sammen med blant annet Kent Beck, som readme nylig intervjuet, er vår kjære onkel en av opphavsmennene bak «The Agile Manifesto»

Uncle Bob har skrevet flere innflytelsesrike bøker innen temaer som smidig utvikling, og er kanskje mest kjent som forfatteren av boken «Clean Code». Han er i tillegg en av hovedpersonene i «Software Craftsmanship»-bevegelsen der det fokuseres på programvareutvikling som et håndverk der utviklere skal kunne levere arbeid de er stolte av og være en del av et miljø av dyktige fagfolk som kan lære av hverandre.







grensesnitt og gjerne også i hvordan koden henger sammen. Typisk vil en større komponent instansiere opp avhengighetene sine, for eksempel i en konstruktør eller når komponenten trengs.

Dette fører til at komponenter mister gjenbrukbarhet, da koden er avhengig av lavnivå-komponenter som kanskje ikke er relevante i andre sammenhenger. Går vi tilbake til printer-historier fra forrige prinsipp kan vi tenke oss at det er vanskelig å gjenbruke en utskriftsfunksjon som instansierer opp, og er avhengig av, en detaljert implementasjon for en av Xerox sine printere.

Dependency Inversion søker å løse dette problemet ved å innføre interfaces for hva som trengs av de lavere nivåene. Her kan man så implementere dette interfacet for hver enkelt lavnivå-komponent som tenkes utbyttet, og det blir plutselig mulig å bytte ut den tidligere nevnte printeren. Når dette er gjort benyttes patterns slik som *Plugin*, *Service Locator* eller *Dependency Injection* for å kunne konfigurere og gi en komponent de avhengighetene den trenger ved kjøretid.

For å illustrere dette enkelt kan vi se for oss to måter å implementere et skriveprogram med tre enkle lag. Tradisjonelt ville vi kanskje instansiert selve teksteditoren, denne instansierer så en mengde menyer og tjenester, deriblant en utskriftsfunksjon. Når denne utskriftsfunksjonen blir instansiert vil konstruktøren opprette en instans av driveren til skriveren som brukes når teksteditoren skriver ut.

Med en slik løsning ser vi at gjenbruken av teksteditoren på et annet system vil være vanskelig da det er flere lag med kode som alle instansierer spesifikke avhengigheter de krever.

Om vi snur opp ned på dette mønsteret kan vi heller tenke oss at printerfunksjonen tar inn et printerdriver-interface i konstruktøren, og at skriveprogrammet tar inn et



printerfunksjon-interface i konstruktøren sin. Dette gjøres ved å først lage en system-spesifikk bootstrapper (oppstartsmekanisme) som lager en korrekt skriverdriver, etterfulgt av en printerfunksjon som får denne skriverdriveren inn når den blir laget. Når disse er laget kan vi instansiere teksteditoren og gi editoren den printerfunksjonen som nettopp ble laget. Da er det mye lettere å konfigurere hvilke avhengigheter som skal brukes.

Ønsker vi for eksempel å bytte ut printerfunksjonen med noe som printer til PDF så kan vi lage en printerfunksjon som ikke trenger en skriverdriver, men i stedet skriver rett til fil, for deretter å gi funksjonen til teksteditoren.

### Konklusjon

SOLID-prinsippene er et viktig verktøy i enhver utviklers verktøykasse for å skrive gode systemer som er mulige å jobbe videre med og vedlikeholde. En litt banal og artig, men likevel fornuftig, filosofi innen programmering er at når man skriver kode skal den skrives som om personen som kommer til å overta jobben din er en gal øksemorder som vet hvor du bor.

Dersom du kjenner til og følger SOLID-prinsippene er du definitivt et greit stykke på vei for å skrive akkurat slik kode - og kunne sove godt om nettene. Som en ekstra bonus vil kode som etterstreber å følge SOLID-prinsippene være lettere å teste, kanskje spesielt som følge av DIP, SRP og OCP.

Er du en av de som har hatt problemer med å skrive gode unit-tester i prosjektarbeid? Eller ønsker du å prøve testdrevne utvikling? Testbar kode og SOLID går hånd i hånd. Skriver du kode som følger SOLID-prinsippene vil du oppleve at det er vesentlig lettere å skrive fornuftige enhetstester for den. Kanskje enda mer magisk: Skriver du gode tester først vil koden din automatisk trekke mot å følge SOLID-prinsippene. Prøv da vel!

### LES MER

For mer spennende lesing om noen av konseptene nevnt i denne artikkelen anbefales følgende stikkord i din favorittsøkemotor:

**Code Smells, Cohesion, Coupling, Dependency Injection, Service Locator, Test drive development**



Visma Sirius er et selskap i Visma-konsernet og er en nordisk leverandør av verdiskapende IT-løsninger rettet mot offentlig sektor og utvalgte bransjer i privat sektor. Selskapet er representert i Norge, Sverige og Danmark med mer enn 450 ansatte. Visma Sirius leverer systemutvikling av virksomhetskritiske IT-løsninger, applikasjonsforvaltning og konsulenttjenester. Fokus for løsningsleveransene er selvbetjente saks-/fagsystemer, skjema- og portal-løsninger og forretningssystemer.

## Er du nyutdannet og ønsker å jobbe som konsulent?

*Er du ferdig med din utdanning rundt jul 2011, og ønsker å jobbe som konsulent?*

*Visma Sirius ansetter nyutdannede med oppstart kontinuerlig.*

Ved å jobbe som konsulent i Visma Sirius vil du få bruke din fagkunnskap til å løse utfordringer og behov for kundene våre. Vanligvis innebærer det å arbeide i et større eller mindre team av konsulenter sammen med kunden i et prosjekt.

I en kompetansebedrift som Visma Sirius er de ansatte vår viktigste ressurs. Vi tilbyr derfor gode muligheter for personlig og faglig utvikling. Slik som vi lever av å skape de beste it-løsningene til våre kunder, lever vi også for å skape en god arbeidsplass for oss selv.

I Visma Sirius får du jobbe med noen av de mest spennende IT-prosjektene i Norge!

Send søknad til [jobb.no.sirius@visma.com](mailto:jobb.no.sirius@visma.com)

eller les mer på [visma.no/Sirius](http://visma.no/Sirius)



Følg oss på Facebook:  
[www.facebook.com/vismasirius](http://www.facebook.com/vismasirius)



### Nordens ledende fagmiljø innen økonomi og IT

Visma gjør virksomheter mer effektive, gjennom å levere programvare, outsourcingtjenester, innkjøps-løsninger, innfordringstjenester, butikkdata-løsninger, samt IT-relatert utviklings- og konsulentvirksomhet. Vi forenkler og automatiserer virksomhetsprosesser innenfor både privat og offentlig sektor. 320 000 nord-europeiske kunder benytter våre produkter og tjenester, og ytterligere 330 000 bruker oss som hosting-partner. Vi er over 5 000 ansatte og hadde i 2010 en omsetning på NOK 4 168 mill.







AMAZON KINDLE 4:

# Tusen bøker i sekken

For ikke så lenge siden kom Amazon ut med en ny generasjon Kindle, og forventningene var store. Det er få store innovasjoner fra forrige modell, men forbedringene er mange, subtile og smarte.

TEKST: JOHN-ERIK JOHANSEN  
FOTO: TINA JUNI STORØ, AMAZON.COM

For de av dere som ikke har hørt om denne enheten før er Kindle et digitalt lesebrett med e-ink teknologi. E-ink er elektronisk papir som ikke reflekterer lys på samme måte som LCD-skjermer gjør. Dette gjør at man ikke blir så sliten i øynene når man leser på det, og man bruker bare strøm når man bygger en ny side.

Direkte fra lesebrettet får man tilgang til Amazon sin digitale bokhandel, slik at en kan kjøpe bøker direkte og laste dem rett ned til enheten.

Det nye med fjerde generasjons Kindle er blant annet at de har fjernet tastaturet, slik at den

blir mindre og lettere. Skjermen er bedre, med høyere kontraster, og hele enheten føles mer solid når du holder den i hånda. Dette fordi de i den siste generasjonen har byttet ut deler av skroget med metall istedenfor plast. Sidebytterne går raskere på grunn av den nye prosessoren, og lesebrettet oppleves som mer responsivt.

Da jeg åpnet pakken var det første jeg la merke til hvor mye mindre Kindle var enn den forrige modellen. Jeg klarte ikke å tro at skjermen var like stor, så jeg måtte fram med målebånd. Det viste seg at det var den.

Fra Kindle shop får du tilgang til over en million bøker, og enheten kan lagre minst 1400 av disse. Den leveres med en innebygget nettleser og en egen applikasjon

for Wikipedia. Den har også en innebygget ordbok som kommer med definisjon av det ordet du holder markøren over, noe som har vært meget hjelpsomt under lesing av fagtekster.

## Studievennen

Å ha en Kindle i skolesekken er ganske nyttig for en student. Det blir ofte tungt å dra med seg skolebøker i sekken, og man glemmer av og til å skifte ut bøkene dag for dag etter hva man har tenkt å lese til. Mange av lærebøkene er rimeligere på Amazon.com i e-bok-format enn i bokhandelen. Og er det noe man ikke finner der så går det an å lese ebøker kjøpt i norske nettbutikker på Kindle.

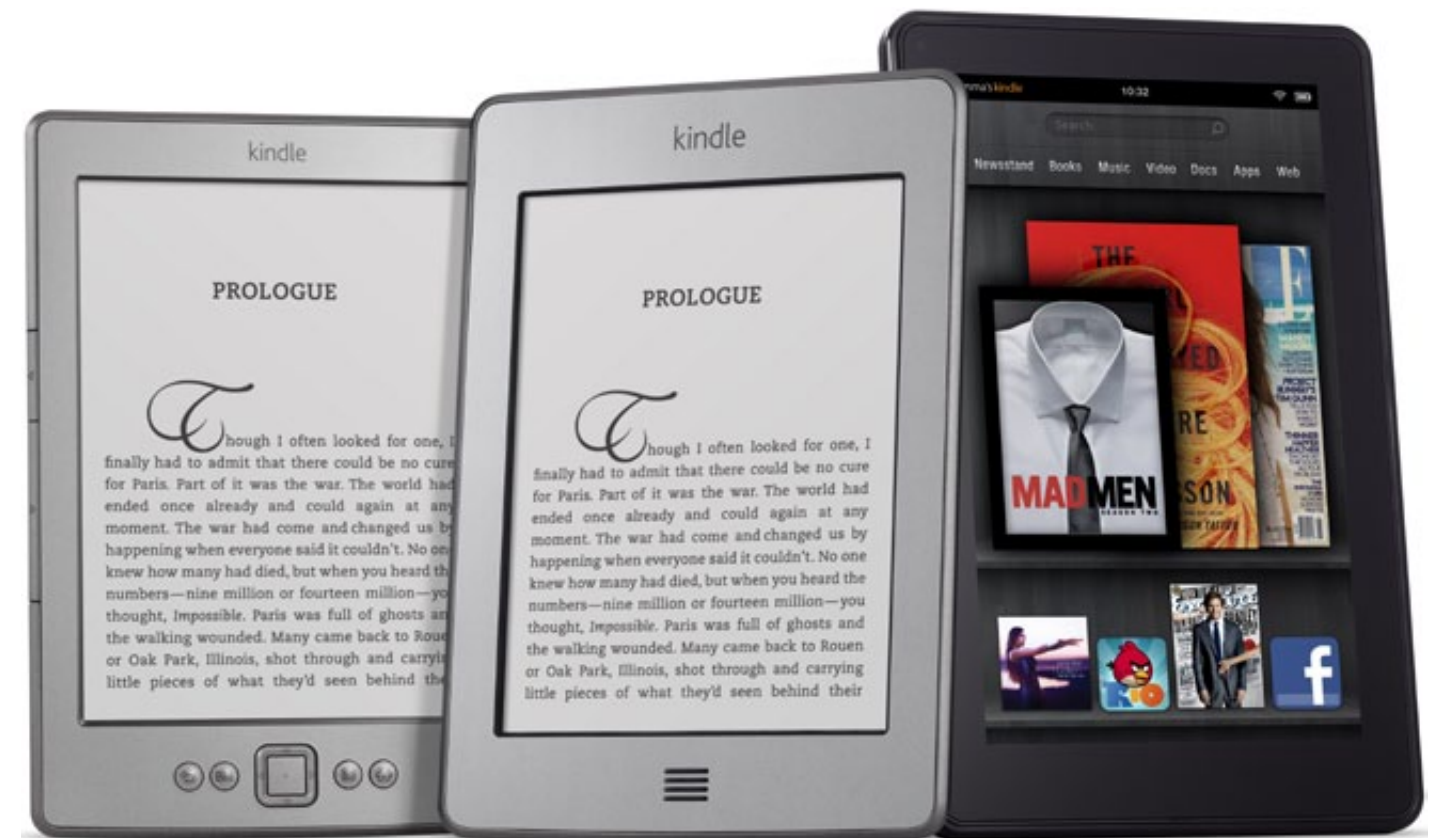
Har du ofte pensum, forelesningsnotater og foiler i PDF-format

kan de også fint leses på den. Om sidene er litt for store, kan du sette den i landskapsmodus og bla deg ned en halvside av gangen. Dette gjør at undertegnede har droppet papirbøker helt og har all faglitteratur med seg i sekken til studie og fornøyelse.

Er det derimot et krav at du skal kunne lese hele siden på en gang, så er kanskje storebroren, tredje-generasjons Kindle DX noe for deg. Den gir deg komplett A4-format på skjermen, og gjør lesing av store dokumenter med mange figurer og illustrasjoner mer behagelig.

## Konklusjon

Alt i alt er siste generasjon Kindle en forbedring av den forrige. Den er mer robust, noe som gir



en bedre kvalitetsfølelse enn den gamle. Den tar mindre plass og er lettere i sekken, faktisk så får den plass i baklomma på de fleste jeans. Jeg savner av og til tastaturet når jeg skal søke etter ting på Wikipedia eller på nettet, men dette er ikke noe problem på Kindle Touch. Det er den perfekte studiekameraten og gjør pensumlesing utenfor skolen og hjemmet mye lettere (spesielt fysisk).

## KINDLE

**Skjerm** 6"  
**Vekt** 170g  
**Mål** 17,2 x 11,4 x 0,9 cm  
**Batteri** Opptil 1 måned  
**Inngang** Usb 2.0 (micro-B)  
**Utgang** Ingen  
**Minne** 2 GB  
**Pris** Fra \$79

Dette er Amazons enkleste og rimeligste E-bok leser. Kommer kun med Wi-Fi.

## KINDLE TOUCH

**Skjerm** 6" multitouch  
**Vekt** 213/221g (m/u 3G)  
**Mål** 17,2 x 12 x 1 cm  
**Batteri** Opptil 2 måneder  
**Inngang** Usb 2.0 (micro-B)  
**Utgang** Stereo audio jack  
**Minne** 4 GB  
**Pris:** Fra \$99 / \$149

Litt mer eksklusiv, med touch-skjerm, større minne og bedre batteri. Kommer med Wi-Fi og 3G.

## KINDLE FIRE

**Skjerm** 7" multitouch  
**Vekt** 413 g  
**Mål** 19 x 12 x 1,1 cm  
**Batteri** Opptil 8 timer  
**Inngang** Usb 2.0 (micro-B)  
**Utgang** Stereo audio jack  
**Minne** 8GB  
**Pris** \$200

Amazons første nettbrett er Android basert. Det er lagt opp en helhetlig løsning med tilgang til et eget bibliotek med film og serier. Kommer kun med Wi-Fi.



# INNSIKT I HACKERNES VERDEN DEL 3: CROSS SITE REQUEST FORGERY

I del 1 og 2 av denne serien har vi sett hvordan SQL-injections og cross-site scripting fungerer. I denne delen skal vi se på en angrepsmetode som er veldig relatert til XSS; cross-site request forgery.

TEKST: DAG ERIK VIKAN  
ILLUSTRASJON: SVERRE JOHANN BJØRKE

Cross-Site Request Forgery (CSRF) er en metode for å angripe en bruker av en webapplikasjon, mens serveren som hoster applikasjonen er helt trygg. CSRF er ganske likt XSS, som vi tok for oss i forrige utgave. Men som du kanskje husker fra sist gang, må man injisere html/javascrip i webapplikasjonen for å kunne utføre et XSS-angrep. Det som gjør CSRF til et effektivt angrep er det faktum at en angriper ikke trenger å finne noe XSS-hull. La oss se på et veldig enkelt eksempel:

```
http://webapp/deluser.php?id=34
```

URL-en du ser her er en typisk lenke som en administrator av en webapplikasjon må klikke på for å slette bruker med id 34. Dersom du bruker din egen nettleter og besøker denne URL-en, vil det ikke skje noe spennende. Årsaken er naturligvis at du ikke er administrator, eller du kanskje ikke er logget inn i webapplikasjonen i det hele tatt. Det som CSRF kort fortalt går ut på, er å lure nettleteren til en innlogget bruker til å besøke URL-er som den ovenfor eller å submitte skjemaer til webapplikasjonen. Lenken må oversendes til offeret på et vis, via for eksempel email eller IRC-chat. URL-en ovenfor ser skummel ut, og veldig få vil finne på å klikke på den. Men det finnes subtile måter å få offerets nettleter til å fyre avgårde HTTP-forespørsler i bakgrunnen. Her er et godt eksempel:

```
<img src='http://webapp/deluser.php?id=34'>
```

Blir en nettleter presentert med denne HTML-koden, vil den gjøre en HTTP GET-forespørsel mot URL-en, fordi den prøver å hente ned bildet. Selv om koden er hostet på en annen webserver, vil nettleteren fortsatt gjøre forespørselen; dette kunne for eksempel vært et bilde som ble lagt inn i en forumpost på samme webapplikasjon som angripes. Da trenger offeret kun å lese forumposten din, og det vil være veldig effektivt. Eksempelet over vil forårsake en feil fordi bildet ikke finnes, men det har ikke noe å si for en angriper som bare vil ha brukeren slettet.

## Hvorfor fungerer dette?

Den observante leser vil påpeke at dette ikke burde fungere, fordi forespørselen kommer fra et annet domene enn webapplikasjonen og at sesjonskoken ikke burde følge med. Men jo, den gjør det, og dette fungerer. HTTP er tilstandsløst, og en webserver kan ikke gjenkjenne samme bruker over flere HTTP GET- eller POST-forespørsler. Så når nettleteren til offeret sender avgårde forespørselen i figur 1 tror webapplikasjonen at forespørselen kommer fra en gyldig bruker, og gjør bare jobben sin og sletter brukeren.

## Hva med POST-skjemaer?

Selv om forespørselen man ønsker å utføre benytter et POST-skjema, er man ikke mye tryggere av den grunn. For eksempel kan en webapplikasjon ha et skjema der man får oppdatere profilen sin, slik som i figur 2.

På et slikt skjema er det umulig å gjøre et CSRF-angrep via en HTML-bildetag, men med javascript kan man konstruere et skjema for deretter å submitte det, se figur 3.

Man må naturlig nok lure offeret til å vise denne siden i sin nettleter, men i motsetning til XSS kan denne siden ligge på en vilkårlig server.

## Eksempelangrep mot en fiktiv bank

La oss si at en angriper vil overføre penger fra offeret til seg selv på nettbanken til offeret. La oss, av pedagogiske grunner, også anta at vi er tilbake til 90-tallet, hvor forsvarsmekanismer var ikke-eksisterende. Først må angriperen undersøke hvordan en pengeoverføring skjer; det vil si hvilke variabelnavn som brukes, GET eller POST, rekkefølge på forespørsler osv.

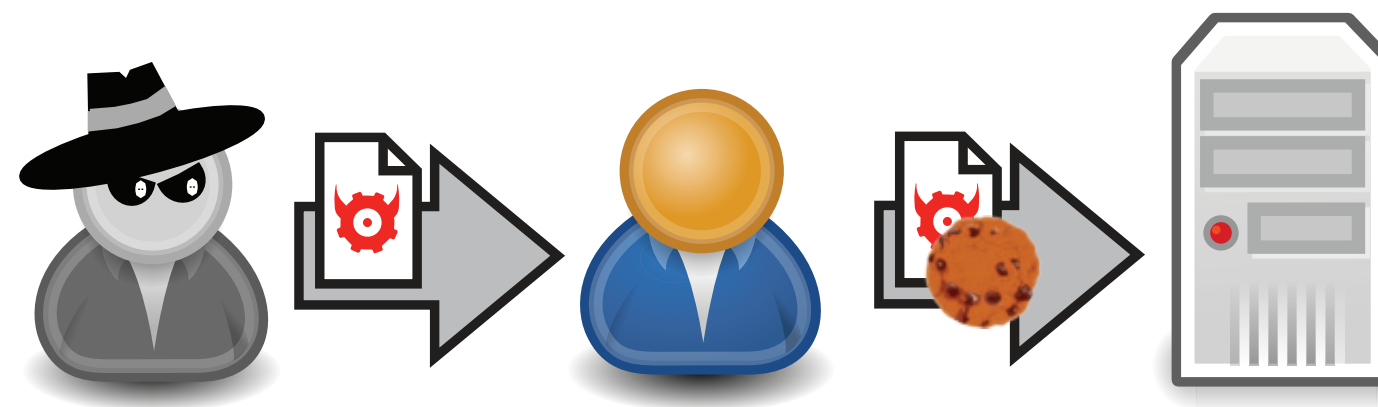
Etter en stund har angriperen konstruert to skjemaer som vil gjennomføre en transaksjon. Hele angrepskoden vil se ut som figur 4.

Her gjøres først overførselen via en HTML-bilde-tag. Men på denne fiktive nettbanken er ikke dette nok, brukeren må også godkjenne overførselen ved å trykke på en knapp. Dette simuleres ved å opprette et POST skjema, for deretter å submitte skjemaet med korrekte verdier som vil godkjenne transaksjonen.

Etter å ha konstruert koden for å stjele pengene, gjenstår det kun å overlevere denne til offeret. Angriperen sender da en email med falsk avsender, bank@fiktivbank.com, med en liten tekst som oppfordrer offeret til å klikke på lenken. Lenken kan for eksempel se slik ut:

```
http://fictivbank.com/important_info.html
```

Her er domenenavnet forskjellig med kun en bokstav som er byttet ut. Angriperen sender denne emailen til 300 kunder av banken. Hvor mange av disse tror du angrepet vil fungere på?



## Er POST tryggere enn GET?

La meg med en gang si at uansett hvordan webapplikasjonen er bygget opp burde endetil-ende-kryptering være på plass. Ellers sendes sensitiv informasjon over Internett i klartekst. Ifølge W3C standarder skal man bruke GET til å hente ut informasjon, og POST til skjemaer som endrer på noe. Å bruke GET eller POST på et skjema utgjør ikke store forskjellen. Likevel vil bruk av POST forhindre et par ting som for eksempel at skjemaerverdiene ikke vil bli lagret i nettleterhistorikken, og det vil heller ikke være mulig å utføre angrepet via en HTML-bildetag. Men for en dedikert hacker har ikke dette særlig mye å si.

## Hvordan forhindre?

For å forhindre CSRF-angrep kan man få serveren til å sette inn en tilfeldig verdi i skjemaet som sendes til brukeren slik vi viser i figur 5.

Dette kalles en CSRF-token, og lagres samtidig på webserveren. Når skjemaet blir sendt inn sammenligner webapplikasjonen verdien den får fra brukeren med verdien den har lagret; hvis de ikke er like avvises forespørselen. Dette vil forhindre CSRF-angrep, for hva skal angriperen legge med som verdi for CSRF-tokenen når den er tilfeldig generert?

```
1. GET /webapp/deluser.php?id=34 HTTP/1.1
2. Host: www.example.org
3. Cookie: PHPSESSID=64cd786173875c162d3b84cdf5bbac9d;
```

1

```
1. <form action='profile.php' name='attack_form' method='POST'>
2. <input type='text' name='real_name'>
3. <input type='text' name='email'>
4. <input type='text' name='age'>
5. <input type='submit' name='save'>
6. </form>
```

2

```
1. <form action='profile.php' name='attack_form' method='POST'>
2. <input type='text' name='real_name'>
3. <input type='text' name='email'>
4. <input type='text' name='age'>
5. <input type='submit' name='save'>
6. </form>
7. <script>document.attack_form.submit();</script>
```

3

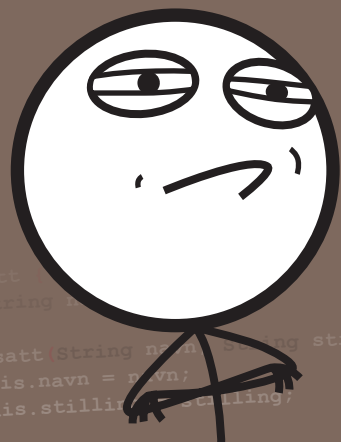
```
1. <img src='transfer.php?from=23543432567&to=65458745650&
2.   ammount=9001'>
3. <form action='transfer.php' name='attack_form' method='POST'>
4. <input type='hidden' name='confirm' value='yes'>
5. </form>
```

4

```
1. <form action='profile.php' name='attack_form' method='POST'>
2. <input type='text' name='real_name'>
3. <input type='text' name='email'>
4. <input type='text' name='age'>
5. <input type='submit' name='save'>
6. <input type='hidden' name='csrftoken'
7.   value='49297cb072918d1a0084a7570305821a'>
8. </form>
```

5





# kjøkken HACKS



```

1.  /**
2.   *
3.   * Beate Baier Biribakkens scampipasta
4.   * Et enkelt, imponerende og velsmakende måltid
5.   *
6.   * @author Erik Lothe
7.   * @version 1.0
8.   */
9.  */
10.
11. import kitchen.*;
12.
13. public class ScampiPasta extends Meal {
14.
15.     Ingrediens pasta    = new Tagliatelle (Enhet.HALV_PAKKE);
16.     Ingrediens scampi    = new Scampi (Enhet.EN_PAKKE);
17.     Ingrediens cremeFraiche = new CremeFraiche (Enhet.HALV_BOKS);
18.     Ingrediens pesto     = new Pesto (Enhet.HALVT_GLASS);
19.
20.     Kasserolle gryte = new Gryte ();
21.     Kasserolle stekepanne = new Stekepanne ();
22.
23.     public ScampiPasta () {
24.         stekepanne.add(scampi);
25.         stekepanne.stek();
26.
27.         gryte.add(pasta);
28.         gryte.kok();
29.
30.         stekepanne.removeVann ();
31.         stekepanne.add(cremeFraiche);
32.         stekepanne.add(pesto);
33.         stekepanne.stek();
34.
35.         gryte.removeVann ();
36.         gryte.server ();
37.         stekepanne.server ();
38.
39.         enjoyMeal (true);
40.     }
41. }

```

Evner du å skrive noe annet enn **lorem ipsum**? Har du **originale idéer**? Har du lyst til å ta **blider**? **Intervjuer**? **Korrekturlese**? Har du egne **meninger**? Hater du **Comic Sans**? Eller har du et talent for **urban exploring**. Ønsker du deg en **pressehalt** til jul?

**OFFLINE**  
**HAR BEHOV FOR FLERE FOLK**  
 opptak@online.ntnu.no  
**SØK NÅ!**

Navn: Nicolay  
 Hobby: Amerikansk fotball (7 norgesmesterskap)  
 Lidenskap: Systemutvikling og design  
 Jobb: Senior systemutvikler



## Få ditt gjennombrudd

### Steria skal ha 50 graduates

Er du på tampen av studiene og lurer på hvilken karrierevei du skal velge? Vil du jobbe med noen av Norges mest spennende og krevende it-prosjekter? Vil du utvikle alle dine talenter – både faglig og personlig? Vil du også ha tid og overskudd til et liv utenfor jobben? Da bør du seriøst vurdere Steria.

Selskapet er kåret til en av Norges mest attraktive it-arbeidsplasser. Søk om opptak til vårt graduateprogram med oppstart høsten 2012 på [www.steria.no/jobb](http://www.steria.no/jobb).





# Refresh



I denne spalten oppdaterer vi deg kort om hva som har skjedd på IDI, hos Online og ellers i studiebyen siden forrige Offline.

## Stort inntak på våreropptaket til master informatikk

IDI har i år åpnet for et opptak på 23 personer på våreropptaket til masterstudiet i informatikk. Plassene fordeler seg med syv plasser på *Systemarbeid og MMI*, åtte plasser på *Kunstig intelligens og læring*, tre plasser på *Komplekse systemer*, fire plasser på *Informasjonsforvaltning* og én plass på *Spillteknologi*.

## Nye representanter for IME i Studenttinget

Anja Ziolkowski, Morten Skadsem Holthe, Kristine Valentinsen, Madeleine Lorås er valgt inn som representanter for IME-fakultetet i Studenttinget for 2012. Vi vil gratulere Anja Ziolkowski som går informatikk.

## Ny ledelse i Studenttinget valgt

Torsdag 24. november ble ledelsen og arbeidsutvalg for Studenttinget 2012 valgt. Vi gratulerer Cathinka Thyness (DMF) som ny leder, Jon Mikkel Haugen (IVT) som ny fag- og forskningspolitisk nestleder og Anja Ziolkowski som organisatorisk nestleder.

## Espen Skarsbø Olsen har giftet seg

Vår kjære kjekke, fantastiske, smarte, uerstattelige, unike, vakre og karismatiske Espen har giftet seg, og Offline ønsker ham og Tonje lykke til videre!

## Årets lærer og årets formidler

IME-fakultetet har kåret Andrew Stacey som underviser i lineære metoder til årets lærer, og Alf Inge Wang som forsker på spillteknologi på IDI til årets formidler. Alf Inge Wang underviser også i Programvarearkitektur (TDT4240) som fikk gode tilbakemeldinger i semesterundersøkelsen på IDI for våren 2011.

## Ny professor på IDI

Theoharis Theoharis fra Hellas ønskes velkommen til IDI som ny professor innen datagrafikk. Han tar over etter Torbjørn Hallgren som pensjonerer seg til jul. Theoharis vil derfor være faglærer i TDT4195, Bildeteknikk, fra og med neste semester.

## Programmering GK og Informatikk basisfag legges ned!

Det er bestemt at Programmering grunnkurs (ITI103) og Informatikk basisfag (ITI101) legges ned. ITI103 erstattes med Informasjonsteknologi grunnkurs (TDT4105). IKT, kultur og samfunn (ITI603) flyttes til høstsemesteret for å fylle hullet etter ITI101. ITI102, Informasjonsteknologi grunnkurs vil også bli lagt ned. Det er flere emner som planlegges å legges ned i neste års gjennomgang av emnene på IDI.

God jul & Godt nyttår!



Fra alle oss i redaksjonen og komitéene forøvrig.