

URI: *urn:nbn:se:bth-17900*



Programmering i skolmatematiken

problemlösning eller problemskapande

Klas Wärnå

juni 2019

Faculty of Computing
Blekinge Institute of Technology
SE-371 79 Karlskrona Sweden

This diploma thesis is submitted to the Faculty of Computing at Blekinge Institute of Technology in partial fulfillment of the requirements for the diploma degree in Software Engineering. The thesis is equivalent to 10 weeks of fulltime studies.

Contact Information:

Author:

Klas Wärnå

E-mail: klas.warna@gmail.com

University advisor:

Universitetsadjunkt Andreas Arnesson

Department of Computer Science

Faculty of Computing
Blekinge Institute of Technology
SE-371 79 Karlskrona, Sweden

Internet : www.bth.se
Phone : +46 455 38 5000
Fax: +46 455 38 50 57

Abstrakt

Denna uppsats handlar om hur programmering som verktyg i matematisk problemlösning har implementerats sedan det infördes som obligatoriskt i flera av gymnasiets mattekurser hösten 2018. Syftet är att reda ut vilka programmeringsspråk och programbibliotek som används ute i klassrummen och hur det upplevs av elever och lärare med avseende på svårighetsgrad, nytta för matematiken och om det bidragit till ett ökat intresse för programmering i allmänhet hos eleverna. Även eventuella könsrelaterade skillnader i dessa frågor har undersökts, då en av de politiska intentionerna med införandet varit att en förhoppning att främja flickors it-intresse.

Arbetet tar sitt teoretiska avstamp i en genomgång av den politiska bakgrunden till införandet, tidigare studier i ämnet och en kort historik om programmeringens roll i läroplanerna.

Med hjälp av digitala enkäter har fallstudier företagits i två klasser från olika gymnasieskolor i Skåne, där elever och lärare fått ta ställning till antal påstående genom att gradera på en skala hur väl påståendena stämmer samt komplettera med eventuell textkommentar. Enkäterna var anonyma men utformade på ett sådant sätt att lärare och elever kunnat kopplas samman klassvis.

Av resultatet har framkommit att många elever visserligen uppfattar programmeringen som svår och inte anser att den underlättar förståelsen för matematik. Men det framkom också att skillnaden var stor i dessa frågor och svaren tycks hänga samman med lärarens tidigare erfarenhet av programmering och i vilken omfattning programmeringen lärts ut. Intresset för programmering i allmänhet tycks inte ökat nämnvärt p.g.a. att det ingår i matematiken, men det låter sig heller *inte* entydigt sägas om någotdera könet var mer positivt eller negativt inställt till programmeringen. En något större andel tjejer än killar valde det allra mest negativa alternativet i enkätfrågorna, men klumpar man samman de mest negativa respektive de mest positiva alternativen på varsin halva om mitten, var tjejerna å andra sidan något överrepresenterade bland de positiva alternativen.

Python verkar vara ett lämpligt programmeringsspråk att använda. Det är viktigt att tillse att lärare har tillräcklig kunskap i programmering och att omfattningen i utlärandet av programmering inte blir alltför ytlig, för att eleverna skall tillägna sig nyttan av det. Obligatorisk programmering i matematik kan potentiellt bidra till att fler för upp ögonen för ämnet och på sikt minska den sneda könsfördelningen bland yrkesprogrammerare.

Nyckelord: *programmering, matematik, undervisning, gymnasiet, programmeringsspråk, digitalisering, digital kompetens*

INNEHÅLL

1. Inledning	5
1.1 Syfte	6
1.2 Frågeställningar	7
1.3 Avgränsning	8
2. Metod och genomförande	9
2.1 litteraturstudium	9
2.2 Empiriskt studium	10
2.2.1 Enkätundersökning med graderingsfrågor och fritextssvar	11
2.2.2 Digitalt genomförande.....	11
2.2.3 Urval	12
3. Läroplaner och programmeringsspråk	14
3.1 Programmering i gymnasiematematiken i tidigare läroplaner	14
3.2 Senaste läroplan	15
3.3 Motiv till införandet av programmering i matematiken	15
3.4 I England	16
3.5 Programmeringsspråk för matematik	17
4. Resultat	19
4.1 Presentation av data	19
4.2 Programmeringsspråk, programbibliotek och programmeringsnivå	19
4.3 Resultat av graderingsfrågor hela materialet oaktat andra variabler.....	19
4.4 Skillnader i resultatet med avseende på kön	22
4.5 Skillnader i lärarnas erfarenhet av programmering	24
5. Diskussion	27
5.1 Programmeringsspråk... ..	28
5.2 Programmeringen är svår	28
5.3 Programmeringsnivån varierar	29
5.4 Kan öka den matematiska förståelsen.....	29
5.5 Inget ökat intresse för programmering.....	31
5.6 Könsskillnader	31
6. Slutsats	33
7. Förslag på fortsatt forskning	34
8. Referenslista	35
9. Bilagor	38

1. Inledning

Från och med höstterminen 2018 är programmering en obligatorisk del i flera av gymnasieskolans matematikkurser. Problemlösning i matematik skall kunna ske inte bara med hjälp av *"digitala medier och verktyg"* [1, s.105] som tidigare. Nu skall problemlösning innefatta *"[...]Strategier för matematisk problemlösning inklusive modellering av olika situationer, såväl med som utan digitala verktyg och programmering"* [2, s.9]. Citaten ovan kommer från kursen 1c, men liknande formuleringar återfinns även i matematikkurserna 2c, 3a, 3b, 3c, 4 och 5.

Programmering ses som en del av "digital kompetens" som skolverket vill värna om. Det har bl.a. sneglats på länder som Storbritannien som redan 2013 infört programmering i den obligatoriska läroplanen.

Emellertid nämns inte i någon enda punkt VAD som menas med programmering eller HUR det skall göras. Något specifikt programmeringsspråk nämns inte, ej eller hur omfattande programmeringen för problemlösningen skall vara. Skolverket nämner i ett kommentarmaterial till och med

Programmering är medvetet beskrivet i ämnesplanerna på ett sådant sätt att det är möjligt att variera hur mycket och i vilka former det kan förekomma när det gäller problemlösning i undervisningen. När det inte anges vilken typ av verktyg som ska användas är det möjligt för eleverna att välja det verktyg som de tycker är mest lämpligt. [3, s.9]

En sådan typ av oklarhet torde rimligen kunna leda till en divergens i hur programmering implementeras på olika skolor. Förmodligen kommer olika programmeringsspråk användas på olika skolor och i olika klasser av olika lärare och olika elever. Likaså kommer förmodligen uppfattningar att gå isär i fråga om förtjänsten av programmering i matematikundervisningen. Möjligheten att göra adekvata val av eleverna borde också hänga samman med vederbörandes tidigare erfarenhet av programmering, vilket ingalunda kan förutsättas att alla har. Detta samtidigt som den statliga utredning som ligger till grund för införandet av programmering i läroplanen i sina analyser påpekar att det *"finns ett problem inom svensk skola med bristande likvärdighet [...] hur digitala resurser används i undervisningen"* och *"ser ett behov av att staten i styrdokumentet tydliggör digitaliseringsperspektivet"* [4, s. 214].

1.1 Syfte

Syftet med detta arbete är försöka reda ut hur programmering har implementerats i matematiken och hur djupgående det lärs ut för ändamålet. Dessutom hur programmeringen upplevs av elever och lärare i svårighet och i nyttoaspekt för problemlösning i matematiken, alltså om skolverkets intentioner överensstämmer med det upplevda resultatet.

Tidigare studier visat att det rått en stor osäkerhet bland lärare ute på skolorna inför implementeringen och av programmeringen. Det har bl.a. upplevts att informationen om vad som avses med programmering är diffus, att styrdokumentet är oklara gällande vilken form och omfattning programmeringen skall ha. Vidare att många lärare känner sig obekväma inför det nya momentet p.g.a. att de själva ej har erforderlig kunskap för att lära ut programmering [5]. En annan studie har visat på skillnader mellan olika lärare i förhållningssätt till programmering i uppfattningen huruvida programmering är något som alla elever har förmåga att lära sig eller inte. [6]

Förhoppningen är att detta arbete kommer vara till nytta för dem som just nu brottas med införandet av programmeringen i matematiken. T.ex. gymnasielärare i matematik, däribland författaren själv, eller personer som på ett eller annat sätt kommer i kontakt med den didaktiska aspekten av programmering för elever utan tidigare programmeringsvana på icke-högskolenivå. Eventuellt kommer komplikationer gällande likvärdighet att belysas, i den mån sådana kan påvisas, i hur olika programmeringen implementerats på olika ställen. Därvidlag finns även en förhoppning att några ”good practices” utkristalliserar sig - några goda exempel på hur programmering i matematiken kan implementeras och fungera på ett bra sätt.

1.2 Frågeställningar

Följande frågeställningar hoppas kunna bli besvarade:

- (1) Hur djupgående lärs programmering ut? (d.v.s. vilka programmeringstekniska procedurer lärs ut: loopar, villkor, funktioner, klasser o.s.v.)
- (2) Hur upplevs nyttan av programmering för förståelse av matematik ur elev- och lärarsynvinkel?
- (3) Har intresset för programmering som sådant ökat p.g.a. att det är en del i matematikkursen? (d.v.s. har den enskilda eleven fått ett större programmeringsintresse än tidigare p.g.a. att det ingår i matematikkurserna)
- (4) Kan man skönja någon könsrelaterad skillnad i resultaten?

1.3 Avgränsning

I litteraturen som används, i den aktuella debatten och i diskussionen i stort pratas ofta om *digitalisering* i Sverige och i skolan. Denna uppsats riktar enbart in sig på den del av digitaliseringen som avser programmering. Även begreppet programmering kan i sammanhanget åsyfta programmering som eget ämne i skolan, eller programmering som verktyg i diverse skolämnen. Här berörs endast den programmering som omnämns i kursplanerna för matematik i gymnasieskolan.

2. Metod och genomförande

2.1 Litteraturstudium

Sökandet efter lämplig litteratur tog sin början på *Google Scholar*. Sökningar företogs med söksträngen ”programmering matematik gymnasiet”, ”programming school curriculum”, ”programmeringsspråk för matematik” och ”programming languages for mathematics”. Därefter företogs sökningar i BTH:s eget bibliotek via *BTH summon* på ”programmering matematik” och ”programming mathematics”. Skolverkets egen sökmotor har också använts med sökordet ”programmering”.

Då införandet av obligatorisk programmering i matematik på gymnasiet nytt fr.o.m. höstterminen 2018 kan mängden litteratur redan skriven om hur det implementerats ute på skolorna av naturliga skäl inte vara så stor. Vidare är litteratur äldre än tre, fyra år inte så relevant i detta fall, då diskussionen är så pass ny. I den då helt nya och omarbetade läroplanen från 2011 fanns programmering som del av matematikämnet över huvud taget inte med.

När arbeten skrivna före 2016 sorterats bort bland resultaten från Google-Scholar-sökningen ”programmering matematik gymnasiet”, återstod ett antal uppsatser med liknande tema som detta arbete, men som i huvudsak behandlat hur fenomenet upplevs ute på skolorna *inför* implementerandet. De flesta av dem var också examensarbeten. En doktorsavhandling i resultatet har emellertid också använts som referens i detta arbete. På klassiskt manér har referenser i funnen litteratur i lämpliga fall använts för att finna ytterligare litteratur i ämnet.

Att hitta litteratur som behandlar lämpliga *programmeringsspråk* för matematik på grundnivå har visat sig förknippat med vissa komplikationer. Söksträngen ”programmeringsspråk för matematik” gav i stort sett samma träffar som ovan, med uppsatser som behandlar införandet, men där själva programmeringsspråket i sig inte låg i fokus. Den engelska söksträngen ”programming languages for mathematics” gav träffar med hur några olika programmeringsspråk var för sig kan användas för matematisk programmering, de flesta emellertid på högre matematiknivå för högskola, eller på för låg nivå, som behandlar hur programmering kan införas i förskoleåldern.

På *BTH summon* gav söksträngarna ”programmering matematik”, ”programming mathematics” förvisso en mängd träffar, men berörde nästan uteslutande matematik och programmering på högskolenivå och den mesta litteraturen var 10 år eller äldre. Söksträngen ”programmering matematik gymnasiet” gav bara sex träffar, varav fem från 90- eller 00-talet, men en var en

videolänk från utbildningsradion från en föreläsning 2018 om att införa programmering i skolmatematiken, som var relevant i sammanhanget.

Via söksträngen ”programmering” i den egna sökmotorn på skolverkets hemsida länkas man så småningom till webbaserade fortbildningskurser i programmering för lärare. I tillhörande material som finns publicerat på nätet förekommer exempelkod i några olika programmeringsspråk, mest Scratch, Python och i några fall JavaScript. I några av dessa publikationer nämns ytterligare en handfull olika programmeringsspråk för matematikändamål och en kort diskussion om för- och nackdelar mellan dessa förekommer.

Någon vetenskaplig avhandling som opartiskt jämfört hur väl olika programmeringsspråk lämpar sig som verktyg för matematik i gymnasiet har inte kommit till författarens kännedom.

För att hitta mer om bakgrunden till det politiska motivet med införandet, samt information om nya och tidigare formuleringar i kursplaner har i viss mån ”vanliga” google-sökningar företagits med söksträngarna ”skolverket programmering gymnasiet”, ”kursplan matematik”, ”regeringen programmering matematik”. En strikt sortering av svaren har emellertid gjorts och endast de beaktats, som skrivits av institutioner som skolverket, regeringskansliet, statens offentliga utredningar (SOU) och liknande. Dessa institutioner har givit ut ett antal betänkanden och utredningar inför utökandet av digital kompetens i skolans kurs- och läroplaner, som använts som bakgrund och för att hitta information om motivet till de reviderade läro- och kursplanerna.

Via sökningen på ”programming school curriculum” på Google-Scholar var det svårt att hitta resultat som var direkt relevanta för detta arbete som gäller det svenska införandet. Däremot refererade flera av de redan funna källorna till en viss brittisk utredning som i stor omfattning legat till grund för motsvarande implementation för skolorna i England, ett land som legat steget före Sverige i detta avseende. Den finns också med bland referenslitteraturen som använts i detta arbete.

2.2 Empiriskt studium

För att få en aktuell bild av hur implementeringen ute på fältet upplevts av berörda elever och lärare skulle ett antal intervju-studier kunna företas. En annan möjlighet vore en enkätstudie. Av skäl som framkommer i nästa stycke föll valet på det senare.

2.2.1 Enkätundersökning med graderingsfrågor och fritextsvar

Då en jämförelse vill kunna göras om huruvida skillnader kan föreligga i hur programmeringen upplevs från elevers sida beroende på lärarens programmeringserfarenhet, upplägg eller kanske på val av programmeringsspråk/programbibliotek, vore en möjlighet att studera några olika klasser med lite olika upplägg. Genom att skapa ett antal graderingsfrågor som eleverna skall ta ställning till, skulle man därefter kunna jämföra klasserna med varandra för att se om några signifikanta skillnader i svarsfördelningarna föreligger, som kan bero på upplägg eller andra bakomliggande faktorer.

Därför föll valet på att företa fallstudium där några olika klasser undersöks via enkäter, och dessutom ombeds varje lärare i respektive klass besvara liknande frågor samt några kompletterande frågor som man inte kan vänta att eleverna känner till. Därmed finns också en möjlighet att eventuellt skönja någon diskrepans mellan hur ämnet upplevs av lärare respektive elever.

Det finns ett antal verktyg ute på marknaden för webbaserade enkätundersökningar av olika slag. Även gratis dito. Kraven i detta fall var emellertid relativt höga: Enkäten skall kunna användas i godtyckligt många klasser vid olika tillfällen. De skall vara anonyma, men ändå kunna sorteras klassvis. Lärarna skall ha andra frågor än eleverna, men ändå kunna knytas till en viss klass. Vidare är det önskvärt att kunna garantera att insamlad data inte sprids vidare till okända tredjepartsanvändare. Därför föll valet på att programmera ett särskilt enkätverktyg enbart för ändamålet.

2.2.2 Digitalt genomförande

Enkätundersökningarna har gjorts klassvis i digital form. Deras exakta utformning återfinns i bilagan till detta arbete. Enkäterna är anonyma men konstruerade så att läraren och elever från samma klass kan grupperas ihop. Både läraren och eleverna svarar. Frågorna är emellertid inte helt likalydande beroende på om informanten är lärare eller elev. Frågorna i enkäten är både av ett sådant slag, där man på en graderad skala tar ställning till hur väl ett påstående stämmer eller ej och av typen fritext.

Både elever och lärare får svara på vilket programmeringsspråk de använder och lärarna om de använder specifika programbibliotek (elevenkätfråga 1, lärarenkätfråga 2 och 3). Det är en del av den information som bör has i åtanke vid beaktandet av elevernas andra enkätfråga.

Den är en fyrgradig graderingsfråga huruvida programmeringen upplevs lätt eller svår. Genom att titta på svaren på den frågan i kombination med andra enkätfrågor kanske en svårighetsgradskillnad beroende på val av språk eller andra faktorer kan skönjas.

Genom att låta lärarna svara på vilka programmeringstekniska procedurer som lärs (fråga 4) ut kan förhoppningsvis uppsatsens första frågeställningen besvaras, som avser hur omfattande programmering lärs ut i matematiken. Eftersom läroplanen intet säger om detta är svaren av intresse.

Den andra uppställda frågeställningen som behandlar den upplevda nyttan av programmering i avseende som hjälpmedel för matematiken. Detta besvaras i en graderad fråga i både lärar- och elevenkäten (lärarenkätfråga 5, elevenkätfråga 4). Eventuellt kan skillnader beroende på val av programmeringsspråk eller andra faktorer skönjas även i denna fråga om den kombineras med svaren i andra frågor med t.ex. språkval och lärarens programmeringserfarenhet.

Den tredje frågeställningen handlar om huruvida det allmänna intresset för programmering ökat tack vare dess förekomst i matematiken. Elevernas fjärde enkätfråga är en fyrgradig direktfråga om detta. Genom att samtidigt jämföra med respektive elevs tidigare erfarenhet av programmering i enkätfråga 5 kan det vara möjligt att finna data som talar för att intresset för programmering i och med detta ökat. Särskilt intressant vore det om det gäller tjejer, vilket kan fastställas med jämförelse med sista enkätfrågan som avser informantens kön. (J.fr. med statistiska centralbyrån 2017, enligt vilken endast 20% av dem mellan 16 och 64 år som arbetade med mjukvaru- och systemutveckling var kvinnor. [7]). Den fjärde frågeställningen avser utröna om någon könsskillnad föreligger.

Så gott som samtliga frågor innehåller en textruta för frivilliga kommentar. Sista enkätfrågan är bara en textruta för övriga kommentarer. Även dessa kan användas för att utröna relevanta resultat utifrån frågeställningarna, som kan bekräfta eller vederlägga dessa. Därutöver kan dessa kommentarer möjligen visa intressanta aspekter som författaren över huvud taget inte haft i åtanke, innan arbetet påbörjades.

2.2.3 Urval

Genom att skicka förfrågan om enkätbaserat deltagande via mejl till ett antal matematiklärare på några utvalda gymnasieskolor i sydvästra Skåne har författaren sökt finna olika klasser att undersöka.

Författaren är medveten om att det finns en risk med denna typ av icke-slumpvisa urvalsmetod om man vill ha en riksomfattande bild. Dels kan en slagsida för någotdera upplägg eller uppfattning av implementeringen förekomma, då informanterna bl.a. p.g.a. geografisk närhet i något avseende är relaterade till varandra. Det kan inte uteslutas att sinsemellan likartade varianter på samma skola eller skolor i samma region utvecklats. Dels är det totala underlaget litet.

Allt detta har emellertid vägts i förhållande till den tid och resurser som funnits till förfogande inom arbetets ram och valet av metod får ses som en kompromisslösning som duger för en fallstudie i ämnet där några klasser blir undersökta.

3. Läroplaner och programmeringsspråk

3.1 Programmering i gymnasiematematiken i tidigare läroplaner

Fenomenet med programmering i matematiken i den svenska gymnasieskolan är i själva verket inte helt nytt. I Skolöverstyrelsen nämner 1981 i ett supplement till LGY 70 (läroplanen för gymnasieskolan 1970) begreppet *datalära* som ett avsnitt i matematikämnet. Det syftar bl.a. till att ge elever kunskap om en dators olika tekniska hårdvarukomponenter, men även begreppet *programmering* förekommer och förslag ges till programmeringsspråk. ”*Som programmeringsspråk används ett enkelt högnivåspråk typ BASIC*” [8, s.9]

En viss tillbakagång i programmeringsavseende tycks emellertid kunna skönjas i nästa gymnasiereform, *lgy94*, som började gälla 1994. I *lgy94* nämns att skolan i sin matematikundervisning skall stäva efter att eleverna

Utvecklar sina kunskaper om hur matematiken används inom informationsteknik, samt hur informationsteknik kan användas vid problemlösning för att åskådliggöra matematiska samband och för att undersöka matematiska modeller. [9, s.2]

I ett senare avsnitt nämns visserligen att tillgången på tekniska hjälpmedel delvis har förändrat matematikämnet, men påtalar att ”*[d]e tekniska hjälpmedlen har dock begränsat värde utan kunskaper om begrepp och metoder*” [9, s.2]. Man tycks alltså vilja betona vikten av att inte enbart förlita sig på tekniska hjälpmedel, snarare än att uppmuntra ett ökat användande av dem.

Begrepp som *datorer*, *grafiska räknare*, *grafisk*, *numerisk* eller *symbolhanterande programvara* förekommer i denna läroplan, men endast i vaga formuleringar. Dessutom nämns de endast som mål som eleverna skall ha uppnått efter avslutad kurs och inte alls i några betygskriterier. Ordet *programmering* nämns endast i kursen ”Diskret matematik” där eleven efter avslutad kurs förväntas ”*känna till hur grundläggande satslogik används i programmering och problemlösning*” [9, s.6]. Detta var en valbar kurs på natur- och teknikprogrammen. I de obligatoriska matematikkurserna nämns inte programmering alls¹.

¹ Programmering förekom de facto ändå i undervisningen under denna gymnasieform i flera matematikkurser, om än i begränsad omfattning. Som implementation av användning av ”numerisk programvara” användes bl.a. programmering i de BASIC-liknande språken som fanns inbyggda på de miniräknare de flesta eleverna använde vid denna tid. Författaren var själv en bland de många matematiklärare som lärde ut hur miniräknarna kunde programmeras för att underlätta olika former av numeriska beräkningar vid t.ex. rekursiva formler.

3.2 Senaste läroplan

2011 börjar ännu en ny läroplan för gymnasieskolan, kallad *gy11*, gälla. Både i centrala innehållet och i betygskriterierna för samtliga matematikkurser nämns *digitala verktyg* som hjälpmedel vid problemlösning. [1]. *Programmering* nämns dock inte alls i den ursprungliga versionen.

Först i den reviderade upplagan 2017, som är obligatorisk fr.o.m. höstterminen 2018 förekommer ordet *programmering*, som verktyg att använda i samband med problemlösning i matematiken [2]. Det gäller matematikkurserna 1c, 2c, 3c, 4 och 5, d.v.s. samtliga matematikkurser för natur- och teknikprogrammet, tillika kursen matematik 3b som förekommer på bl.a. ekonomi-, humanist- och samhällsprogrammet och är i de flesta fall valbar men icke-obligatorisk. De elever som faktiskt påverkas av det nya obligatoriska införandet av programmering i gymnasiematematiken är alltså endast de elever som har valt teoretiska och matematikorienterade gymnasieprogram eller inriktningar.

Som tidigare nämnts i inledningen föreslås inget specifikt programmeringsspråk, ej heller i vilken omfattning eller hur denna programmering skall gå till.

Programmering är medvetet beskrivet i ämnesplanerna på ett sådant sätt att det är möjligt att variera hur mycket och i vilka former det kan förekomma när det gäller problemlösning i undervisningen. När det inte anges vilken typ av verktyg som ska användas är det möjligt för eleverna att välja det verktyg som de tycker är mest lämpligt [3, s.19].

3.3 Motiv till införandet av programmering i matematiken

I oktober 2011 presenterade dåvarande regeringen Reinfeldt sin it-politiska agenda *It i människans tjänst – en digital agenda för Sverige* med bl.a. de ambitiösa målen att ”Sverige skall bli bäst i världen på att använda digitaliseringens möjligheter” [10 s.15]. 2012 tillsätts en kommitté, *Digitaliseringskommissionen*, för arbetet med att genomföra den digitala agendan. [4]

Några år senare, 2015, i samband med strategin att främja digital kompetens, skrev dåvarande utbildningsminister Gustaf Fridolin m.fl. 2015 i en debattartikel att man vid behov kan föreslå förändringar i kurs- och ämnesplanen för att ”stärka den digitala kompetensen hos eleverna så att den ska motsvara de krav som ställs på arbetsmarknaden och för vidare utbildning” [11].

Om programmering sägs i samma artikel

Programmering handlar inte enbart om att skriva kod. Problemformulering och -lösning, kreativitet och logiskt tänkande är exempel på centrala förmågor. Förmågor som även kan bidra positivt till kunskapsinhämtningen inom andra ämnen. [11]

I citatet ovan omnämndes programmering visserligen i samband med ändringar i läroplanen för grundskolan, inte gymnasiet, men det framgår ändå att man hoppas på att kunskap i programmering skall få positiva effekter även i andra ämnen.

I en liknande debattartikel 2016 skriver Fridolin tillsammans med politikern Mikael Damberg

Genom att eleverna får bekanta sig med programmering som en naturlig del av undervisningen skapas också förutsättningar för en jämnare könsfördelning i utbildnings- och yrkesval längre fram [12].

Detta överensstämmer med vad digitaliseringskommissionen skrev 2014 i sitt delbetänkande

Ökade inslag av it i undervisningen bedömer vi till exempel kan främja flickors intresse för it och på sikt bidra till att minska den könsmässiga snedrekryteringen till högre utbildningar med it-inriktning. [4, s.129]

Förutom att rusta eleverna för framtidens krav på digital kompetens på arbetsmarknaden hoppas man alltså på positiva effekter i andra skolämnen och en jämnare könsfördelning i utbildnings- och yrkesval.

3.4 I England

England är ett land som har legat steget före Sverige i skolprogrammeringshänseende och kan därför vara intressant att titta på i sammanhanget. Där finns programmering med som obligatoriskt ämne i läroplanen för alla åldrar redan 2013 [13]. Programmering ingår där i det egna skolämnet ”Computing”.

Året innan skrevs av the Royal Society den omfattande rapporten *Shut down och restart – the way forward for computing in UK schools*, som argumenterar för ökat inslag av Computing i det brittiska skolsystemet. I denna drar man paralleller mellan ”Computer Science” och fysik. Man menar att på samma sätt som det finns ett värde i att lära ut fysik till alla, fastän ytterst få blir fysiker, är det värdefullt att lära ut Computer Science till alla. Man använder bland annat argument som att det är en fundamental del av vår värld, det kan utveckla ett sätt att tänka i

logiska resonemang, vid abstraktioner och problemlösning största allmänhet samt även gagna skapande och kreativa lösningar. Man har även sett den dittillsvarande sneda könsfördelningen i högre studier i datavetenskap som problematisk. [14]

3.5 Programmeringsspråk för matematik

Som tidigare nämnts angavs BASIC som lämpligt programmeringsspråk för att lösa matematiska problem på tidigt 80-tal. Sedan dess har inte bara datorerna utan även programmeringsspråken utvecklats och blivit flera.

Några opartiska studier till förmån för det ena eller andra programmeringsspråket att använda i matematik har inte kommit till författarens kännedom.

I den senaste kursplanen lät skolverket val av språk vara flytande [3], men genom att beakta av skolverket utgivet fortbildningsmaterial kan man få sig en fingervisning. Under larportalen.skolverket.se återfinns publiceringar avsedda för vidareutbildning av personer verksamma i skolan. Kommande stycken ger några exempel.

I häftet *Aktivitetsbank* [15] ges ett antal konkreta förslag på var och hur programmering kan användas för problemlösning i matematikundervisningen på gymnasiet. Exempelkoderna är genomgående skrivna i Python och man tar även hjälp av några programbibliotek, t.ex. random, matplotlib och Numpy.

Ännu fler exempel på matematiska problem som kan lösas med programmering återfinns i *Programmering i gymnasieskola och vuxenutbildning –uppgifter till workshop* [16] där även förslag på lämpliga programmeringsspråk nämns vid varje problem. I de flesta fall rekommenderas emellertid ”alla språk”. Ibland varnas för kalkylblad. I ett enstaka fall nämns specifikt Octave/MATLAB eller Python3 (s17), men samtliga demonstrerade kodexempel är i Python. Dock finns ett separat häfte med lösningsförslagen skrivna i MATLAB tillgängligt [17].

I referensmaterialet till ovanstående nämns ytterligare ett antal programmeringsspråk och en mindre diskussion om för och nackdelar med olika språk förekommer. Fördelar med MATLAB anses vara att det är väletablerat för matematiska beräkningar och simuleringar och även används på industriell nivå och anses liksom Python vara ganska lätt att komma igång med. JavaScript och PHP omnämns därför att de är vanliga inom webbprogrammering och elever

och lärare kan ha stött på dem i sådana sammanhang. Vidare kan JavaScript användas i GeoGebra som är ett vanligt digitalt verktyg i matematikundervisning. [18]

En lathund för grundläggande syntax tas upp för MATLAB, Python3, JavaScript och PHP. Vidare omnämns språken Java C/C++/C#, men man menar att referensblad för dessa vore överflödiga trots att språken är vanliga. Detta därför att språken anses svåra att komma igång med initialt och i de fall de används i matematiken, torde det vara av dem som redan är väl förtrogna med språken och därmed inte behöver en lathund för grundläggande syntax. [17]

I *Verktyg – lärarhandledning* ges diverse länkar till online-miljöer för att skriva och exekvera program t.ex. repl.it. Där nämns förutom Python och Javascript också Wolfram och Scratch [19]. Det sistnämnda är ett webbaserat visuellt programspråk med s.k. blockprogrammering som återfinns på många ställen i skolverkets material, men då oftast i sammanhang då programmering introduceras i de yngre åldrarna.

Daniel Barker, namnkunnig i pedagogiska kretsar bl.a. för undervisningskonceptet ”flipped classroom” [20], föreläser i utbildningsradions informationsvideo om programmering i matematik och använder där Python, som han ansåg fungera utmärkt till matematik och ha en låg tröskel. Programmen skrevs i online-miljön repl.it och föreläsaren använde initialt inga programbibliotek utan ansåg att det var viktigt att börja med mycket enkla programmeringsexempel som låg nära matematiken om man ville ha med sig alla elever på tåget. [21]

4. Resultat

Sammantaget inkom resultat från 38 elever, 8 flickor och 30 pojkar fördelade på två klasser från olika skolor. Båda klasserna läste kursen matematik 1c, den första mattekursen på natur- och teknikprogrammen. Lärarna till klasserna hade sig emellan olika erfarenhet av programmering.

4.1 Presentation av data

Enkäterna innehöll såväl frågor med fritextsvar som frågor där det handlade om att på en fyrgradig skala ta ställning till hur väl ett påstående överensstämde med informantens egen uppfattning. Resultat från den senare typen presenteras i form av stapeldiagram med en procentuell skala på y-axeln, för att ge en tydlig överblick. Staplarna på x-axeln har värdena 1, 2, 3, 4 där 1 är det mest "negativa" och 4 det mest "positiva". För att underlätta läsningen i diagrammen är frågetexten där i vissa fall förkortad och inte exakt likalydande med formuleringen i enkäten. (Se bilaga för originalformuleringar).

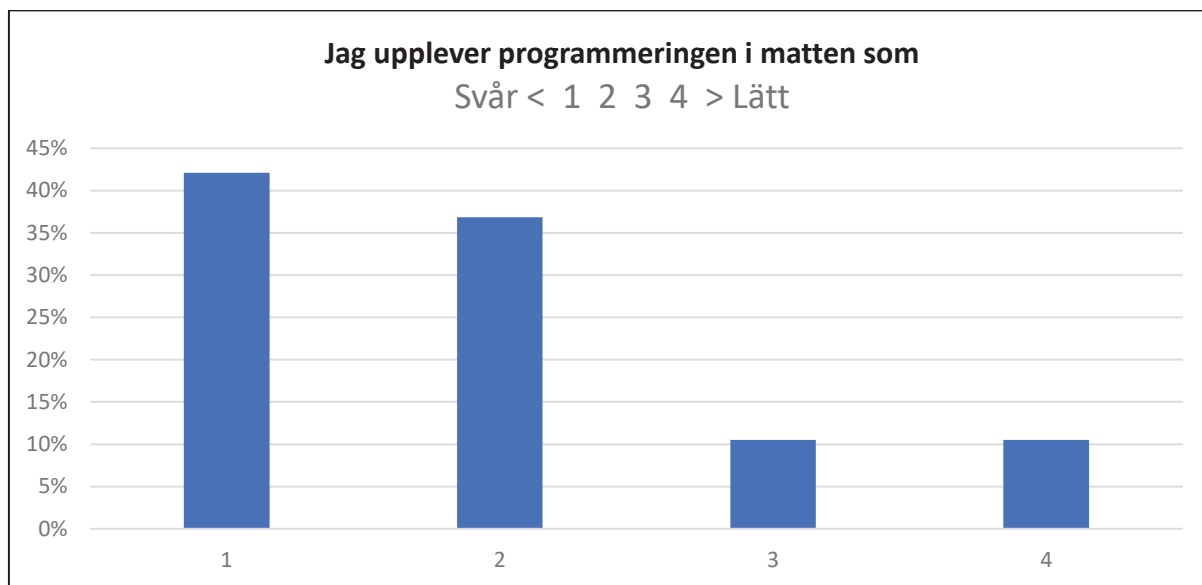
4.2 Programmeringsspråk, programbibliotek och programmeringsnivå

De undersökta klasserna använde programspråken Python. En klass använde inget särskilt programbibliotek, medan den andra använde Matplotlib och math. Den ena läraren angav att hen inte lärde ut några speciella programmeringsprocedurer alls, medan den andra använde loopar, funktioner och villkorstester.

4.3 Resultat av graderingsfrågor hela materialet oaktat andra variabler

Figur 1 till och med figur 3 visar det sammantagna resultatet av elevenkäterna i de frågor som handlade om att gradera ett påstående. Ingen viktning eller hänsyn har här tagits till det faktum att den totala könsfördelningen var ojämn eller att den ena klassen var större än den andra.

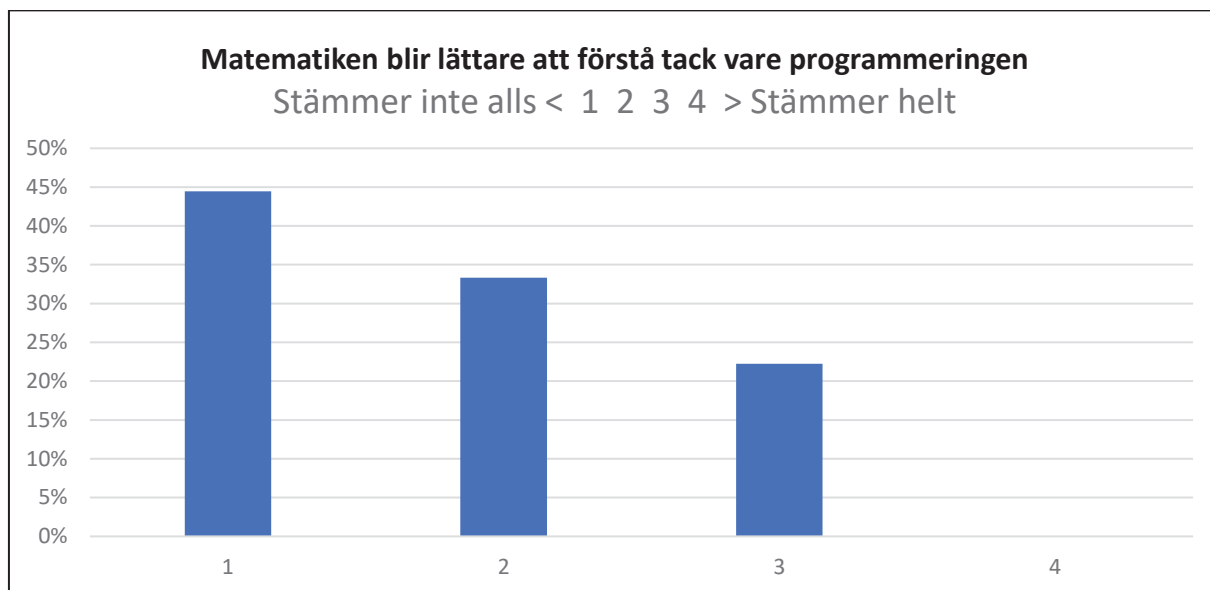
Här framkommer om vi tittar på helheten en övervägande negativ bild. Programmeringen upplevs av flertalet som svår, underlättar inte förståelsen av matematik och få har fått ökat intresse för programmering som sådant.



Figur 1. Upplevd svårighetsgrad av programmering. Hela underlaget.

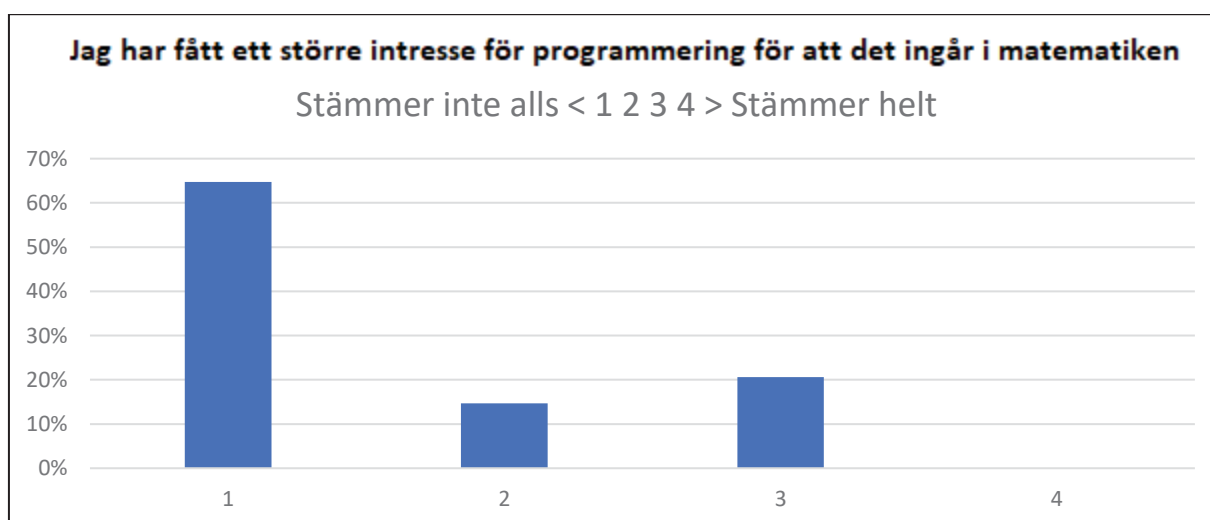
I frisvarsfältet till frågan om upplevd svårighetsgrad (Figur 1) framkom kommentarer som ”känns väldigt klyddigt”, ”svåra matteproblem ska lösas med program som man inte förstår”, ”Tycker det kan vara svårt eftersom vi inte har en så stor bakgrund”, ”Det enda vi lär oss göra är att skriva av en annan kod, nästan ingen förstår vad det är vi faktiskt gör”. Undantag finns emellertid också. En informant som graderade 2, på svårighetsfrågan, valde ändå 3 på frågan om ökat intresse och skrev i kommentaren ” Jag tycker det är bra att man lär sig grundläggande programmering, eftersom det förbättrar ens logiska tänkande.”

Av dem som angav alternativ 4, att programmering upplevdes lätt, angav 75% att de sysslade med programmering tidigare på fritiden.



Figur 2. Programmering underlättar matematiken. Hela underlaget.

Liknande staplar och kommentarer framkom gällande huruvida programmeringen underlättar förståelsen för matematiken (Figur 2): ”Hur skulle det hjälpa?”, ”förstår inte kopplingen”, ”tycker den är lätt som den är”. Men även ett försiktigt positivt svar framkom ”Om man bara kan och har kunskaperna kring programmering så tror jag att det kan bli ett bra hjälpmedel. Men just nu är det bara förvirrande.”



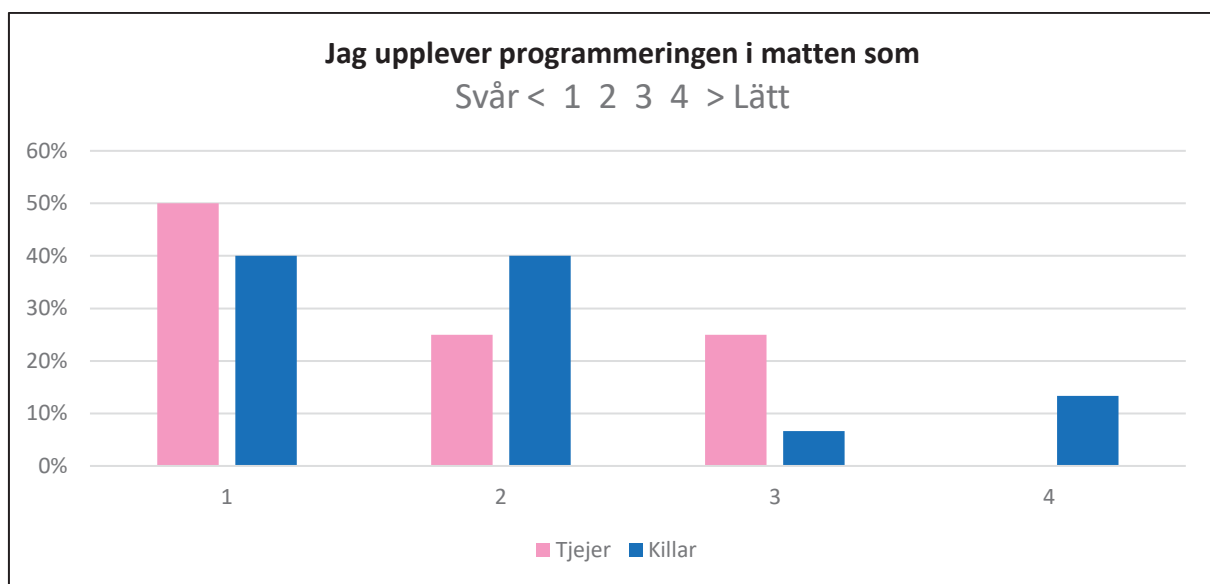
Figur 3. Påverkat intresse för programmering. Hela underlaget.

En snabb blick på figur 3 ger intrycket att intresset för programmering inte har ökat tack vare införandet därav i matematiken. Ett svar i stapeln längst till vänster innebär emellertid inte med nödvändighet att informanten är alltså ointresserad av programmering, utan kanske var

intresserad redan tidigare. I fritextsvaren hos en informant som graderade 1 finns en kommentar som antyder detta "Jag gillade programmering sedan innan". Medan kommentarerna "Jag blir nästan avskräckt för det känns svårt", "tvärtom, jag har fått ett mindre intresse för det" blir svåra att tolka annorledes än att matematikprogrammeringen inte bidragit positivt till intressegraden.

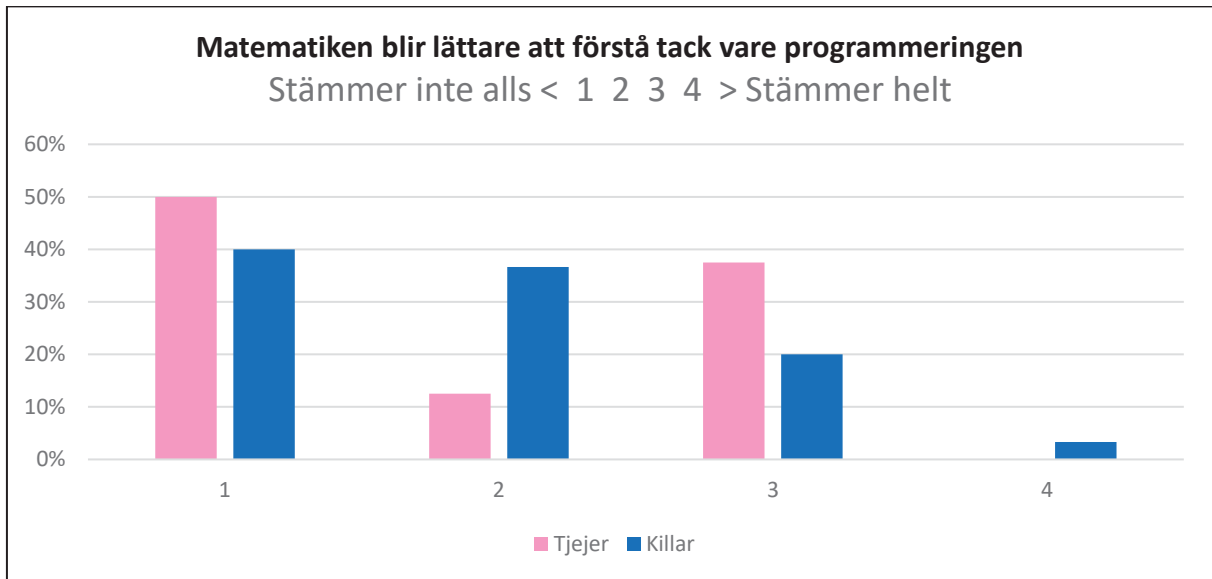
Den femte frågan gällde tidigare erfarenhet av programmering och var en fritextfråga. En granskning av svaren visar att de allra flesta sysslade med programmering på ett eller annat sätt tidigare, men i mycket liten omfattning. Flera av svaren var ganska stereotypa och nämnde blockprogrammering på tekniken i skolan. Någon enstaka hade programmerat Raspberry Pi och ytterligare några angav att de programmerat på fritiden, men inte i vilken utsträckning.

4.4 Skillnader i resultatet med avseende på kön



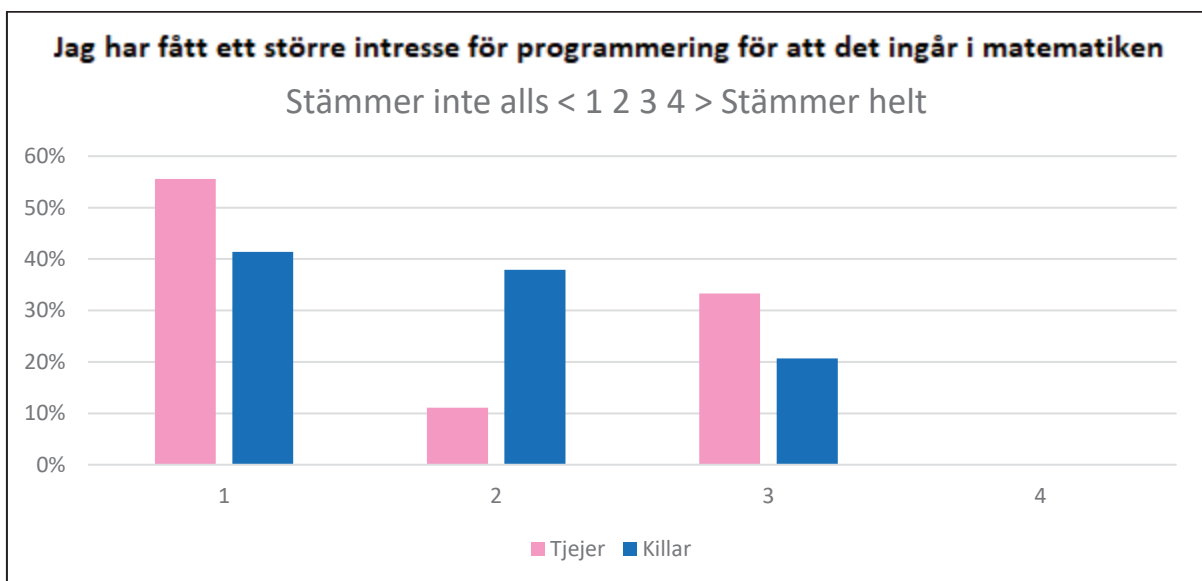
Figur 4. Upplevd svårighetsgrad fördelat på kön

Som figur 4 visar, valde en något högre andel tjejer än killar alternativet 1, det starkaste alternativet att programmeringen upplevdes som svår och ingen tjej valde alternativ 4, det starkaste alternativet att programmeringen upplevdes som lätt. Slår man å andra sidan ihop alternativ 1 och 2, att programmering upplevs som mer eller mindre svår, samt alternativ 3 och 4, att programmering uppfattas som mer eller mindre lätt, anser en större andel av killarna (80%) än tjejerna (75%) att programmering upplevs som svår.



Figur 5. Matematiken upplevs lättare tack vare programmering fördelat på kön

En liknande svarsfördelning fanns gällande huruvida programmeringen upplevdes bli lättare tack vare programmering. En större andel tjejer än killar hade valt alternativet att det minst stämde, men om man slår ihop de två alternativen med övervikt på ”stämmer inte” respektive de två alternativen med övervikt på ”stämmer” ansåg en större andel av tjejerna än killarna att det stämmer att matematiken blir lättare att förstå med hjälp av programmering.



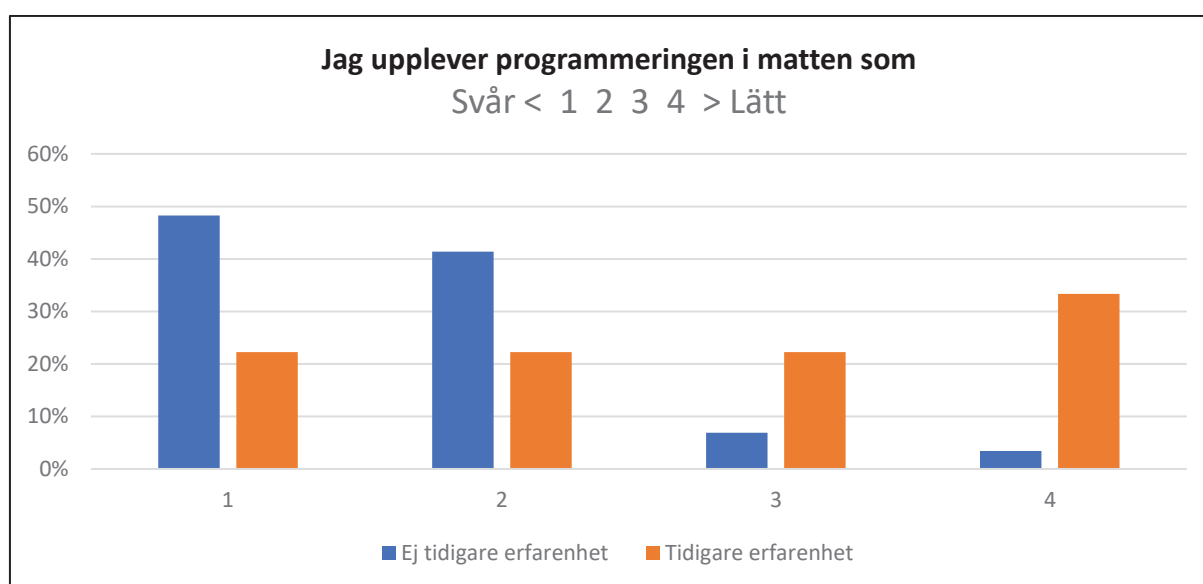
Figur 6. Ökat intresse för programmering fördelat på kön

Könsfördelningen gällande huruvida matematikprogrammeringen bidragit till ett ökat programmeringsintresse generellt (figur 6) var mycket snarlik fördelningen i den förra frågan (figur 5).

4.5 Skillnader i lärarnas erfarenhet av programmering

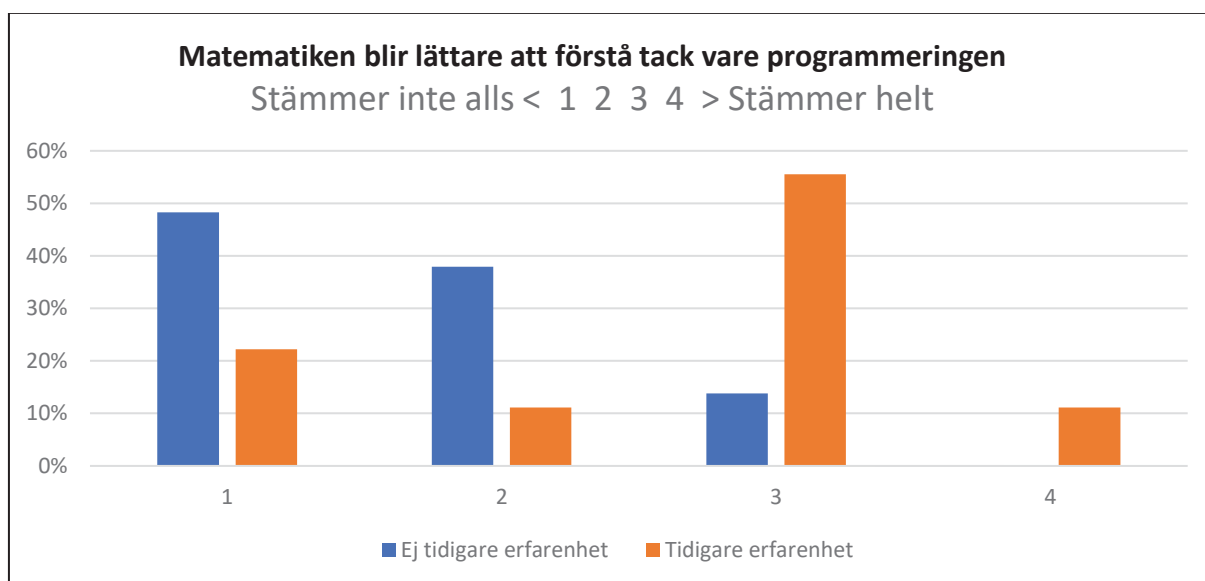
En av lärarna angav att hen inte hade någon tidigare erfarenhet av programmering, medan den andra hade erfarenhet av programmering sedan tidigare. Den sistnämnda angav också att hen använde några speciella programbibliotek för matematik och lärde ut företeelser som funktioner, loopar och villkorssatser.

Flera fritextsvar antydde att programmering endast gjorts i mycket liten omfattning i matematiken. Alla dessa kom från klassen, vars lärare hade mindre programmeringserfarenhet. Även läraren själv antydde att omfattningen av den programmering de tillämpat kanske inte låg i fas med skolverkets intentioner.



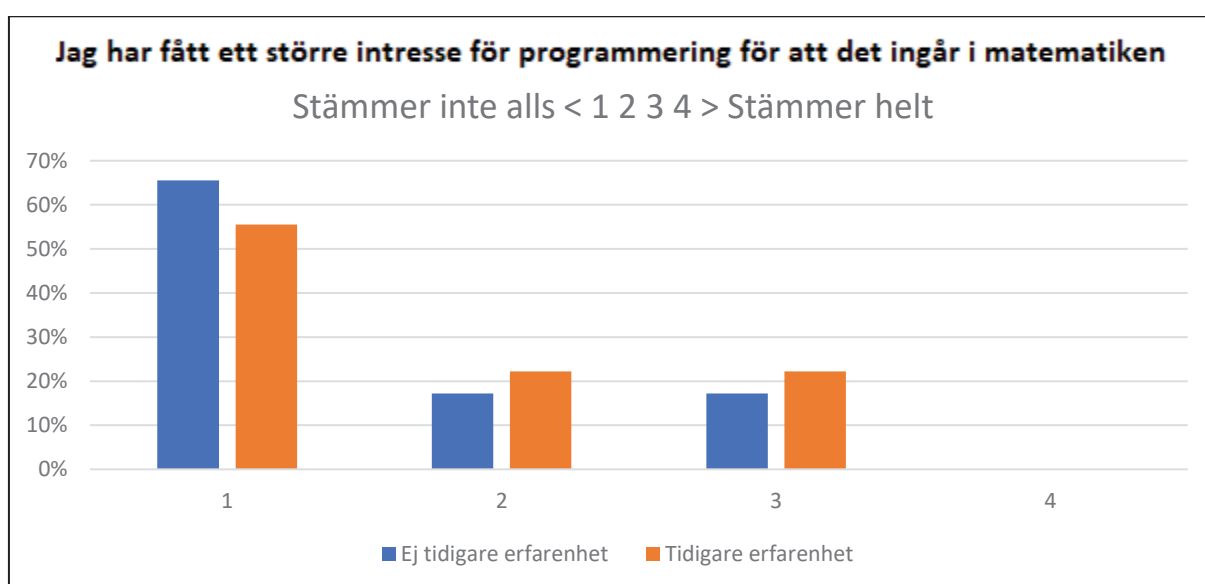
Figur 7. Upplevd svårighetsgrad relaterat till lärarens programmeringserfarenhet

Figur 7 antyder att de flesta elever hos läraren utan tidigare erfarenhet tycker att programmering är svår. Av eleverna hos läraren som hade tidigare erfarenhet är fördelningen väldigt jämt utspritt avseende hur svår programmeringen uppfattas.



Figur 8. Matematiken upplevs lättare p.g.a. programmering relaterat till lärarens programmeringserfarenhet

Figur 8 gör gällande att den största andelen tycker att matematiken blir lättare att förstå tack vare programmeringen hos läraren som hade tidigare programmeringserfarenhet, men ej hos läraren som saknade tidigare programmeringserfarenhet. Det överensstämmer med lärarens uppfattning i frågan i lärarenkäten huruvida programmeringen bidrar till ökad förståelse för matematik där läraren med tidigare programmeringserfarenhet angav att "[d]en ökar om man använder programmering som komplement."



Figur 9. Ökat intresse för programmering relaterat till lärarens programmeringserfarenhet

Någon större skillnad i huruvida programmeringen i matematik bidragit till ett ökat intresse för programmering generellt går inte att skönja med avseende på lärarens tidigare programmeringserfarenhet. Figur 9 visar tvärt om en väldigt likartad fördelning i de båda fallen.

5. Diskussion

Författaren är medveten om att stora generella slutsatser inte kan dras på grundval av det lilla underlaget. Likaså finns ingen möjlighet att se eventuella skillnader i resultat baserat på om eleverna läser olika kurser eller använder olika programmeringsspråk eftersom samtliga eleverna i undersökningen läste matematik 1c och använde Python.

Författaren hyste dock en förhoppning på en högre svarsfrekvens. Två klasser samt deras respektive lärare räcker visserligen för fallstudier, men det hade också varit intressant att undersöka några fler. Orsaker till den låga svarsfrekvensen kan bl.a. vara dålig tajming – denna tid på året brukar i matteklasser präglas av stress inför nära anstående nationella prov och att lektionstiden inte räcker till. Enkäter är därför kanske inte det första som prioriteras. Dessutom var de i sitt utförande lite mer komplicerade för läraren än att bara trycka på en länk, eftersom en kod behövde hämtas och också ges till eleverna. En annan orsak kan vara hur enkätundersökningen ”marknadsfördes” i informationen som skickades till matematiklärare i massmejlform. Det är svårt att sticka ut i bruset av alla de marknadsundersökningar och annat som skickas på liknande sätt. Även fler påminnelser kunde skickats ut än vad som gjordes.

Slutligen kan man naturligtvis också spekulera i, om det faktum, att programmeringsinförandet ute på skolorna som av många upplevts som oklart och otydligt, enligt tidigare studier, haft betydelse. Den som är tvungen att genomföra något nytt, men ej fått tillräcklig information om hur det skall göras, är kanske därför osäker på om det gått hem på rätt sätt, eller i sämsta fall fullständigt struntar i att genomföra det. Oavsett vilket är vederbörande i så fall förmodligen inte överväldigat intresserad att låta någon utomstående granska resultatet, även om det görs anonymt.

Att vara ute tidigare, marknadsföra sin studie bättre, skicka till fler och ”ligga på” mera får vara lärdomar att bära med sig till nästa gång.

Inte desto mindre går det att konstatera att det programmerade enkätverktyget fungerade och frågorna var användbara så att det ur det undersökta materialet gick att dra ett antal intressanta slutsatser och få svar på frågeställningarna såsom kommande stycken presenterar.

5.1 Programmeringsspråk

Skolverket ger, som tidigare nämnts, inga direkta direktiv om vilka programmeringsspråk som bör användas. I litteraturgenomgången kan man ändå skönja en viss övervikt åt Python i fortbildningslitteraturen för ändamålet och i skolverkets exempelmaterial. Klasserna i denna studie använde sig av Python och det verkar vara ett vanligt val av språk, åtminstone lokalt. Författaren tog också kontakt med en anställd vid *Vattenhallen Science Center*², en form av institution vid Lunds universitet som bland mycket annat fortbildat lärare i regionen i programmering för matematik de senaste åren. Där kände man vid tillfället inte till någon klass eller skola som använt sig av något annat programmeringsspråk än Python för ändamålet. För författarens räkning skickades även en förfrågan ut via en mejllista till lärare, som varit involverade i deras fortbildning. Samtliga inkomna svar därifrån uppgav också Python. Även om detta inte ger en rikstäckande bild eller ger vetenskapligt stöd för Pythons förträfflighet, verkar Python i alla fall inte vara olämpligt som val av programmeringsspråk i matematiken.

Gällande programbibliotek använde en av lärarna i undersökningen sig av math och Matplotlib. I exempelaterialet från skolverket används bl.a. även random och NumPy [15]. Vilka programbibliotek som behöver används i ett givet sammanhang, hänger naturligtvis samman med vilka typer av matematikproblem som ska lösas.

5.2 Programmeringen är svår

I elevenkäten uppmanades informanterna i en av frågorna ta ställning på en fyrgradig skala om de upplevde programmeringen i matematiken som lätt eller svår. Detta var ingen fråga i som detta arbete direkt avsåg att besvara, men den fick vara med i enkäten ändå som bakgrundsinformation för övriga frågeställningar. Det är ju t.ex. inte otänkbart att en viss elev uppfattar programmeringens förtjänst i att lösa ett matematiskt problem som mindre än en annan elev enbart p.g.a. programmeringen i sig upplevs för svår av ena eleven men inte av den andra. Det har också visat sig utifrån materialet att frågan om upplevd svårighetsgrad faktiskt hänger samman med andra faktorer som avser mätas i denna undersökning.

Sett utifrån hela materialet tycks majoriteten av eleverna uppleva att programmeringen i matematiken som svår. Det tycks emellertid föreligga en skillnad beroende på om läraren haft

² Se <http://www.vattenhallen.lth.se/om-oss/>

erfarenhet av programmering sedan tidigare. I gruppen där läraren saknade tidigare erfarenhet av programmering angav en tydlig majoritet att de upplevde programmeringen i matematiken svår. I gruppen där läraren däremot hade tidigare erfarenhet, uppgav en liten majoritet att de inte tyckte att programmeringen var svår, men fördelningen var i den gruppen mycket jämn mellan alla graderingsalternativen.

5.3 Programmeringsnivån varierar

Den första frågeställningen (1) gällde hur djupgående programmeringsspråket lärs ut. Inte heller om detta ges några tydliga direktiv från skolverket. Det undersökta materialet gav varierade svar i denna fråga:

Läraren som saknade tidigare programmeringserfarenhet lärde inte ut något som loopar, villkor eller funktioner alls. Då kan inga ”förgreningar” i programflödet åstadkommas. Programmeringen skulle kanske i detta fall kunna liknas med att förfarandet att skriva in formler på en avancerad miniräknare som klarar prioriteringsregler, möjligen med tillägget att man enkelt kan ändra värdena på olika variabler för upprepade beräkningar.

Läraren med tidigare programmeringserfarenhet hade däremot lärt ut loopar, funktioner och villkorstester och använde sig av programbibliotek, vilket möjliggör betydligt komplexare funktionalitet i programmen.

5.4 Kan öka den matematiska förståelsen

Förhoppningen att programmering kommer att öka elevers logiska tänkande i allmänhet stod, som nämnts tidigare, i klartext i den engelska utredningen *Shut down och restart – the way forward for computing in UK schools* [12] (se avnitt 3.4). I svenska läroplanen nämns programmering som ett verktyg för matematisk problemlösning [2]. Då är alltså frågan om den införda programmeringen i verkligheten ger de önskade resultaten.

Den andra frågeställningen (2) gick ut på att undersöka just huruvida programmeringen upplevs göra nytta för förståelsen av matematiken, både från lärarens och elevernas perspektiv. Huruvida programmeringen gagnade matematiken ger det undersökta materialet en kluven bild av, men här blir det intressant om vi jämför med resultatet med vad lärarna uppgav om tidigare programmeringserfarenheten och nivån som programmeringen lades på. Här syntes nämligen

den största skillnaden i det undersökta materialet beroende på lärarens tidigare programmeringserfarenhet och hur djupgående programmeringen lärdes ut.

I gruppen vars lärare saknade tidigare programmeringserfarenhet ansåg en stor majoritet att påståendet *”Matematiken blir lättare att förstå tack vare programmeringen”* inte stämde. Även flera fritextkommentarer från informanter i den gruppen illustrerade detta (se föregående kapitel). Här kommer ytterligare en: *”Tycker det tar mest upp tid som kan användas till matten.”*. Det hör dock till saken att flera av eleverna i den gruppen ansåg att de sysslade för *lite* med programmering (för att förstå själva programmeringen). Någon kommenterade påståendet i frågan *”Kanske men inte på vår nivå eftersom vi har jobbat så lite med programmering och så mycket med matte så är avståndet i nivå alldeles för stort i mitt tycke.”*. Några större djupdykningar hade ju inte heller gjorts i utlärandet av programmering och inga programbibliotek hade använts.

I gruppen vars lärare hade tidigare erfarenhet angav emellertid de flesta grad 3 att påståendet *”Matematiken blir lättare att förstå tack vare programmeringen”* stämde. Den gruppen hade lärt sig fler programmeringsorienterade tekniker och använt programbibliotek

Så svaret på den andra frågeställningen (2) är alltså att en majoritet av eleverna *kan* uppleva att programmeringen gör nytta i förståelsen för att lösa matematiska problem, men i så fall får inte programmeringskunskaperna vara alltför ytliga. Man måste kunna använda sig av programbibliotek, villkorssatser och funktioner m.m.

Detta kräver emellertid att läraren behöver besitta en del erfarenhet av programmering för att åstadkomma detta. Eftersom långt ifrån alla, för att inte säga endast ett fåtal, matematiklärare i dagsläget har denna erfarenhet kan man inte räkna med att programmeringen i de flesta matematikklasser kan hålla en så hög nivå. Då är vi inne på temat med jämlikhet i utbildningen gällande användandet av digitala resurser, som nämndes i inledningen (se s.4).

Den enda källa, sanktionerad av det svenska utbildningsväsendet, som kommit till författarens kännedom och i något avseende konkret berör nivån på programmeringen är föreläsningen i temat programmering i matematiken från utbildningsradion [21]. Där betonade föreläsaren å andra sidan att ur elevperspektivet inte lägga programmeringen på för hög nivå initialt, samt att låta programmeringen följa matematiken. Frågan är naturligtvis därför hur hög nivå programmering då kan förväntas ha i förhållande till den relativt låga nivå kursen matematik 1c har i matematikavseende och när gör avancerade programmeringskonstruktioner mer nytta än skada.

5.5 Inget ökat intresse för programmering

I diskussionen som föregått införandet av programmering som obligatorisk del i några mattekurser framlades bland annat en förhoppning att stärka elevernas digitala kompetens inför framtiden [11] och att främja flickors intresse för it [12], liksom att kunskapsinhämtningen även i andra ämnen skulle påverkas positivt av programmering [11] (se s.15-16). De eventuellt långsiktiga positiva effekterna låter sig naturligtvis inte mätas såhär kort efter införandet eller i ett arbete av sådan begränsad omfattning som detta. Men den tredje frågeställningen (3) avsåg i alla fall att undersöka huruvida intresset för programmering som sådant ökat då det införts i matematiken. Något sådant kunde emellertid inte påvisas utifrån materialet. Majoriteten ansåg att de inte fått ett ökat intresse för programmering, oavsett om man grupperar efter kön eller lärarens programmeringserfarenhet. Denna fråga var emellertid behäftad med viss komplikation, eftersom att de, som redan hade ett stort programmeringsintresse, inte fick ökat intresse är kanske inte säger så mycket (har man intresse så har man). Frågan borde kanske formulerats på något sätt att den bara mätte i vilken utsträckning elever med ej tidigare programmeringsintresse möjligen fått ökat dito p.g.a. matteprogrammeringen.

Hur som helst angav en mycket liten andel att intresset ökat. Likaså hade få en tidigare större erfarenhet av programmering, vilket de rimligen borde haft om intresse funnits, varför en kvalificerad gissning kan göra gällande att få elever i underlaget hade ett programmeringsintresse och programmeringen i matematiken har inte gjort en denna grupp markant mindre.

5.6 Könsskillnader

Den sista frågeställningen (4) avsåg slutligen att utröna huruvida någon skillnad i resultat på de övriga frågorna gick att skönja relaterat till informantens kön. Särskilt intressant hade förstås varit om det framkommit att tjejer i hög utsträckning upplevt att de fått ett större programmeringsintresse med tanke på den sneda könsfördelningen bland mjukvaruutvecklare på arbetsmarknaden [7] och politikernas förhoppning att främja just flickors it-intresse [12]. Det visade sig emellertid inte helt komplikationsfritt att utifrån det undersökta materialet uttala sig huruvida tjejer eller killar var mest "positiva" respektive "negativa" i någon av frågorna eftersom svaren fördelade sig på olika sätt. I samtliga graderingsbara frågor (upplevd svårighetsgrad med programmering, ökad förståelse för matematik p.g.a. programmering, ökat intresse för programmering) angav förvisso en lite större andel tjejer än killar det mest negativa

alternativet. Killarnas andel var större än tjejernas på den näst mest negativa alternativet. Men andelen tjejer var också större än andelen killar på det näst mest positiva alternativet och där var skillnaden mellan tjejer och killar större. (se figurerna 4, 5, 6 i föregående kapitel). Därför kan man utifrån materialet påstå att tjejerna var mer negativt inställda än killarna om man bara tittar på det mest negativa alternativet. Men man kan också påstå att de var mer positivt inställda än killarna om man lägger ihop deras relativa svarsfrekvens i de två mest positiva alternativen kontra de två mest negativa.

Bara det faktum att man INTE entydigt kan se någon könsskillnad i matematikens programmering är emellertid intressant eftersom skillnaden mellan andelen män och kvinnor som jobbar med programmering i näringslivet är så påtaglig (se s.12). Det undersökta materialet ger i alla fall inte stöd för någon tes om biologiskt betingad högre fallenhet hos män än hos kvinnor för programmeringsfärdighet. Snedfördelningen torde ha andra orsaker.

6. Slutsats

Fastän inget entydigt svar på lämpligaste programmeringsspråk framkommit av studien visade sig Python vara vanligt förekommande, välkänt och lätt att komma igång med. Den som är helt rådvill, kan alltså med fördel använda detta och vara i gott sällskap. Programmeringen bör i låga matematikkurser inte ligga på för hög nivå, då den av många upplevs svår, men ändå i tillräcklig omfattning för att göra nytta för matematiken och inte rädas att innefatta användandet av villkorssatser, loopar, funktioner och lämpliga programbibliotek.

Den upplevda nyttan med programmeringen tycks både hos elever och lärare hänga samman med hur lärarens tidigare programmeringsförtrogenhet. För att det skall vara ett användbart verktyg för matematisk problemlösning och även träna det logiska tänkandet i allmänhet i enlighet med skolverkets och politikerns intentioner inför införandet, är det viktigt att tillse att de matematiklärare som skall undervisa programmering i sina kurser får den fortbildning de behöver. Annars riskerar vi få en ojämlig utbildning.

Studien gav inte stöd för ett ökat programmeringsintresse bland eleverna. Skall detta då tolkas som ett misslyckande i frågan att jämna ut den skeva könsfördelningen bland dem som arbetar med programmering? Där tycks ju förvisso intresset vara större hos män än kvinnor.

Nej. Även om de flesta programmerare i dagsläget är män, är de flesta yrkesverksamma människor inte programmerare. Alla behöver inte vara intresserade av programmering i det avseendet. Vissa könsrelaterade skillnader i materialet syntes, men det gick inte säga att det ena eller andra könet upplevt nyttan med programmering större eller mindre, eller att någondera könets allmänna programmeringsintresse ökat mer än det andras. Att fler män än kvinnor i dagsläget programmerar till yrket hör kanske samman med att programmering, inte minst i populärkulturen, länge associerats som en typisk manlig företeelse och med nördkultur.

Genom att låta programmering vara en obligatorisk del av matematiken, även om inte alla elever får ett större programmeringsintresse, så kanske potentiellt programmeringsintresserade tjejer får upp ögonen för programmering, som annars inte fått det p.g.a. nämnda associationer. Detta gäller naturligtvis även potentiellt programmeringsintresserade killar som ratat ämnet p.g.a. nördkulturen.

Med andra ord finns mycket att vinna med att införa programmering i matematiken, både för matematikförståelse och potentiellt för att jämna ut en skev könsfördelning bland yrkesprogrammerare. Viktigt är dock att tillse att alla berörda lärare ute på skolorna får tillräcklig programmeringskunskap för att genomföra detta på ett bra sätt.

7. Förslag på fortsatt forskning

Genom arbetes gång har ett flertal resultat framkommit om hur olika elever uppfattar programmering på olika sätt beroende på ett antal kringliggande faktorer, men få konkreta tips framkom på hur undervisningen kan läggas upp i detalj. Resultatet antyder att lärare som skall undervisa programmering i matematiken inte bör ha alltför skral kunskap i ämnet, och att nivån inte får vara alltför ytlig, för att eleverna skall uppleva att programmeringen gör nytta för den matematiska förståelsen.

Det hade därför varit intressant att i mer detalj titta på några klasser där eleverna upplever att programmeringen *gör* nytta och se hur upplägget har sett ut där. Vilken typ av programmering passar till vilken typ av uppgifter? Gör man detta i en större omfattning skulle det kanske på sikt kunna utmynna i ett ”bibliotek” med lämpliga lektionsupplägg för respektive matematikkurs, vilket skulle sätta en standard och både underlätta för de lärare som skall bedriva undervisningen och minska risken för ojämn utbildningsnivå mellan olika klasser och skolor.

8. Referenslista

[1] Skolverket (2011), *Läroplan, examensmål och gymnasiegemensamma ämnen för gymnasieskola 2011*

ISBN: 978-91-38325-94-0

[Hämtad 2019-04-13 <https://www.skolverket.se/download/18.6bfaca41169863e6a659807/1553964056811/pdf2705.pdf>]

[2] Skolverket (u.å.), *Matematik*

[Hämtad 2019-04-13 från <https://www.skolverket.se/undervisning/gymnasieskolan/laroplan-program-och-amnen-i-gymnasieskolan/gymnasieprogrammen/amne?url=1530314731%2Fsyllabuscw%2Fjsp%2Fsubject.htm%3FsubjectCode%3DMAT%26courseCode%3DMATMAT01c%26tos%3Dgy&sv.url=12.5dfce44715d35a5cdfa92a3>]

[3] Skolverket (2017), *Få syn på digitaliseringen på gymnasial nivå*

ISBN: 978-913832714-2

[4] SOU 2014:13, *En digital agenda i människans tjänst – en ljusnande framtid kan bli vår*

ISBN: 978-91-38-24081-6

[5] Jacobsson, R.M.; Melander, A. (2018), *Att införa programmering i matematik på gymnasiet* (Examensarbete)

[6] Rolandssons, L. (2015), *Programmed or Not. A study about programming teachers' beliefs and intentions in relation to curriculum* (Doktorsavhandling)

ISBN: 978-91-7595-463-9

[7] Statistiska centralbyrån (2019), *Yrkesregistret med yrkesstatistik 2017*

[Hämtad 2019-04-25 från

https://www.scb.se/contentassets/1fe7f957920f4eaf97bddcc0270553f2/am0208_2017a01_sm_am33sm1901.pdf]

[8] Skolöverstyrelsen (1981), *Matematik för treårig naturvetenskaplig linje och fyraårig teknisk linje, Supplement 75*

ISBN: 91-40-70692-3

[9] Skolverket (2014), *Ämne – Matematik (Gymnasieskola före ht 2011)*

[Hämtad 2019-04-15 från <http://ncm.gu.se/media/kursplaner/gym/Gym2000.pdf>]

[10] Regeringskansliet (2011), *It i människans tjänst – en digital agenda för Sverige*

[Hämtad 2019-04-15 från <http://bredbandskartan.pts.se/Documents/Bredbandsstrategier/It%20i%20m%C3%A4nniskans%20tj%C3%A4nst%20-%20en%20digital%20agenda%20f%C3%B6r%20Sverige.pdf>]

[11] Fridolin, G., Hadzialic, A., Kaplan, M, ”Programmering in på schemat i ny skolstrategi”, *Ny Teknik 2015-09-29*,

[Hämtad 2019-04-15 från <https://www.regeringen.se/debattartiklar/2015/09/programmering-in-pa-schemat-i-ny-skolstrategi/>]

[12] Fridolin, G., Damberg, M., ”Vårt löfte till barnen – mer teknik i skolan”, *Aftonbladet* 2016-08-23,

[Hämtad 2019-04-15 från <https://www.regeringen.se/debattartiklar/2016/08/vart-lofte-till-barnen--mer-teknik-i-skolan/>]

[13] Department for Education (2013): *The national curriculum in England*

[Hämtad 2019-04-15 från https://www.gov.uk/government/uploads/system/uploads/attachment_data/file/210969/NC_framework_document_-_FINAL.pdf]

[14] The Royal Society. (2012). *Shut down and restart.*

[Hämtad 2019-03-15 från: <https://royalsociety.org/topics-policy/projects/computing-in-schools/report/>]

[15] Rolandsson, L., Eddeland, S, Dahlberg, U. (2018), *Aktivitetsbank*, (Skolverket 2018),

[Hämtad från: https://larportalen.skolverket.se/LarportalenAPI/api-v2/document/path/larportalen/material/inriktningar/1-matematik/Gymnasieskola/448_matematikundervisningmeddigitalaverktygII_GY/se-aven/Material/01.aktivitetsbank_gy.docx]

[16] Skolverket (2018), *Programmering i gymnasieskola och vuxenutbildning –uppgifter till workshop*

[Hämtad 2019-04-15 från: https://larportalen.skolverket.se/LarportalenAPI/api-v2/document/path/larportalen/material/inriktningar/1-matematik/Gymnasieskola/448_matematikundervisningmeddigitalaverktygII_GY/se-aven/Material/02.Uppgifter%20-%20workshop%20om%20programmering%20i%20gymnasieskolan.docx]

[17] Skolverket (2018), *Programmering i gymnasieskola och vuxenutbildning – lösningsförslag*

[Hämtad 2019-04-15 från: https://larportalen.skolverket.se/LarportalenAPI/api-v2/document/path/larportalen/material/inriktningar/1-matematik/Gymnasieskola/448_matematikundervisningmeddigitalaverktygII_GY/se-aven/Material/04.L%C3%B6sningsf%C3%B6rslag%20-%20workshop%20om%20programmering%20i%20gymnasieskolan.docx]

[18] Skolverket (2018), *Programmering i gymnasieskola och vuxenutbildning – referensblad till workshop*

[Hämtad 2019-04-15 från: https://larportalen.skolverket.se/LarportalenAPI/api-v2/document/path/larportalen/material/inriktningar/1-matematik/Gymnasieskola/448_matematikundervisningmeddigitalaverktygII_GY/se-aven/Material/03.Referensblad%20-%20workshop%20om%20programmering%20i%20gymnasieskolan.docx]

[19] Skolverket (2018), *Verktyg-lärohandledning*

[Hämtad 2019-04-15 från: https://larportalen.skolverket.se/LarportalenAPI/api-v2/document/path/larportalen/material/inriktningar/1-matematik/Gymnasieskola/448-matematikundervisningmeddigitalaverktygII_GY/del_01/Material/Flik/Del_01_Moment_A/Artiklar/MA2_Gy_01A_04_verktyg.docx]

[20] Wikipedia [hämtat 2019-06-09 från: https://sv.wikipedia.org/wiki/Daniel_Barker]

[21] Utbildningsradion (2018), *Programmering i matematiken*

[Sett 2019-04-30 på: <https://urskola.se/Produkter/205025-UR-Samtiden-Matematikbiennalen-2018-Programmering-i-matematiken>]

9. Bilagor

Enkätfrågorna för lärare

Fråga 1

Vilken mattekurs

Denna enkätundersökning gäller mattekurs:

matematik 1c
matematik 2c
matematik 3b
matematik 3c
matematik 4
matematik 5
annan kurs

Om "annan kurs", ange vilken

--

Fråga 2

Programmeringsspråk

**Vilket eller vilka programmeringsspråk använder ni,
när ni löser matematikuppgifter med hjälp av programmering.**

--

Fråga 3

Programbibliotek

Använder ni något speciellt programbibliotek? I så fall vilket / vilka?

Fråga 4

Omfattning

Vilka programmeringstekniska företeelser lärs ut, såsom vilkorssatser, loopar, funktioner, objekt, klasser o.s.v.?

Fråga 5

Förståelse

Programmering som hjälpmedel ökar elevernas MATEMATISKA förståelse där det används, jämfört med om samma uppgifter lösts på traditionellt sätt

Stämmer inte alls Stämmer helt

Frivillig kommentar:

Fråga 6

Egen erfarenhet

Hur omfattande är din egen kunskap och erfarenhet av programmering?

Berätta:

Fråga 7

Övrigt

Här har du möjlighet att skriva några övriga kommentarer som kan vara relevanta i sammanhanget.

Enkätfrågorna för eleverna

Fråga 1

Programmeringsspråk

Vilket eller vilka programmeringsspråk använder ni när programmering används som hjälpmedel i matematiken? (som t.ex. Scratch, Javascript eller Python)

Fråga 2

Lätt eller svårt

Jag upplever *programmeringen* som används i matematiken som:

Svår Lätt

Frivillig kommentar:

Fråga 3

Förenkla matematiken

Matematiken kan bli lättare att förstå tack vare programmering:

Stämmer inte alls Stämmer helt

Frivillig kommentar:

Fråga 4

Intresse för programmering

Jag har fått ett större intresse för programmering som sådant tack vare att det ingår i matematiken.

Stämmer inte alls Stämmer helt

Frivillig kommentar:

Fråga 5

Erfarenhet

Har du erfarenhet från programmering sedan tidigare (innan du använde det i denna mattekurs).

T.ex. från andra skolämnen eller fritiden?

Berätta:

Fråga 6

Bakgrund

Jag är:

Tjej Kille

Fråga 7

Övrigt

Har här du möjlighet att skriva några övriga kommentarer som kan vara relevanta i sammanhanget: