Data- og komtekstudentenes linjeforeningsmagasin • Utgave 3, 2023

readme

Fem smertefulle faser

CEO i selvskryt

Let's get this bread

Et forhistorisk språk?

readme er gratis og blir skrevet av og for studenter ved Datateknologi og Kommunikasjonsteknologi og digital sikkerhet ved NTNU.

25. årgang utgave 3, 2023

REDAKTØR
Viljan van Raaij

LAYOUTANSVARLIG
Iver Ringheim

MEDVIRKENDE
Elena Kristine Blakstvedt Willmann
Eli Fjellbirkeland Johannesen
Maximillian August Steinsrud
Solveig «Doggen» Heitmann
Madelen Hellervik Lothe
Petter Obama Buseth
Ian McGregor Aksum
Emmanuel Quarcoo
Vetle Roos Mangrud
Magnus Hanesand
Martine Mansåker
Martine Karlsen
Øyvind Monsen
Emir Derouiche
Karoline Sæbø
Tobias Skjelvik
Vegard Ervik

FORSIDE
Karoline Sæbø

KONTAKT
readme@abakus.no

readme, Abakus
Sem Sælands vei 7–9
7491 Trondheim

NETTUTGAVE
readme.abakus.no

netcompany

Leder

Hei alle abakuler. Mitt navn er Viljan van Raaij, og jeg ønsker dere velkommen til min aller første utgave som redaktør i **readme**. Jeg har vært så heldig å kunne ta over denne rollen, og det neste året gleder jeg meg masse til å jobbe sammen med denne herlige gjengen i kreasjonen av mange nye utgaver om alt mulig rart. Faktisk så er dette første gang jeg skriver i **readme**, til tross for at jeg har vært med i snart to hele år. Men en gang må jo være den første.

Nå er eksamensperioden i gang, og med det håper jeg at denne utgaven av **readme** vil være en god grunn til å prokrastinere lesingen enda mer. Jeg kan glede dere med at dere kommer til å kunne være eksperter på både brødbaking og backendutvikling etter denne utgaven. Dere som skal ha programvareutvikling kan forberede dere mentalt til dette fantastiske faget gjennom artikkelen «PUs femfasemodell». I tillegg blir det selvfølgelig nok en side med gløsløker, en side med sladder og ikke minst en midtside.

Når vi først er inne på midtsider har jeg en oppfordring til alle dere som leser: Midtsiden er ment for alle, så hvis dere er en vennegjeng på alt fra to til ti personer og dere har en morsom idé, så er det bare å sende en mail til *readme@abakus.no*. Kanskje blir dere valgt ut til å være med neste gang.

Til slutt unner jeg dere alle en velfortjent sommerferie sammen med venner og familie. Vi ses!

Redaktør

I denne **utgaven**

7 Fem faser alle datastudenter må gjennom

10 Slik bør din LinkedIn se ut

20 Kickstart din bakekarriere

Et samlende symbol

Tekst: *Magnus Hanesand*
Foto: *Viljan van Raaij*

Denne teksten gir uttrykk for skribentens egne meninger

Abakus er en stor linjeforening hvis studieprogrammer er flere og programmenes spesialiseringer er enda flere. Det er ikke lett å skape en samlet linjeforening, men det er likevel et mål det synes å være konsensus om. Hvor går veien mot et samlet Abakus, og hvordan skal vi komme oss dit?

SPLITTET LINJEFORENING

Det er et velkjent problem at spesielt klassene på data kan slite med miljøet. Det er vanskelig nok å få et forhold til 140+ personer, men det finnes nok av faktorer som gjør saken enda verre. Et stort antall internoverganger og mangelen på eksklusive emner gjør det praktisk umulig å vite hvem som faktisk går i klassen din til enhver tid. En økt satsning på klassearrangementer fra Abakus sin side er et godt tiltak, men man kommer ikke unna at forholdene i stor grad ligger til rette for at klassene på data blir splittet.

Abakus i seg selv er for så vidt også splittet. En linjeforening bestående av to like, men samtidig veldig forskjellige studier. Data og komtek hører til forskjellige institutter og har forskjellige oppholdsarealer. Data sokner naturlig til A3 og Abakus-kontoret, mens komtek sokner til komtek-loungen. Jeg tror alle på data syns det er koselig at mange komtekere tar turen til A3 til tross for tilgang på lounge, men faktum består til tross – linjeforeningen er delt.

HVA MANGLER VI?

Data kommer aldri til å bli lite nok til at et klassemiljø på komtek-nivå er mulig, og å frata komtek loungen deres blir meningsløst. Det blir med andre ord Abakus sin oppgave å skape et samlende miljø – men hvordan gjør man det? Vi er ikke en 100+ år gammel linjeforening med haugevis av unike tradisjoner. Ei heller har vi noen definerende klesplagg, slik som kiltene til Hybrida. I mangel på samlende symboler, benytter jeg derfor muligheten til å offentlig lansere Abakus' maskot her i **readme** for aller første gang.

Personer som har fulgt godt med i **readme** eller revyen de siste årene, vil kanskje kjenne igjen en rød ku. Kuen er ingen ringere enn Abakua, personifiseringen av alt Abakus er og står for, alt som er godt, spennende og lærerikt her i livet. Jeg kan ikke se for meg et mer samlende symbol – en maskot på nivå med Mikke Mus og Michelin-mannen. Er du alene og savner klassekameratene dine? Tenk på Abakua. Sliter du med siste innspurt i objekt? Tenk på Abakua. Trenger man egentlig en maskot til linjeforeningen sin? Selvfølgelig gjør man det, det er Abakua.

Fullt på A3?

Frykt ikke!

Tekst: *Martine Meng Karlsen*
Layout: *Viljan van Raaij*

Hvor mange ganger i løpet av semesteret har du sagt «Jeg tar det igjen i eksamensperioden»? Mest sannsynlig altfor mange ganger. Nå er eksamensperioden her, og du må faktisk ta igjen alt du ikke gjorde de siste månedene. Dette fører ofte til lange dager med lesing. Er det for mange distraksjoner hjemme, fullt på A3, eller for mye unødvendig støy på kafé? Vi har samlet noen tips til leseplasser du kan benytte deg av i eksamensperioden.

SAMARBEID

I og med at Gløshaugen rommer hovedkvarteret til Abakus, nemlig kontoret, kan man nok med trygghet fastslå at Gløshaugen er den mest populære campusen. Dessverre er det ikke plass til alle på A3, men det finnes heldigvis andre, kanskje til og med bedre, alternativer. Rom som tilbyr store bord, skjermer og tavler passer utmerket hvis man skal samarbeide og diskutere. Noen gode rom er Vrimleområdet i U1, Lise og Smia på Stripa og Landmålerhallen på Byggteknisk. Rom som ellers brukes til undervisning står som regel tomme i eksamensperioden og kan brukes av hvem som helst, som R2 og R22.

Et tips er å søke etter kategorien «Studentaktive læringsareal» på MazeMap for å finne flere muligheter. En ekstra mulighet for komtek-studenter er at Sahara i Elektro E/F heller ikke brukes i eksamensperioden. Derfor kan rommet benyttes til eget bruk, enten for å ha en plass å sitte eller for å bruke datamaskinene dersom det skulle være behov for det.

MASTERSAL

Går du i 4. eller 5. klasse har du muligheten til å søke om mastersalplass. Da er du garantert å alltid ha en plass å jobbe og er ikke avhengig av å stå opp tidlig for å sikre en plass på lesesal. I tillegg får man tilgang til teknisk utstyr, som datamaskiner og programvare, som kan være nyttige ressurser i arbeidet ditt. Det er også mest sannsynlig andre studenter i samme situasjon som deg der, slik at det er fullt mulig å finne noen å samarbeide og diskutere med. Husk at det generelt skal være stille på sal, og at man må respektere de rundt. Man søker om plass hvert semester, men det er ingen garanti for at du får en plass, ettersom det er flere studenter enn tilgjengelige plasser. Om du ikke har plass i lesende stund, så er det nok dessverre for sent for denne eksamensperioden, men det kan være verdt å notere seg til neste semester, eventuelt til man selv går i 4. eller 5. klasse.

DIVERSE BIBLIOTEK

Om du liker å jobbe alene i stille omgivelser er biblioteket et godt alternativ som arbeidsplass. Selv om du kanskje hovedsakelig er på Gløshaugen, har NTNU flere bibliotek spredt rundt i store deler av Trondheim. Om du vektlegger beliggenhet, åpningstider eller ledige plasser også etter klokken 08:00, er opp til deg. Realfagbiblioteket har cirka 400 leseplasser og er åpent fra klokken 06:00 til 24:00 alle dager med NTNU-kortet. Det er kanskje også det mest populære biblioteket, så det kan være lurt å være tidlig ute om du ønsker å sikre deg plass her.

Er man villig til å reise litt lenger enn til Gløs, kan man slumre litt lenger på morgenen. Biblioteket Dragvoll har nemlig omtrent like mange leseplasser og samme åpningstider, men ifølge anonyme kilder skal det være mindre trengsel og lettere å få plass når som helst på dagen.

1. Landmålerhallen

2. Lise & Smia

3. Vrimleområdet i U1

Dersom Dragvoll blir for langt unna, eller for udannet, så er både Gunnerus- og Lysholm-biblioteket på Kalvskinnet gode alternativer. Lysholmbiblioteket skal, ifølge NTNU sine sider, være det biblioteket med nest flest leseplasser, etter Realfagbiblioteket og Dragvoll. Her er det både rom for å sitte alene og i grupper. Biblioteket er åpent klokken 07:00 til 24:00 med kort, men ligger enda mer sentralt enn Realfagbiblioteket. Gunnerusbiblioteket har på sin side en mer begrenset åpningstid fra klokken 09:00 til 18:00 på hverdager, med unntak av fredag fra klokken 09:00 til 15:00, og stengt i helgene. Det er færre leseplasser med bare 76 stykk på sal, samt noen ekstra bord og rom som kan benyttes til gruppearbeid, men til gjengjeld er det moderne, lite trengsel og like sentralt som Lysholm.

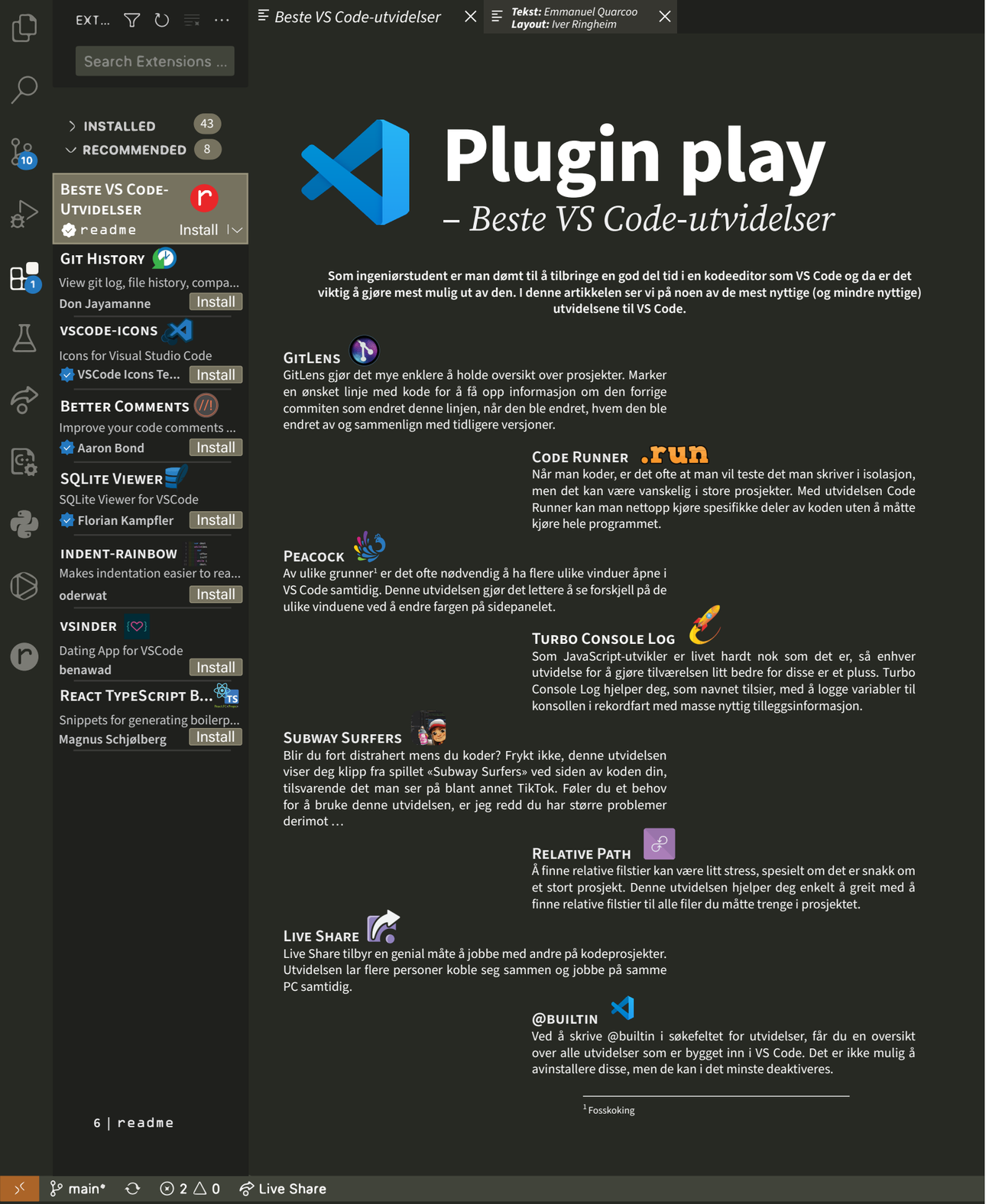
NB! Vær obs på at pulter ulike steder kan være ustøe, noe som ikke er veldig behagelig å skrive på, og benyttes på eget ansvar.

GRUPPEROM

Grupperom, eller andre rom etter behov, kan på alle NTNU sine campuser reserveres gjennom *ntnu.no/romres* og kan være en måte å sikre seg en ledig leseplass. I tillegg kan det å tvinge seg selv til å jobbe innenfor et visst tidsrom føre til at man ikke skyver arbeidet foran seg. Det varierer veldig hvor mange ledige tidspunkter et grupperom har, så det kan være lurt å reservere god tid i forveien om det er spesifikke rom man ønsker. Det er mulig å reservere rom opptil to uker i forveien. Om man er fleksibel på beliggenhet, holder det å reservere et par dager før man tenker å bruke rommet.

Ansvarsfraskrivelse: Artikkelen er kun ment som tips for abakuler som leter etter steder å lese, og er ikke en fasit på pågangen de ulike stedene. `readme` tar ikke ansvar for fulle lesesaler, at favorittrommene dine nå er reservert eller at du var for sen med å søke mastersalplass.

4. Hovedbygningen 326



PUs femfasemodell

Tekst: Eli Fjellbirkeland Johannesen
Layout: Elena Willmann

FORNEKTELSE
Til tross for at faget heter «programvareutvikling» er det lite programvare å vise til og svært lite utvikling som foregår. Allikevel er samtlige gruppemedlemmer i denne fasen overbevist om at alt kommer til å gå etter planen. Alle utviklingsoppgavene kommer til å bli fullført, alle brukerhistoriene gjennomført, og alle kravene til produkteier kommer til å bli innfridd.

SINNE
Gruppemedlemmene begynner å kjenne tidspresset. Det er ikke gruppens egen inkompetanse som har skylden for det elendige produktet. Neida, det er nok VS Code sin feil, eller de derre knappene fra Material UI som alle bruker som plutselig ikke fungerer lenger. Gruppens sinne blir rettet mot alt og alle. Også i det sterke båndet mellom gruppemedlemmene begynner det å slå sprekker. Han ene fra informatikk — som har kodet nettsider siden før han kunne gå — krangler med indøkeren om hvordan man bruker Git mest mulig effektivt. Han blir sint og påpeker at indøkeren så vidt har bidratt med én kodelinje i prosjektet. Indøkeren svarer: «Jeg har ikke kodet siden Java!» og vender snurt tilbake til rapporten som er 1000 ord over maksgrensen.

FORHANDLING
Det er altfor mye å gjøre og altfor lite tid. Til tross for krav om både parprogrammering og tett samarbeid, begynner gruppen nå å innse at de ikke har tid til slikt vas. Dessuten begynner man å innse at det å holde rapporten innenfor ordgrensen er svært vanskelig. Nå gjelder det å vurdere hvilke smidige utviklingsprinsipper man skal følge, hvilke man kan unngå ved å heller reflektere rundt dem i rapporten og hvilke man er nødt til å gjøre for å kunne korte ned på rapporten. Videre utfordres gruppemedlemmenes verdier. Hva er viktigst å ha med i rapporten, informatikerens detaljerte beskrivelse av teknologistakken eller indøkerens fullstendige analyse av brenn-diagrammet?

DEPRESJON
Innleveringsfristen nærmer seg med stormskritt. Hele tilværelsen har blitt fullstendig oversvømt av PU. Ingen klarer å spise, sove eller dekke sine andre livsnødvendige behov uten å ha gjort et par *commits* først. Maingreinen er fullstendig ødelagt, og ingen vet hvordan de skal fikse den. Selv ikke han supersmarte fra informatikk. Hele avsnittet du skrev i rapporten ble kuttet for å spare ord. Ja, det avsnittet du brukte tre hele dager på å skrive forrige uke. Det var visst ikke så relevant å reflektere rundt gruppens bruk av *gitmojis* i committmeldingene allikevel. Livet har nådd et nytt bunnpunkt.

AKSEPT
Alle innser at prosjektet ikke holder mål, men også at det ikke lenger er i stand til å reddes. Det er mange ufullstendige setninger i rapporten. Det mangler liksom noen ord her og der, men det går nok fint. Rapporten kom seg under ordgrensen i det minste, og sensor skjønner nok budskapet uansett. Det ferdigutviklede produktet er heller ikke så mye å vise til. Det er så vidt det kjører, men om du bruker `npm run`-kommandoen tre ganger og hopper over den ene problematiske databasetesten, så skal programmet i teorien kjøre på *localhost:3000*. Av og til kjører det på *localhost:6729* i stedet, ingen vet helt hvorfor. Det kjører, det er det viktigste.

Gløsløken

Napp meg i løken.

STUDENTER DOPET NED PÅ SAMFUNDET

Tekst: *Gammel Mannesand*

ST. OLAVS HOSPITAL – Flere studenter ved NTNU endte på legevakten etter en lørdag på Samfundet. Felles for samtlige er at de, kort tid etter å ha kommet på huset, følte et sterkt ubehag i munnhulen og fikk behov for å innta store mengder vann. Gløsløken møter en student som ønsker å være anonym fra sykesengen: «Alt gikk så fort. Jeg skulle bare ha meg noe å drikke på Edgar, og så husker jeg ikke så mye mer.» Toksikolog hos St. Olavs Hospital, Svein Forgiftesen, har kanskje svar til studenten: «Vi venter fortsatt på de siste resultatene, men så langt har vi en klar teori. Enten du har tatt det selv eller noen andre har gitt deg det, har du klart å få i deg Dahls fra plastkopp.»

NY KI KUTTER SKATTEN

Tekst: *Yoshisaur Munchakoopas*

SILICON VALLEY – Det er en ny spiller på banen i teknologiens Mekka, også kjent som Silicon Valley. Alle i Silisiumdalen snakker om Fraudster.ai og deres innovative kunstige intelligens tjeneste som demokratiserer skattefusk. Oppstartsbedriften påstår at snylting på skatt ikke lenger er forbeholdt samfunnets rikeste. Algoritmene deres analyserer skattemeldingen din og gir forslag til hvordan du best kan lyve for å unngå skatt uten å bli tatt. Fraudster.ai reklamerer med inntil 80% lavere skatt, de tilbyr i tillegg forsikring som dekker omkostninger om du mot formodning skulle blitt tatt for å oppgi uriktige opplysninger til skatteetaten.

ET ANNERLEDES KONSULENTFIRMA

Tekst: *Stresstle Hån Manglerarbeidslyst*

IKKE OSLO – Gløsløken har i mange år jobbet med å få oversikt over kamparenaen som er teknologiske konsulentfirmaer. Hver uke finner vi nye firmaer som prøver å skille seg ut ved å ha agile team, fokus på læring, og fri annenhver fredag. Det firmaene ikke er klar over er at dette ikke får dem til å skille seg ut, ettersom alle de andre firmaene har de samme tiltakene. Denne uken har vi tross dette endelig funnet et firma som greier å skille seg ut. Selskapet er «Pettersen og Co. programmeringshjelp AS Inc.» og kan flagge med egenskaper som «Vi er ikke tverrfaglige», «Vi setter menneskene bakerst» og «konkurranseudyktig lønn». Selskapet har også ukentlige aktiviteter som «overtidsfredag», «møtemandag» og «kritisk-feil-i-prod-søndag» for å skape uengasjement i de dysfunksjonelle teamene. Vi i Gløsløken anbefaler alle nyutdannede som er lei av alle konsulentklisjeene å forsøke en forfriskende arbeidsplass som dette.

p&c
For en verre arbeidsdag

Utgavens master

Spill mot kjønnsgapet i teknologi-bransjen

XploreIT: A Minigame Approach to Increase Girls' Awareness and Interest in ICT[1]

Tekst: *Cathrine Akre-Aas, utvikler i Bekk (M.Sc 2021) og Ingrid Kindem, utvikler i Askeladden & Co (M.Sc 2021)*

Teknologibransjen trenger flere kvinner og dette er noe vi brenner for. Derfor valgte vi å skrive en master rundt hvordan man kan motivere og inspirere unge jenter til å velge en teknologiutdannelse. Det å skrive en ren teorioppgave under korona tenkte vi fort kunne bli litt kjedelig, så derfor forsøkte vi å løse problemet ved hjelp av et spill.

Som en forsmak på masteren, skrev vi sammen en fordypningsoppgave om hvordan man kan designe et *serious game* (les: spill med pedagogiske eller informative mål) for tenåringsjenter, med overordnet mål om å minke kjønnsgapet i bransjen. Vi kom frem til fire læringsmål dette spillet skulle ha, i tillegg til spillelementer som viste seg å ha en positiv innvirkning på jenters spillopplevelse.

Da vi begynte på masteren, måtte vi grave litt dypere. For å få innsikt i effekten av kjønn i spillsammenheng, intervjuet vi en kjønnsekspert. Deretter brukte vi funnene fra fordypnings-oppgaven til å designe og arrangere en co-design workshop med kvinnelige universitetsstudenter. Denne måtte holdes digitalt på grunn av korona og dette er noe vi absolutt ikke anbefaler. I workshopen skulle deltakerne kombinere læringsmål, minispillmaler og spilldesign-retningslinjer for å lage minispill som oppfylte læringsmålene. Resultatet var ulike minispill-idéer som ble tatt med videre inn i XploreIT.

Vi utviklet XploreIT på plattformen til Attensi; et teknologiselskap som spesialiserer seg i utvikling av treningsspill og simuleringer for virksomheter. En spillekspert evaluerte den første prototypen gjennom et intervju, hvor tilbakemeldingene vi fikk ble brukt til å forbedre spillet.

Til slutt sto vi igjen med XploreIT, et mini-spillbasert serious game for tenåringsjenter, med mål om å øke bevisstheten og interessen for IKT. XploreIT ble evaluert av rundt 30 ulike jenter, som går på ungdomsskolen, gjennom digitale spilløkter. Spilldataen og svarene fra spørreskjemaene jentene svarte på før og etter å ha spilt XploreIT ble brukt til å si noe om spillets innvirkning på målgruppen.

Evalueringen av XploreIT viste at minispillbaserte serious games kan øke tenåringsjenters bevissthet om og interesse for IKT. Vi fant ut at bevisstheten og interessen økte mest blant de eldste tenåringene (15–16 år) sammenlignet med de yngre (13–14 år). Imidlertid økte bevisstheten og interessen også mer blant spillerne som presterte godt i spillet, muligens på grunn av økt mestringsfølelse. Vi skulle ønske vi kunne sagt noe om de langsiktige virkningene spillet hadde, men det var ikke mulig på ett semester. Derimot håper vi at noen av jentene som spilte XploreIT ender opp i Abakus om noen år.

[1] Informasjons- og kommunikasjonsteknologi

Maximillian August

Top 0,01% in Norway. Former curator of moneyflow in vital subeconomy[2]. #riseandgrind #everymondaymatters

Jobb: Profesjonell motivator, produktivator og vibe kurator

Studier: Datateknologi NTNU

Sted: *Oslo, Norway*

[1] Grünerløkka.

[2] I 2012 hadde jeg en meget suksessfull nettside for trading av FIFA-coins.

Maximillian August tester

Tekst: *Maximillian August Steinsrud*
Layout: *Øyvind Monsen*

Som student er det viktig å være tidlig ute med å få seg et nettverk slik at man ikke havner i feil sosial sirkel[1] når man går ut i jobb. Ifølge flere ekstremt oppegående menn, som jeg har en enorm respekt for, er det nettopp LinkedIn man da må benytte seg av. Jeg har derfor skapt en upåklagelig god LinkedIn profil og brukt denne til å utforske potensielle stillinger, samt se hva annet denne plattformen har å tilby.

Maximillian August

Without telling anyone, I humbly woke up today.

Suddenly, one of my old co-workers called me and said that he woke up too (exactly at the same time).

How did they know?

Was it a huge coincidence?

I think not.

Because when people wake up professionally, like I do, it's only natural that other people will follow.

My advice as an influencer?

Just aim high. The rest will happen on its own.

Maximillian August

The Human Centipede was a 2009 shock horror film about a German surgeon who got bored of separating conjoined twins and wanted to create them instead.

The film somehow managed to spawn 2 sequels. Here are 5 things the film taught me about product management 👇

Se mer

Maximillian August

Yesterday I was walking to an interview. There was a starving dog on the road. I stopped to feed him & missed the interview. The next day I got a call asking to come in to do the interview. I was surprised, but I went. Then the interviewer came in. He was the dog.

Maximillian August

Maximillian August

Maximillian August

Had a great meeting with the chief executive of the marine branch at NTNU this friday. We had a long discussion about the future of AI and big data in accordance with the salmon industry, and how it may affect marine life in the coming decade.

#BearWithMe #patagonia

Maximillian August

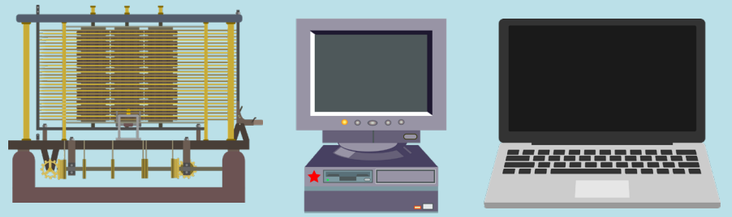
Just gave a great speech on the liquidity market cap in the IT and Finance sector for Patagonia today. I'm thankful for this great opportunity, and hope that some of my many points on the importance of global sustainability came across to my listeners.

Thank you to Patagonia for giving me the opportunity and my amazing fleece jackets.

#humblebrag #patagonia

Programmeringsspråk-LORE

Tekst: *Vetle Roos Mangrud*
Layout: *Petter Buseth*



Som programmeringsstudenter er vi nødt til å forholde oss til mange ulike programmeringsspråk. Det er lett å tenke at et språk kun er verktøy og syntaks, og glemme historien. Hvem er personene bak? Hva er *lore*-en til språkene? La oss børste støvet av floppy-diskene og se hva vi finner.

DET FØRSTE PROGRAMMET

På midten av 1800-tallet beskrev matematikeren Charles Babbage sine planer om å lage en generell datamaskin – «The Analytical Engine». Den hadde et komplisert design og ble derfor aldri bygget, men ideen var fortsatt interessant, spesielt for matematikeren Ada Lovelace. Det mest kjente verket hennes er en oversettelse av en artikkel av italieneren Luigi Menabrea. Sammen med oversettelsen la hun ved notater, som endte opp med å bli tre ganger så lange som oversettelsen av selve artikkelen, hvor hun blant annet prøvde å forklare The Analytical Engine. I notatet «Note G» beskrev hun et eksempelprogram som skulle regne ut en rekke Bernoulli-tall. Dette programmet blir regnet som verdens første, selv om det aldri har blitt kjørt. Lovelace kan derfor kalles verdens første utvikler, *slakt dronning!*

HØYNIVÅ TIL SJEFEN MÅTTE GÅ AV

Tyskeren Konrad Zuse har æren av å designe verdens første høynivå-programmerings-språk, *Plankalkül*. Høynivå vil si at språket var sterkt abstrahert vekk fra detaljene i datamaskin-arkitekturen. Språket var bygget på å utvide proposisjonslogikk til å støtte blant annet løkker og grener i logikken (tenk *if*-setninger). Programmeringen var mer diskmat og mindre ITGK, men det var et viktig steg. I 1945 sto tyske Zuse klar med en upublisert bok der han beskrev Plankalkül. Uheldigvis for Zuse er 1945 også året da andre verdenskrig ble avsluttet, og han var på det tapende laget. Arbeidsstasjonen hans ble bombet og de allierte forbød ham å bygge nye datamaskiner. Han fikk aldri æren han fortjente for språket, men blant hans andre oppnåelser finner man en av verdens første programmerbare elektroniske datamaskiner, Z3. Han er ganske legendarisk allikevel.

C-NY-SISTE-SISTE-FERDIG (3).DOCX.PDF

Basic Combined Programming Language, eller *BCPL*, var et språk som ble designet i 1967 for å kunne ha en veldig lett kompilator. Språket er ikke mye brukt i dag, men det fortjener allikevel noen poeng for å introdusere krøllparenteser i programmering og det første «Hello World»-programmet. To år senere lagde Ken Thompson og Dennis Ritchie en variant av BCPL som de trengte et navn til. Navnet de gikk for var «B», og er sannsynligvis en forkortelse av BCLP.

B skulle ikke være slutten på Ritchies hunger etter det perfekte programmeringsspråk. Han la gradvis til ny funksjonalitet til B, og det gikk gjennom navneendringer. Det het «New B» (NB) i en liten periode, men det skulle snart få sitt ultimate navn, «C».

Ritchies språkutviklingskarriere skulle slutte med C, men en danske og noen amerikanere syntes C enda ikke hadde nådd sitt fulle potensial. De ville utvide C for objektorientert programmering, og lagde hver sin variant. Amerikanerne Tom Love og Brad Cox valgte å kalle språket sitt «Objective-C», mens dansken Bjarne Stroustrup først gikk for det beskrivende navnet «C with Classes», som senere utviklet seg til «C++».

De som har god oversikt over programmeringsspråk med navn bestående hovedsakelig av bokstaven «C» vet at det fortsatt er et viktig språk igjen, og nei, jeg snakker ikke om Holy-C. Rundt år 2000 ble det skapt et nytt programmeringsspråk hos Microsoft som skulle støtte flere paradigmer, som objektorientert programmering og funksjonell programmering. Dette språket ble hetende «C#», som både symboliserer det musikalske symbolet kryss, som hever en note med en halvtone, og at det kan se ut som fire plusstegn. Sagt på en annen måte kan C# tenkes på som «C pluss pluss pluss pluss».

Man finner mye interessant i historien til programmeringsspråk, og utviklingen er ennå pågående. For eksempel jobber Google med Carbon, et språk som skal revolusjonere C++ nok en gang. I tillegg har vi relativt unge språk som Rust og Go som konkurrerer om å bli de mest populære språkene. Det er verdt å følge litt med på prosjekter som skal gjøre livet vårt enklere.

Foto og layout: *Ian McGregor Aksum, Øyvind Monsen og Emmanuel Quarcoo*

lunsj AMEN

#116
TING DU BØR VITE SOM SIVING

AS A SERVICE

Tekst: *Vegard Ervik*

I teknologiverden dukker det opp nye fagbegreper raskere enn noe annet sted. Som teknologistudent må du passe på å ha kjennskap til brorparten av disse for å kunne vise muskler på intervju og for å kunne mingle med mennesker i jobb. Noen av fagbegrepene som har skutt fart den siste tiden er *IaaS*, *PaaS* og *SaaS*, og i denne artikkelen skal du få vite hva de betyr.

FAGBEGREP «AS A SERVICE»

Før vi kan begynne å snakke om hva SaaS, PaaS og IaaS er, så kan vi starte å finne ut av hva «as a Service» betyr. Det som kanskje gjør «as a Service» forvirrende, er at «cloud» burde vært en del av begrepet uten å være det. For når folk snakker om «as a Service» snakker de nemlig om noe som finnes på skyen, en skytjeneste. Nå som vi vet at geniene som fant på «as a Service» gjemte skybegrepet, kan vi diskutere de ulike aaS-ene.

PROGRAMVARE SOM EN TJENESTE

Software as a Service, eller SaaS, betegner programvare, bare at du ikke trenger å installere noe som helst. Det vil si programvare du enten kan besøke via nettleseren eller ha som en *Progressive web Application* (PwA) – en app med minimal installasjon. Vanskeligere enn det er det ikke. Eksempler på SaaS er LinkedIn, Dropbox og Age of War. I en SaaS er alt inkludert, som lagring, programvaren og hosting.

PLATTFORM SOM EN TJENESTE

Platform as a Service, eller PaaS, er dritten i midten mellom SaaS og IaaS. PaaS er ikke like populært som hverken SaaS eller IaaS, men fagbegrepet er likevel greit å vite om. En PaaS er en programvare som lar utviklere programmere en webapplikasjon uten å bry seg om hosting, lagring og alt annet backendere skryter av å være gode på. Eksempler på PaaS er Heroku, Vercel og Google App Engine.

INFRASTRUKTUR SOM EN TJENESTE

Infrastructure as a Service, eller IaaS, er en tjeneste som lar deg bruke prosesseringskraft, hosting og minne via skyen. Med andre ord lar en IaaS deg ha en backend på maskiner og servere du ikke eier selv. Ofte tilbyr også IaaS mange tjenester som gjør det lettere og mer attraktivt å bruke deres verktøy fremfor å sette opp hele backenden selv. Eksempler på IaaS er Google Cloud, Amazon Web Services (AWS) og Microsoft Azure.

HVORDAN HENGER DET HELE SAMMEN?

Felles for alle begrepene er at de lar deg gjøre ting via skyen. For å ha en visuell modell av hvordan begrepene henger sammen, kan du se for deg at begrepene utgjør en pyramide, der IaaS er nederst, PaaS er i midten og SaaS er på toppen. Grunnen til dette er at alt du kan gjøre med en SaaS kan du få til med en PaaS, og alt du kan få til med en PaaS kan du gjøre med en IaaS. Så dersom du har et ønske om å lage Mujaffas BMW (SaaS) i Google Cloud (IaaS), i stedet for å bare spille det i nettleseren, er det ingenting som stopper deg. Det eneste negative er at det sannsynligvis er lettere å bare spille det.

104
TING DU IKKE TRENGER Å VITE SOM SIVING

PUSLESPILL

Tekst: *Solveig Heitmann*

De aller fleste abakuler har nok lagt et puslespill i sitt liv. Et puslespill kan for eksempel være en morsom familieaktivitet på en søndagskveld, eller et nyttig verktøy for å forbedre et småbarns logikk og problemløsningsevne. Kanskje aller oftest er det en perfekt hobby for å ta tankene vekk fra hverdagsstress ved å kun fokusere på brikkene foran seg. Men hva er egentlig historien bak puslespill og hva skjer i puslespillenes verden i dag?

HVA ER HISTORIEN?

Puslespillet begynte sin historie allerede tilbake i 1762 da den engelske gravøren John Spilsbury skjærte ut de individuelle landene fra et verdenskart i tre. Ønsket hans var at studenter skulle legge puslespillet for å bedre lære seg geografi. Oppfinnelsen hans slo med en gang godt an, og man begynte å lage puslespill med mange forskjellige motiver. Puslespill fikk virkelig sin opptur under de harde trettiåra, da folk trengte noe kreativt å gjøre for å få tankene vekk fra den vanskelige hverdagen de levde i. Det er verdt å merke seg at på engelsk heter puslespill «jigsaw puzzles». «Jigsaw» er, som navnet tilsier, en type sag oppfunnet i 1880. Navnet kommer av at denne sagen tidligere ble brukt til å skjære ut puslespillbrikker, og selv om den ikke lenger brukes i dag, sitter navnet fortsatt igjen.

VERDENSMESTERSKAP

De fleste tenker nok på pusling som en koselig aktivitet hvor man tar ting i sitt eget tempo og har som mål å ha det gøy, men visste du at pusling faktisk er en konkurransesport? Det er kanskje ikke så rart om du aldri har hørt om verdensmesterskapet i pusling, da det først så dagens lys i 2019, og har på grunn av COVID-19 kun blitt holdt igjen i 2022. Mesterskapet er delt inn i tre kategorier etter hvor mange som pusler sammen: individuell, par og lag (3-4 stykker). Det er litt forskjellige krav i hver kategori, men kort sagt har de alle en viss tidsgrense og et visst antall puslespill av varierende størrelser som skal legges. I den individuelle kategorien er det en innledende runde, semifinale og finale hvor de i hver runde skal legge et puslespill på 500 brikker på under 90 minutter. Om du synes dette høres enkelt ut har jeg gode nyheter til deg! Hvem som helst kan nemlig melde seg på dette verdensmesterskapet. Det er faktisk 17 nordmenn påmeldt allerede, så om du har litt ledig tid i september er det bare å reise ned til Spania og gi det et forsøk. Kanskje blir du den heldige vinneren som stikker av med førstepremien på nesten tolv tusen kroner?

60 000 BRIKKER SA DU?

Verdens største puslespill er på hele 60 000 brikker og heter «What a Wonderful World». Det ferdiglagte puslespillet er 2,4 meter høyt og 8,8 meter bredt. For å sette størrelsen i perspektiv for en student vil det si at puslespillet er tre ganger så stort som en sju kvadratmeters hybel. Ikke bare er puslespillet stort, men også relativt tungt med sine 34 kilo. Å legge et puslespill med så mange brikker høres helt umulig ut, men det mange ikke vet er at i puslespill med over 9000 brikker deles puslebrikkene opp i seksjoner som hver kommer i en egen pose. Et puslespill på 9000 brikker kommer for eksempel i to poser på 4500 brikker hver, mens verdens største puslespill kommer i 60 poser på 1000 hver. Å legge 60 puslespill på 1000 brikker vil kreve mye tid, tålmodighet og ikke minst fysisk bordplass, men kanskje en abakule tør å ta utfordringen?

Så jeg hører du vil bli

backend-utvikler

Tekst: Magnus Hønesand
Layout: Karoline Sæbø

I forrige utgave lærte du om frontendutvikling. Denne gangen er det backendutvikling som står for tur. Finn frem gudekomplekset, en lunken hvit Monster og kutt all sosial kontakt – her kommer en kort intro til det viktigste du trenger å vite som backendutvikler.

Relasjonsdatabaser

De færreste applikasjoner fungerer uten å bruke data de har lagret. Det kan være snakk om alt fra brukerdata, til innlegg i et forum eller produkter til salg i en nettbutikk. Den vanligste måten å holde denne dataen lagret på er med *relasjonsdatabaser*. Det finnes flere forskjellige typer relasjonsdatabaser, men de mest populære har til felles at de benytter seg av spørrespråket SQL. Du vil typisk kjenne igjen dette på navnene deres, for eksempel MySQL, PostgreSQL eller SQLite.

Entiteter, attributter og relasjoner

Relasjonsdatabaser består av *entiteter* med *attributter* og *relasjoner* mellom entitetene. En entitet er en generalisering av en mengde data, mens attributter er data som typisk hører til en gitt entitet. Et eksempel kan være entiteten «User», som er en generalisering av alle brukere. Denne entiteten har attributter som «username» og «email». Et annet eksempel kan være alle innleggene brukere legger ut i et sosialt medium. Entiteten «Post» kan for eksempel ha attributtene «caption» og «publishing_date».

Nøkler

Enhver bruker og ethvert innlegg trenger at et av attributtene fungerer som en unik identifikator. Dette attributtet kalles en *primærnøkkel*. Et eksempel kan være e-postadresser, ettersom ingen har samme. Hvis det ikke naturlig finnes et slikt attributt, er det vanlig å bare finne på et attributt der man aldri gjentar samme verdi, for eksempel «UserID».

I et sosialt medium med brukere som legger ut innlegg, vil entiteten «Post» (for innlegg) behøve et attributt som refererer til brukeren som lagde innlegget. Ved å sette attributtet lik primærnøkkelen til en bruker, definerer man en såkalt *fremmednøkkel* og har dermed dannet en relasjon mellom en bruker og et innlegg.

Det finnes databaser som ikke er relasjonsdatabaser også – såkalte NoSQL-databaser er enten døgnfluer eller fremtiden, helt avhengig av hvem du spør. Eksempler som MongoDB, DynamoDB og Apache Cassandra er relativt populære, men med mindre du går data med spesialisering i Databaser og søk kan du fint komme deg gjennom hele studiet uten å røre en NoSQL-database.

HTTP

Kommunikasjon av data på nett følger protokoller for at spørringer og svar skal foregå ryddig. Den desidert vanligste heter HTTP (eller HTTPS, som er den sikrere varianten). Når du surfer på nettet, er det nettleseren din som gjør HTTP-spørringer til URL-er. Den samme protokollen brukes også når frontenden skal spørre om data fra backenden.

I HTTP er det definert en rekke forskjellige typer spørringer med forskjellige konvensjoner og funksjonaliteter. De fire desidert vanligste heter GET, POST, PUT og DELETE og brukes henholdsvis til å lese, opprette, oppdatere og slette data. Når nettleseren din gjør en spørring om å se en nettside, sender den en GET-spørring og får HTML-koden tilbake så nettleseren vet hva den skal vise deg. På samme måte sender frontenden i en webapplikasjon GET-, POST-, PUT- eller DELETE-spørringer for å gjøre sine respektive operasjoner på databasen.

API

Så du har en database med data, du har en frontendapplikasjon som skal bruke den, og du har en grunnleggende forståelse av HTTP. Neste steg nå er å bygge et API; bindeleddet mellom databasen og frontendapplikasjonen. API står for «application programming interface» og er en veldig vanlig del av en backendapplikasjon. Å kunne bygge et er minstekravet for å kunne flexe tittelen «backendutvikler» på CV-en. Tanken med et API er at backendapplikasjonen skal presentere data fra databasen på en måte som frontendapplikasjonen lett kan forstå. Backenden skal lytte etter HTTP-spørringer til URL-en sin og svare med data på et format frontenden kan lese.

REST

I en webutviklingskontekst vil du ofte høre begrepet REST API. Begrepet REST står for «representational state transfer» og beskriver en stil å bygge API-er i. Mange API-er kalles RESTfulle uten å faktisk være det – som fersk backendutvikler er det viktigste at du skjønner at når noen snakker om et REST API, snakker de om API-er som sånn ish ser ut som det som står forklart i denne artikkelen.

Endepunkter

API-et deler, eller *eksponerer*, data på forskjellige *endepunkter*. Endepunkter er spesifikke URL-er man i API-et definerer som «møtepunkter» der frontenden kan møte API-et i form av HTTP-spørringer. For eksempel kan API-et definere at når det sendes en GET-spørring til URL-en som slutter på /user/, vil man få en liste over brukere i databasen. Hvis man derimot vil opprette en ny bruker, sender man brukerinformasjonen i en POST-spørring til samme URL.

JSON

Brukerinformasjonen som både mottas og sendes av frontendapplikasjonen, må være på et format som begge parter forstår. Det finnes flere alternativer, blant annet XML, men den desidert mest populære standarden heter JSON – «JavaScript object notation». JSON er designet for å passe godt med JavaScript-applikasjoner og kan brukes som format både på det man sender i spørringen og det man får som svar. Om man for eksempel skulle opprettet en ny bruker i databasen fra tidligere i artikkelen, ville man sendt noe slikt til API-et.

```
{
  ”email”: ”readme@abakus.no”,
  ”username”: ”readme”,
  ”password”: ”PRsuger”
}
```

Arkitekturmønstre

Mange av de mest populære rammeverkene vi bygger API-er i nå, stammer fra en tid da det var vanlig at de også tok seg av brukergrensesnittene til webapplikasjoner. Det vanligste *arkitekturmønstret* man bygget disse applikasjonene i kalles MVC, oppfunnet av nordmannen Trygve Reenskaug. MVC er et mønster man vil kjenne igjen fra andre typer applikasjoner også, men kjernen er at applikasjonen består av tre hoveddeler: modell, visning og kontroller, eller på engelsk «model, view and controller». Kontrollerne er delen av koden som definerer endepunktene, altså delen brukeren interagerer med. Modellene er delen av koden der strukturen på dataene er definert, og delen kontrollerne snakker med. Visningene er det brukeren blir møtt med rent visuelt. Over tid har sofistikerte frontendrammeverk og JSON i stor grad tatt over jobben til visningene, men man vil fortsatt støte på MVC, eller noe som likner, i mange populære rammeverk.

ORM

Et siste konsept man ofte møter på – uansett hvor mye Svein Erik Bratsberg hater det – er ORM-er. ORM står for «object-relational mapping», en teknikk som knytter databaseobjekter til objekter i det programmeringsspråket du jobber i. Slik kan man til det fulleste utnytte fordelene av å jobbe i et objektorientert språk selv om dataene er definert i en database. Eksempler på ORM-er er Entity Framework til ASP.NET, ActiveRecord til Ruby on Rails eller Hibernate til Java-baserte rammeverk.

Språk og rammeverk

I motsetning til Reacts stilling i frontendverden, er det ingen språk eller rammeverk som er *soleklart* mer populære enn andre innenfor backendutvikling. Derfor tar ikke denne guiden for seg noen spesifikke språk eller rammeverk. Rammeverkene varierer ofte veldig i kompleksitet og fyller forskjellige behov. I emner som TDT4140 – Programvareutvikling og IT2810 – Webutvikling er det vanlig å bruke enkle rammeverk som Django skrevet i Python og Express.js skrevet i JavaScript. Noen populære og litt mer omfattende rammeverk inkluderer ASP.NET i C# eller F#, Spring i Java eller Kotlin, Laravel i PHP eller Ruby on Rails i Ruby.

Jobb

Som nylig utlært backendutvikler bør du nå være helt klar for å kapre denne introduksjonsartikkelens siste konsept: «jobb». Den er ganske bankers, i hvert fall frem til GPT-5 kommer. I mellomtiden kan du bruke hvert eneste våkne minutt på å føle deg bedre enn kafé-hipsterne som sitter med Macen sin og jobber med React.

Nyt gudekomplekset ditt, lykke til!

Brød á la readme

Hvorfor burde akkurat du kunne bake brød, mon tro? Jo nå skal du høre. Med få ingredienser og enkelt utstyr kan du minske avstanden mellom herremåltidet du ble servert på en bedriftspresentasjon forrige uke, og maten ditt stakkarslige stipend kan gi deg. Brødbaking gir deg også evnen til å imponere alle skapninger med smaksløker, fra din romantiske interesse til din mor. Så hvorfor ikke prøve det?

Jeg har saumfart YouTubes beste kulinariske skapere (blant dem Andrew Rea fra Binging with Babish, Adam Ragusea, J. Kenji Lopez-Alt og Joshua Weissmann) for å finne den beste null-tull-metoden for å lage brød. Merk deg at dette utvalget bare inneholder menn – av en grunn (for å bryte med historisk dogma, selvfølgelig). Ønsker du å reise deg til nivået disse gigantene lever på, må du kaste deg inn på kjøkkenet, få på et forkle, og brette opp ermene.

Tekst og foto: *Ian McGregor Aksum*

Layout: *Madelen Hellevik Lothe*

OPPSKRIFT:

UTSTYR

For denne oppskriften trenger du enten en gryte med lokk i enten ovnfast metall eller keramikk, eller en bolle du kan vende over deigen. Du burde sjekke at disse kan gå i ovnen. Dette er for å fange damp i samme rom som deigen baker i, slik at skorpen ikke stivner for tidlig, og brødet kan ekspandere og bli luftig.

INGREDIENSER

250 g mel
200 g vann
2 klyper tørrgjær
½ ts salt
1 ts sukker, honning eller sirup

FREMGANGSMÅTE

KVELDEN FØR

1) Bland mel, vann, gjær, salt og søtning i en bolle, og rør sammen med en skje. Dekk med lokk og sett i kjøleskapet eller på benken.

PÅ DAGEN

2) Dryss litt mel over deigen og «skrell» den fra kantene på bollen. Strekk og brett deigen et par ganger mellom hendene, og form den til slutt til en ball. La stå i minst 20 minutter eller opptil en time, gjerne på et varmt sted.

3) Gjenta steget over så mange (eller få) ganger du vil for ekstra keege glutenkjeder.

4) Sett ovnen til 275° C med over- og undervarme. Bruker du en gryte kan du forvarme denne ved å sette den inn nå.

5) Form deigen til en ball med glatt overflate, samle «sømmer» på undersiden, og legg på bakepapir. På dette stadiet kan du la deigen heve litt til, men dette er ikke nødvendig. Skjær noen snitt i overflaten med en ekstra skarp eller taggete kniv for Instagram-verdige bilder.

6) Løft bakepapiret ved hjørnene for å plassere deigen i gryten (pass fingrene!), og sett lokket oppå, eller plasser bakepapiret på et brett med metallbollen over. Pass på at deigen ikke kommer i kontakt med gryte- eller bolleveggene. Senk temperaturen til 250° C og stek i ovnen i cirka 30 minutter. Ta av lokket eller bollen og la brødet steke videre i ti minutter, gjerne ved 25–50° C lavere temperatur.

GENERELLE TIPS:

- Baketiden vil variere med ovn og om du bruker gryte eller bolle, så ikke ikke vær redd for å justere tiden etter brødets utseende. Prøv å slippe ut så lite damp som mulig når du sjekker.
- Hvis du er usikker på om brødet har bakt nok, er det bedre å la det gå litt lenger enn du tror, ettersom en ekstra sprø skorpe er bedre enn en deigete innside.
- Hvis du krøller sammen og flater ut bakepapiret før du plasserer deigen vil papiret enkelt forme seg til gryten når du løfter det inn.
- Du kan forvarme grytelokket i ovnen og varme gryta på komfyren hvis du er redd for å brenne fingrene på den glovarme grytekanten når du løfter deigen inn.
- Du kan bruke gafler for å løfte metallbollen når den er varm.

BIBLIOTEK SYSTEMER

Bibliotek-Systemer As lager det ledende datasystemet i bruk i bibliotekene i landet. Vi ser etter deg som kan arbeide med database- og serverutvikling. Les mer om hva vi kan tilby:

GPU

Grafikkort, også kalt skjermkort eller GPU, er en viktig komponent i moderne datamaskiner. Opprinnelig var de utviklet for gaming, men i dag brukes de til mange applikasjoner hvor høy ytelse kreves.

GRAFIKKORTETS PLASS I DATAMASKINEN

Grafikkortet produserer et signal som sendes til en skjerm gjennom for eksempel HDMI for å vise et bilde. Grafikkortet brukes også gjerne til å akselerere beregninger som kreves for visuelle programmer som dataspill. En prosessor, også kalt CPU, er kjernen i en datamaskin og er laget for å kunne gjøre alle mulige beregninger. En GPU er mindre fleksibel enn en CPU, men er optimalisert for å gjøre simple og repetitive beregninger mye raskere. Laptoper og smarttelefoner har som regel ikke plass til et dedikert grafikkort, men de har en mye mindre GPU innebygd i CPU-en istedenfor.

FRA FILM TIL GRUVEDRIFT

Grafikkort brukes til å lage animasjonsfilmer, så med andre ord kan du takke grafikkort for at Shrek 2 eksisterer. I nyere tid blir også grafikkort brukt for maskinlæring og kunstig intelligens siden de kan utføre parallell behandling av store mengder data mye raskere enn CPU-er. Grafikkort har også blitt brukt for blokkjede *mining* hvor folk fylte rom med grafikkort for å tjene penger på å finne nye blokker til kjeden. Mining økte etterspørselen etter grafikkort drastisk. Dette førte til at grafikkort har vært dyre og vanskelig å få tak i. Heldigvis er GPU mining ikke lenger lønnsomt og etterspørselen har sunket.

TO GOLIATER OG EN DAVID

Det har lenge kun vært to produsenter av grafikkort: AMD og NVIDIA, men Intel har nylig også begynt å produsere grafikkort. Disse produsentene har laget mange forskjellige grafikkort med forvirrende navn som det kan være veldig vanskelig å skille mellom. En klassisk nybegynnerfeil er å sammenligne de forskjellige kortene ut fra hvor mye VRAM de har, da dette ikke gir et helhetlig bilde av hvor kraftig et grafikkort er. Hvis du vil bruke grafikkortet til dataspill burde du sammenligne *benchmarks* av forskjellige kort i de spillene du ønsker å spille. Benchmarks tester hvor ofte grafikkortet er i stand til å oppdatere bildet på skjermen, man ønsker gjerne at dette holder seg over 60 *FPS*[1]. For maskinlæring har NVIDIA desidert best støtte og er så å si det eneste aktuelle valget i skrivende stund, men til gjengjeld er de gjerne litt dyrere enn ekvivalente kort fra AMD og Intel.

[1] «Frames per Second» som er antall oppdateringer av skjermbilde per sekund

Tekst: *Emir Derouiche*
Layout: *Tobias Skjelvik*

Er DU

Scann QR-koden og ta undersøkelsen!

Tekst og layout: *Martine Mansåker*

den jevne abakule?

Utgavens Algoritme

Med: Vegard Ervik

Genetisk algoritme

En genetisk algoritme, eller en GA, er en maskinlæringsalgoritme som tar inspirasjon fra naturlig utvalg. I sin enkleste form bruker den ikke noe nevralt nett og er relativt enkel å forstå.

NATURLIG UTVALG

Se for deg at du vil lære en flokk med kråker å traversere en labyrint. For å gjøre det gir du alle kråkene ett eller flere forsøk på å komme seg gjennom labyrinten. Deretter velger du de beste kråkene, parer dem, og prøver igjen med det nye avkommet. Det nye avkommet har også kanskje fått noen mutasjoner. Etter mange nok generasjoner har du forhåpentligvis en populasjon med kråker som er født til å traversere labyrinter.

SURVIVAL OF THE FITTEST – DENNE GANGEN MED KODE

Se for deg det samme problemet som i avsnittet over, eneste forskjellen er at i stedet for å navigere seg gjennom en labyrint skal man finne toppunktet på en graf, og i stedet for kråker så har man bitstrenger – strenger med bare «0» og «1».

La oss si at du skal finne toppunktet av en funksjon, for eksempel $f(x) = x^2$ der definisjonsmengden er [0, 64], altså må x-verdien på grafen være et sted mellom 0 og 64. Av den grunn kan individene våre være bitstrenger med lengde 6, slik at de kan representere alle heltall mellom 0 og 64. For å starte med første generasjon bitstrenger, generer vi et satt antall tilfeldige bitstrenger – for eksempel 10.

For å fortsette eksempelet med kråkene, må vi ha en måte å måle hvor bra individene i populasjonen vår gjør det. Dette kalles *fitness*. Å sjekke fitnessen til et individ i denne sammenhengen er relativt greit. Dersom individet er 001011 vil fitnessen bli

$$f(001011) = f(11) = 11^2 = 121$$

Etter at du har sjekket fitnessen til alle individene, velger du de beste som foreldre før en ny generasjon. Resten forkastes. De foreldrene skal pares – på fagspråket kalt *crossover* – for å skape en ny generasjon. Dette kan du enkelt gjøre ved å ta to foreldre, velge et skillepunkt, og bruke dette for å skape to nye barn, som vist på illustrasjonen under.

Chromosome1	**11011**|00100110110
Chromosome2	**11011**|11000011110
Offspring1	**11011**|11000011110
Offspring2	**11011**|00100110110

Deretter har hvert barn en sjanse for å få en mutasjon, i dette tilfellet en sjanse for å flippe et bit, altså gjøre en «1» til en «0» eller omvendt.

Nå som du har skapt en ny generasjon, gjør du bare akkurat det samme om igjen og om igjen helt til du er fornøyd med resultatet eller du har kjørt nok generasjoner.

ANDRE VARIASJONER

Dette bruksområdet av en genetisk algoritme er veldig enkel og ikke særlig nyttig. Men genetiske algoritmer kan brukes til mye mer enn å finne toppunktet på en graf. Den funker for eksempel ypperlig til å finne en god løsning på Traveling Salesman Problem. Verdt å vite er at da vil sannsynligvis individene ikke bli representert som en bitstreng, og crossover og mutasjoner vil skje litt annerledes. Prinsippene vil likevel være de samme.

Dersom dette virker interessant og spennende så anbefaler jeg alle å ta valgfaget «IT3708 – Bio-inspirert kunstig intelligens» i fjerde klasse. Man må absolutt ikke gå AI for å gjøre det bra i emnet, og man lærer mye interessant om optimering i faget.

Genus

Lykke til med eksamen

GENUS NO-CODE

Vi gleder oss masse til å jobbe med årets 16 interns i sommer. Det blir spennende prosjekter og sosiale aktiviteter hver uke!

Interessert i sommerjobb i 2024?
Søk i august.

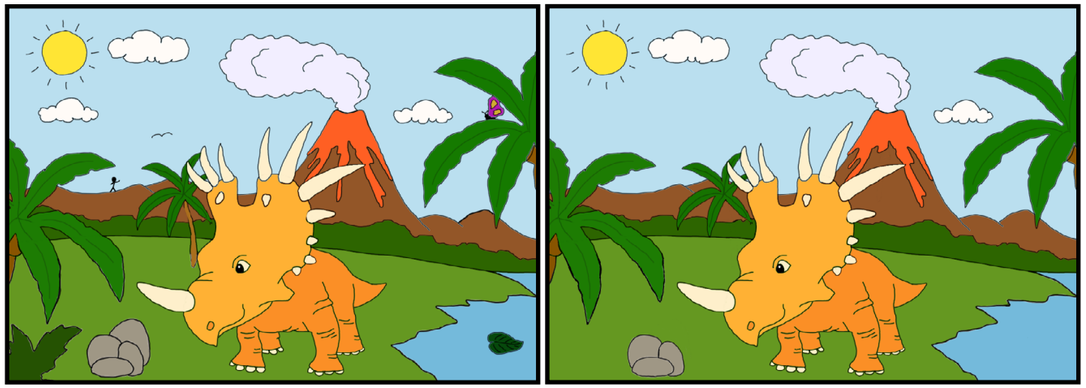
gen.us | gen.us/apply | @genusbiz

Utgavens konkurranse

Finn(s) ti feil

Tekst: *Martine Meng Karlsen*

Finn tar faget TREX4100 og har valgt å skrive eksamenen sin om Triceratops. Han skal blant annet skrive om hvordan klimaet har endret seg i løpet av tiden Triceratops vandret på jorden. I den forbindelse tegnet han to illustrerende bilder av ulike klima som han skal sammenligne. Problemet er bare at han skal skrive om minst ti ulikheter, men klarer ikke finne dem. Hjelp Finn med å unngå kont ved å sette ring rundt minst ti av ulikhetene som skjuler seg i bildet. Send inn bildet med ulikhetene markert til *konkurranse@abakus.no* med emne: «FTF»



Svarer du riktig er du i tillegg med i den årlige trekningen av en middag for to til en verdi av 2000 kr. *sponset av Genus*

VINNERE

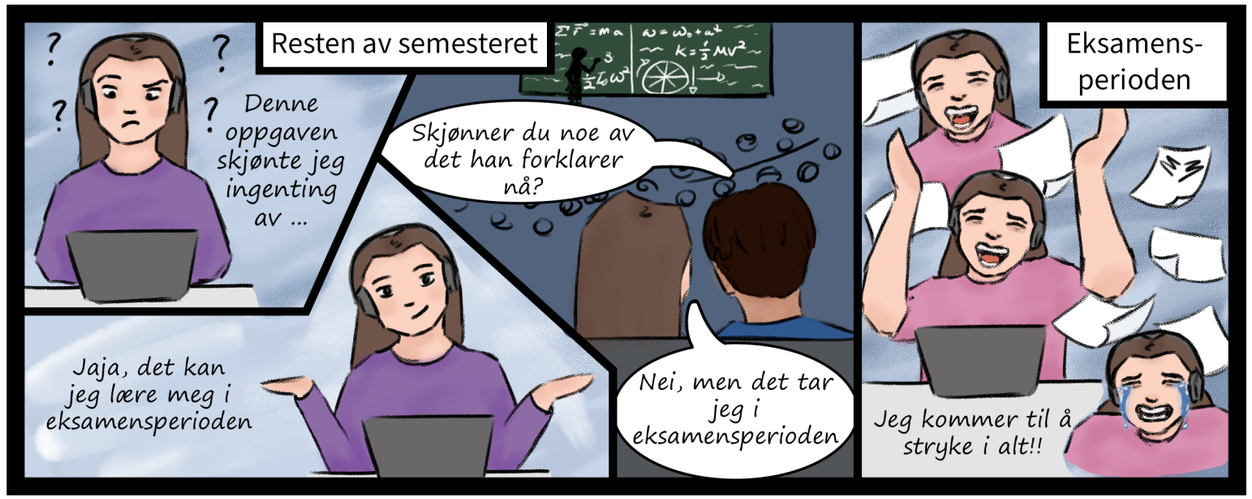
1. Andreas Gravrok
2. Eirik Hoem
3. Daniel Yang Hansen

PREMIER

1. premie

2. premie

3. premie



hett & nytt

Vil du ha snap på baksiden?
Send snap til **RYKTE**!

STRIPPET FOR PENGER

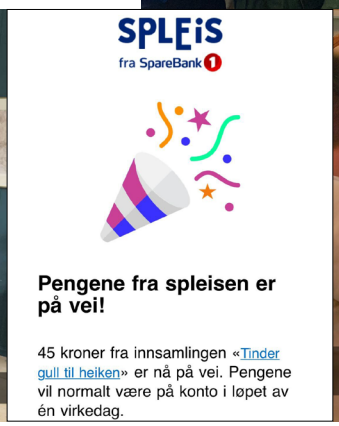
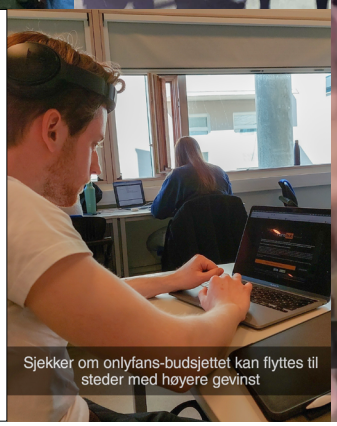
En ung abakules reise til det store utland gikk ikke helt som planlagt. Etter å ha fått i seg det abakulen har beregnet til å være 20 shots tok han raskt lommeboka fatt slik en full abakule vanligvis gjør. Dette skulle vise seg å være et feilsteg, da abakulen endte opp på en strippeklubb der eieren, den såkalte «pimpen», klarte å belaste abakulen over 10 tusen kroner for én time alene med en stripper. Dette plukket heldigvis banken opp, og kortet ble sperret snarest, men ikke før minst 700 kroner allerede var brukt på en lapdance. Hvorvidt kvaliteten på dansen var verdt pengene er diskutabelt, men det kan nevnes at vår trøtte kriger befant seg kastende opp på do i baris i lang tid etterpå. Det endte til slutt med at den rivde mannen «aggressivt» ble hentet av gjengen sin, som dro han ut av toalettet og trygt fikk båret han hjem. `readme` håper fyllesyken ga seg før det var gått en uke, og oppfordrer vennene til å aldri la abakulen få nærme seg et kredittkort igjen.

KORT PÅ VILLE VEIER

`readme` erfarer at kort til HS-kontoen nylig var å finne sammen med PIN-kode på kontoret. Redaksjonen reagerer på at det var *dette* som var strategien for å bruke mer penger på aktive abakuler. Uansett motivasjon er `readme` glade kortet ble lagt igjen på A3, og ikke på kontoret til Berg linjeforening.

GRATULERER MED PUNGEN

Pikk eller pung? Spørsmålet har fascinert mang en abakule. Dermed kom det kanskje ikke som den største overraskelsen da et par yngre abakuler tok FaceTime fatt og ringte opp en pensjonert abakule utenfor festlokalet sent en lørdags kveld. Håpet deres var å få innblikk i hans visdom angående det gitte spørsmål. Den pensjonerte abakulen virket ikke overrasket over å ha blitt oppringt sent denne lørdagskvelden, men var derimot tatt på senga av det fascinerende spørsmålet. Visdommen ble dog likevel spredt, og de yngre abakulene tok godt fornøyd turen tilbake til lokalet. De ble derimot meget flaue da en av festens gjester påpekte at det var *bursdagen* til den pensjonerte abakulen. `readme` ønsker den pensjonerte abakulen gratulerer med overstått, og håper dagen var fylt med litt mer enn pikk og pung.

NAVNEBONANZA

Nylig avgått Fagkom-leder skal etter sigende ha følt på en veldig sterk og plutselig forkjærlighet for komiteen hun inntil nylig har ledet. Klokken 12:14 på en helt vanlig onsdag sendte hun dermed søknad til Skatteetaten om å endre navnet sitt – vel vitende om at hun med det må vente ti år før hun kan endre navnet sitt igjen. Søknaden ble godkjent med stil, og `readme` kan nå gratulere Helene Fagkom Bjørnsen med et flunkende nytt mellomnavn.