



Programmering i matematikämnet på gymnasiet

Hur gymnasielärare har upplevt implementeringen av
programmering i matematiken

Hampus Eriksson

Programmering i matematikämnet på gymnasiet

Hur gymnasielärare har upplevt implementeringen av
programmering i matematiken

Hampus Eriksson

**EXAMENSARBETE INOM TEKNIK OCH LÄRANDE PÅ
PROGRAMMET CIVILINGENJÖR OCH LÄRARE**

Titel på svenska: Programmering i matematikämnet på gymnasiet.

Titel på engelska: Programming in mathematics in Swedish high school.

Handledare: Ida Naimi-Akbar, Kungliga Tekniska Högskolan.

Handledare: Helena Isaksson Persson, Kungliga Tekniska Högskolan.

Examinator: Arnold Pears, Kungliga Tekniska Högskolan.

Förord

Nu börjar mina år på KTH lida mot sitt slut. Från ett första fantastiskt mottagande från mottagare och min adeptgrupp Ares har jag känt att CL verkligen har varit rätt plats för mig. Programmet har gett mig kunskaper inom både lärande och teknik, men framförallt så har jag utvecklats som person och fått vänner för resten av livet.

Jag vill börja med att tacka mina handledare Helena och Ida som sedan vårt första möte stöttat och pushat mig till att vrida och vända på arbetet vilket har taggat mig och gjort att arbetet nått en högre nivå. Tack också till examinator Arnold som både har bidragit med feedback efter ventiler och de arbeten som har använts som referenser i mitt arbete.

Enormt tack till Evelina som har pushat på mig när jag behövt motivation. Med denna speciella vår som ligger bakom oss så har det varit skönt att vara två i arbetet även om vi har skrivit våra egna separata arbeten.

Till de lärare som ställde upp på intervjuerna så att denna studie gick att genomföra vill jag rikta ett stort tack.

Sammanfattning

Hösten 2018 skedde en förändring i matematikämnets kursplan på gymnasiet. Förändringen berörde området problemlösning där programmering nu ska ingå som ett digitalt verktyg för problemlösning.

Syftet med detta examensarbete är att identifiera vilka utmaningar och möjligheter gymnasielärare i matematik har upplevt sedan programmering implementerades i matematikämnet, samt att undersöka vilka strategier som lärare har använt för att programmeringen ska kunna användas i matematiken.

Studiens data samlades in med hjälp av kvalitativa intervjuer med gymnasielärare på naturvetenskaps- och teknikprogrammet. Datan analyserades sedan med hjälp av tematisk analys.

Lärarna i denna studie har upplevt mer utmaningar än möjligheter med programmeringen i matematiken. De största utmaningarna som beskrivs av lärarna är att planeringen tar längre tid och att elevernas förkunskaper är bristfälliga. Programmering ska användas till problemlösning inom matematiken men det som istället har hänt är att mycket tid måste tas av lektionerna för att matematiklärarna ska lära ut grunderna inom programmering till eleverna.

Nyckelord: Programmering, matematik, gymnasiet, problemlösning

Abstract

In the Fall of 2018, a change in the mathematics syllabus took place in the Swedish high school. The change affected the area of problem solving, where programming is now to be included as a digital tool for problem solving.

The purpose of this Master's thesis is to identify what challenges and opportunities high school teachers in mathematics have experienced since programming was implemented in the mathematics subject, and to investigate what strategies teachers have used for programming to be able to be used in mathematics.

The data was collected through interviews with high school teachers in the natural science and technology program. The data was analyzed using thematic analysis.

The teachers in this study have experienced more challenges than opportunities with programming in mathematics. The biggest challenges described by the teachers are that planning takes longer and that students' prior knowledge is inadequate. Programming should be used for problem solving in mathematics, but what has instead happened is that a lot of time must be taken from the lessons for the mathematics teachers to teach the basics of programming to the students.

Keywords: Programming, mathematics, high school, problem solving

Innehåll

1. Inledning	15
1.1. Bakgrund	15
1.2. Syfte & Frågeställningar	3
1.3. Avgränsningar	3
1.4. Arbetets koppling till ingenjörspessionen	4
2. Tidigare forskning	5
2.1. Undervisning i programmering	5
2.2. Svårigheter med programmering	6
2.3. Positiva aspekter med programmering	7
2.4. Problemlösning	7
2.5. Problemlösning med programmering	8
2.6. Förändringen i kursplanen	8
2.7. Lärarens roll i implementering av programmering i skolan	9
2.8. Stöd efter beslutet om förändring i kursplanen	10
2.9. Digitalisering i resten av världen	11
3. Metod	13
3.1. Urval	13
3.2. Intervjuer	13
3.3. Analys	14
3.4. Metoddiskussion	14
4. Resultat & Analys	16
4.1. Kompetensutbildning för lärare	16
4.2. Elevernas kunskaper inom programmering	17
4.3. Tidsbrist	18
4.4. Kursplaner	19
4.5. Förbereda eleverna för vidare studier	20
4.6. Programspråken	20
4.7. Digitala verktyg	21
4.8. Grundkurs i programmering	22
5. Diskussion	23
5.1. Utmaningar	24
5.2. Möjligheter	27
5.3. Strategier	28
6. Slutsats	29
7. Referenser	31
8. Bilagor	35
8.1. Intervjuguide	35

1. Inledning

1.1. Bakgrund

Såväl globalt som nationellt står samhället inför en mängd utmaningar framöver, exempelvis klimatfrågan, ekonomiska frågor och jämställdhet. För att möta dessa utmaningar kommer digitalisering och IT ha en bidragande roll. (Regeringen, 2012) Sett till de globala målen kan det ses i mål 4, god utbildning för alla, att oavsett var man bor i världen så ska det finnas lika tillgänglig förskola, skola och högre utbildning som är gratis och av god kvalitet. (UNDP, n.d) Digitalisering i sig är inte ett av de globala målen men digitaliseringen kan bidra till att målen uppnås. Ett stort användningsområde för digitaliseringen är samarbete över gränser, där digitalisering kan bidra till utbyte av kunskaper och teknik mellan olika delar av världen då de globala målen är just globala. (Digitaliseringsrådet, n.d) Via digitalisering skulle undervisning på alla nivåer samt kompetensutveckling för lärare kunnas göra digitalt från länder som har mer kunskap och erfarenhet till länder där det inte finns lika mycket resurser.

I slutet av 2011 beslutade regeringen om en sammanhållen agenda för Sveriges it-politik där ambitioner inom digitalisering samt förslag på insatser presenterades. Målet var att "Sverige ska vara bäst i världen på att använda digitaliseringens möjligheter". (Regeringen, 2012) Med bakgrund i agendan gavs Digitaliseringskommissionen i uppdrag att utforma ett förslag på handlingsplan för genomförande av uppdraget att nå det it-politiska målet. Kommissionen skulle analysera utvecklingen i förhållande till det it-politiska målet och visa på vilka möjligheter digitalisering kunde bidra till.

I sitt delbetänkande skrev Digitaliseringskommissionen en rad förslag till grund- och gymnasieskolan. De ansåg att de styrdokument som började användas i och med Lgy11 inte var tillräckliga för att tillgodose de behov som dagens utveckling inom digitalisering innebär. Ett av förslagen var att Skolverket skulle ges i uppdrag att föreslå en uppdatering av Lgy11 så att begreppet digital kompetens skulle bli en basfärdighet. Skolverket gavs även i uppdrag att revidera kursplanerna i ämnena bild, historia, matematik, samhällskunskap, slöjd, svenska och teknik. (Digitaliseringskommissionen, 2014) Kommissionen ansåg att IT i undervisning inte hade förändrats mycket och föreslog därför en satsning i hela Sverige där målet var att den digitala kompetensen hos lärarna skulle öka. Skolverket föreslogs ges i uppdrag att ta fram riktlinjer, upplägg och en beräkning på kostnaden för denna satsning och efter det genomföra satsningen.

Digitaliseringskommissionens förslag utmynnade i att regeringen 2015 gav skolverket i uppdrag att föreslå it-strategier för skolan, allt från förskola till gymnasieskola. (Utbildningsdepartementet, 2015) Skolverket hade i mitten på 2016 fullfört sitt uppdrag från digitaliseringskommissionen och hade tagit fram förslag på förändringar i styrdokumentet, med avseende att förstärka och tydliggöra den digitala kompetensen i styrdokumentet. (Skolverket, 2016)

Den 9:e mars 2017 beslutade regeringen om förtydliganden i läroplaner, kursplaner och

ämnesplaner för både grundskolan och gymnasieskolan. Inom gymnasieskolan så innebar detta mindre förändringar i kurserna matematik, samhällskunskap, svenska och svenska som andraspråk. (Regeringskansliet, 2017) Syftet med förändringarna var att "tydliggöra skolans uppdrag att stärka elevernas digitala kompetens".

Hösten 2018 infördes en förändring i kursplanerna i matematik på gymnasiet som innebar att kurserna i c-spåret ska använda programmering som ett digitalt verktyg för problemlösning. I c-spåret ingår kurserna matematik 1c, matematik 2c, matematik 3c, matematik 4, matematik 5 samt matematik specialisering. (Regeringskansliet, 2017) Gymnasieprogrammen i naturvetenskap och teknik är de två program där kurser i c-spåret i matematik är obligatoriska. Alla elever på dessa program läser minst de tre första kurserna och har möjlighet att välja till de andra tre. (Skolverket, n.d.a)

Den ändringen som skedde i matematiken där programmering nämns i kursplanen ligger under rubriken problemlösning i c-spårets kurser och tabell 1 visar på den punkt som nu innehåller programmering i matematikämnets kursplan samt hur den punkten såg ut innan HT18. (Skolverket, n.d.b)

Tabell 1. Den punkt i matematikämnets kursplan som berör digitala verktyg, innan HT18 och från och med HT18.

Kursplan i c-spåret innan HT18	Kursplan i c-spåret från och med HT18
<i>Strategier för matematisk problemlösning inklusive <u>användning av digitala medier och verktyg.</u></i>	<i>Strategier för matematisk problemlösning inklusive <u>modellering av olika situationer, såväl med som utan digitala verktyg och programmering.</u></i>

Tabell 2 beskriver vilka programmeringskurser som finns på gymnasiet och vilka av de nationella programmen som läser vilka kurser. Med valbar menas att "kursen får erbjudas som programfördjupning inom programmet" eller att i teknikprogrammets fall ingå i en av programmets inriktningar, informations- och medieteknik. (Skolverket, n.d.c) Som tabellen visar så läser endast teknik med inriktning informations- och medieteknik programmering som obligatorisk kurs.

Tabell 2. Programmeringskurser på gymnasiet samt vilka program som har möjlighet att läsa dessa.

Programmering 1	Valbar: Alla
Programmering 2	Valbar: EI & Energi, Industritekniskt, Natur, Teknik
Tillämpad Programmering	Valbar: Alla
Webbserverprogrammering 1	Valbar: EI & Energi, Natur, Teknik
Webbserverprogrammering 2	Valbar: EI & Energi, Natur, Teknik

1.2. Syfte & Frågeställningar

Syftet med denna studie är att identifiera vilka utmaningar och möjligheter lärare i matematik på gymnasiet har upplevt sedan implementeringen av programmering i matematikämnet, samt att undersöka vilka strategier som lärare har använt för att programmeringen ska kunna användas i matematiken.

De frågeställningar som studien ämnar att besvara är:

Vilka möjligheter och utmaningar har gymnasielärare i matematikämnet upplevt sedan introduktionen av programmering för problemlösning på programmen naturvetenskapligt- och tekniskt program?

Vilka strategier för implementering och undervisning av programmering i matematikämnet har använts av lärarna?

1.3. Avgränsningar

Studien har avgränsats genom att endast undersöka hur förändringen av kursplanen har påverkat undervisningen för lärarnas del och helt utelämnat hur förändringen har varit utifrån elevernas perspektiv. Studien undersöker endast naturvetenskaps- och teknikprogrammet och andra program har uteslutits vid utförandet av studien då endast naturvetenskaps- och teknikprogrammet obligatoriskt läser matematikkurser med programmering i kursplanen.

1.4. Arbetets koppling till ingenjörsprofessionen

CDIO (2011), som är ett samarbete mellan tekniska universitet vars syfte är att förbättra kurser och utbildningar (Högfeldt & Kann, 2012), har listat fyra stycken huvudegenskaper som en ingenjör enligt dem bör ha.

- Kunskaper inom matematik, vetenskap och teknik samt metoder och verktyg inom dessa områden. Beroende på ingenjörens inriktning kan kunskaper inom andra områden tillkomma.
- Personliga förmågor såsom problemlösning och resonemang, nyfikenhet, systemtänk, etik och jämställdhet.
- Samarbete och kommunikation. Samarbete syftar till att arbeta i ett arbetslag där ingenjören kan leda arbetslaget framåt. Kommunikation syftar på skrift, muntlig kommunikation, grafisk kommunikation samt digital kommunikation. Ingenjören ska kunna kommunicera på både språket som den arbetar i samt på engelska.
- Utveckling av en produkt eller system från design till genomförande, hänsyn till företag, samhälle och miljö.

Detta examensarbete kopplar starkt till de två första punkterna då matematik, problemlösning samt verktyg är en central del av undersökningen. Arbetet undersöker hur förändringen i kursplanen har påverkat lärare som undervisar inom matematik för det naturvetenskapliga och tekniska programmen på gymnasiet. De elever som går dessa två program har möjlighet att studera vidare på tekniska linjer då båda programmen är högskoleförberedande.

2. Tidigare forskning

Detta kapitel ämnar att ge en inblick i tidigare forskning inom programmering, problemlösning och relationer mellan dessa och undervisning, samt ge en teoretisk grund till rapportens avslutande diskussion.

2.1. Undervisning i programmering

Sangwin & O'Toole (2017) definierar programmering som en process som börjar med ett problem och slutar med ett körbart program, ofta kallat för kod. Det finns väldigt lite som talar för att något specifikt tillvägagångssätt är det bästa sättet att lära sig programmering då flera olika sätt kan vara vägen till samma mål (Pears et al., 2007).

Traditionellt sett så är syntax och struktur av själva programmeringsspråket prioriterat i undervisning om programmering, speciellt om det är en grundläggande kurs (Pears et. al, 2007; Malik & Neilson, 2017). Majoriteten av kurslitteraturen om programmering berör just detta. Att välja ett språk till undervisningen kan påverkas av faktorer såsom vilka preferenser skolan har (om de t.ex har bestämt att hela skolan ska endast använda Python), vilken relevans de har för det arbetsområde som de berörda studenterna kommer arbeta med i framtiden, tekniska aspekter för språket samt tillgängligheten. Valet av språk kan också bli mer komplext med tiden då fler och fler programmeringsspråk och program blir tillgängliga. Enkelt syntax gör att användaren snabbare kan lära sig ett språk och i sin tur börja lösa problem snabbare. (Pears et al., 2007)

Mannila et al. (2014) skriver att Scratch eller liknande block-miljöer kan vara en bra start att lära sig programmera i då det nästan är fritt från syntax och användaren får direkt återkoppling när programmet körs. Scratch är ett språk som använder blockprogrammering. Språket används främst för barn eftersom blocken gör programmeringen mer konkret och ingen kod behöver skrivas. I Scratch är det svårare att göra "fel" då blocken sätts ihop av användaren, ungefär som pusselbitar. (Calder, 2019)

Även om en person kan de olika syntaxen och funktionerna i ett språk betyder det inte att personen kan kombinera detta till ett program som löser givna problem. Lära sig programmering innebär att man både måste lära sig ytkunskaper, syntaxen och strukturen av ett programmeringsspråk, men också djupare kunskap, logiken av ett språk och hur olika funktioner kan kombineras för att lösa problem. (Malik & Neilson, 2017)

2.2. Svårigheter med programmering

Robins, Rountree och Rountree (2003) har jämfört nybörjare i programmering med experter. De definierar en nybörjare som en som tänker väldigt ytligt och inte har samma visualiseringsförmåga som mer erfarna programmerare. Nybörjarna har svårt att applicera sin kunskap, ser inte helheten och har en generell sämre förståelse om vanliga begrepp inom programmering, så som variabler och loopar. Författarna listar fem aspekter som nybörjare har svårt med och som de måste bli arbete med för att bli skickligare programmerare:

- Vad ett program är och vad ett program kan användas till
- Hur en dator fungerar i relation till datorprogram
- Syntaxen i programmeringsspråket
- Strukturen av programmeringsspråket
- Planering, utveckling, testning och felsökning av ett program

Nybörjare planerar kortare tid än vad en expert gör och eftersom nybörjarna inte ser helheten på samma sätt tenderar nybörjarna att fixa problemen där de finns utan att tänka på programmet som en helhet. De testar även sina program för lite. Nybörjare kan delas upp i två grupper när det gäller testande. Stoppare (eng. stoppers) stannar lätt när de ser ett problem som de inte direkt vet hur de ska lösa eller har en tydlig väg att gå för att lösa problemet. De som blir frustrerade när programmet visar error löper stor risk att bli en stoppare medans flyttare (eng. movers) fortsätter och försöker ändra i koden genom att testa om och om igen. (Robins et al., 2003) Att nybörjare inte ser helheten på programmet skriver även Teague & Lister (2014) om. De beskriver fem steg av programmerare där helheten och relationerna mellan olika kodrader är en av faktorerna som gör att en person blir bättre på programmering.

Aspekter som påverkade elevens förmåga att lära sig programmering var enligt Qian & Lehman (2017) vilka språkkunskaper (speciellt i engelska) eleven har, valet av programmeringsmiljö och lärarens kunskaper om programmering och hur det kan läras ut. Även elevens matematikkunskaper kan påverka eftersom ibland så "krockar" matematiken och programmeringen, t.ex då en "integer" alltid är heltal så om man sätter en variabel av typen integer till $\frac{1}{2}$ blir värdet på integern 0. I matematiken har variablerna inte alltid en gräns medans i programmeringen så begränsas variablerna med avseende på språket.

2.3. Positiva aspekter med programmering

Psycharis och Kallia(2017) skriver att de studier som visar på att programmering påverkar elevernas förmågor positivt listar olika aspekter med programmeringen. Den mest förekommande positiva aspekten är att problemlösningsförmågan blir bättre med hjälp av programmering. Att lära sig programmera bidrar också till förbättrad förmåga att resonera, ökar samarbetsförmågan och det matematiska tänkandet samt höjer kreativiteten.

2.4. Problemlösning

I kursplanen för matematik på gymnasiet(Skolverket, n.d.b) finns det sju formulerade förmågor, där en berör rutinuppgifter:

Hantera procedurer och lösa uppgifter av standardkaraktär utan och med verktyg.

Och en berör problemlösning:

Formulera, analysera och lösa matematiska problem samt värdera valda strategier, metoder och resultat.

I syftesbeskrivningen för matematikämnet står det att eleverna ska utveckla strategier för att lösa matematiska problem och använda problemlösning som både mål och medel.

Ett problem är enligt Hansson (2019) en uppgift där eleverna ska undersöka något utifrån angiven information i uppgiften. Hon definierar en rutinuppgift som en uppgift "där metoderna är kända av eleven" och en naken uppgift som en uppgift "där endast matematiska symboler används". Det är viktigt att ett problem får fram ett engagemang hos eleven då en uppgift endast kan ses som ett problem om den ställs till en person vars kunskapsnivå inte gör att uppgiften blir för lätt, en rutinuppgift, och inte heller gör att uppgiften blir för svår.

Ett rikt problem ska enligt Taflin (2007) vara lätt att förstå och alla ska ha en möjlighet att arbeta med det. Problemet ska upplevas som utmanande, kräva ansträngning och tillåtas ta tid. Ett rikt problem ska kunna lösas på flera olika sätt, med olika strategier och representationer. Problemet ska kunna bidra till att starta en matematisk diskussion utifrån elevernas olika lösningar, en diskussion som visas på de olika strategierna och idéerna som användes. Problemet ska introducera eleverna till viktiga matematiska idéer, kunna fungera som brobyggare mellan olika matematiska områden och kunna leda till att elever och lärare formulerar nya intressanta problem.

På skolverkets hemsida läses att de vid problemlösning föreslår ett arbetssätt som är uppdelat i fyra faser: Introduktion av problem, enskilt arbete, arbete i smågrupper för att sedan sluta i en helklassdiskussion (Skolverket, 2018).

2.5. Problemlösning med programmering

Problem inom programmering kan delas upp i två typer, problem som är likt de problem du tidigare stött på och helt nya problem (Pears et al., 2007). Pears et al. (2007) beskriver ett förlopp med tre stadier från att en person är nybörjare på programmering tills att den kan använda sina förmågor i andra kontexter. Första steget är att personen lär sig ett programmeringsspråk och strukturen av det språket. Andra steget är att genom att kombinera strukturer och funktioner i ett språk kan större problem lösas. Tredje och sista steget är där när personen har utvecklat mer generella förmågor inom problemlösning, som både kan användas i andra programmeringsspråk men också i andra områden, såsom matematik.

Palumbo (1990) beskriver skillnaden mellan en nybörjare och expert som att experterna kan processa större mängder information än nybörjarna och på så sätt lösa problem snabbare. Experterna kan även lösa flera steg i taget medans nybörjarna måste göra nya val vid varje steg i problemlösandet. Experter har mer kunskap inom programmering än nybörjare vilket kan innebära att de har kunskap om en känd algoritm som kan lösa problemet medans en nybörjare behöver tänka ut en lösning själv.

2.6. Förändringen i kursplanen

Heintz et al. (2017) har gjort en jämförelse mellan implementeringen av IT i skolan i Sverige och i England. I England har de ett dedikerat ämne som heter "Computing" där den som undervisar är en utbildad lärare i just datorkunskap. Anledningar som nämns att Sverige inte införde ett nytt ämne är brist på plats i kursplanerna, studenterna får se och använda digitala verktyg i många olika ämnen och hur de kan använda digitala verktyg för att lösa problem inom olika områden. Nedanstående citat från Heintz et al. (2017) visar på att något som måste tas i åtanke nu i början av implementeringen är att alla elever kommer börja från samma nivå så att oavsett om eleven går på lågstadiet eller gymnasiet måste någon slags grund läggas.

"The progression in the revised curriculum is expressed as if the curriculum has already reached a steady state. This leaves local authorities throughout the country with the responsibility of implementing the changes and solving the bootstrapping challenge."

Kursplaner utan tydliga direktiv kan vara bra till en början då lärare kan lägga undervisningen på en nivå de känner sig bekväma med och som funkar för eleverna. De kan göra så mycket som de känner räcker för att nå målen och sedan höja nivån när det känns bekvämt. Problemet med detta är att vissa lärare kanske aldrig höjer ribban och ligger kvar på en låg nivå. Ett annat problem är att lärare kan anse att det är andra lärare som ska lära ut IT då ingen dedikerad lärare finns för detta. (Heintz et al, 2017) Mannila et al (2014) skriver att ett bra sätt att introducera datalogiskt tänkande i kursplanen är att föreslå för lärarna hur teknik kan användas utifrån nuvarande kursplaner och kursmål.

2.7. Lärarens roll i implementering av programmering i skolan

Utmaningar som nämns av Humble et al. (2019) är att det tar mycket tid och engagemang för läraren att nå till en nivå så att programmeringen blir användbart i undervisningen. När lärare tillfrågas om utmaningar med implementering av programmering i skolan är svaren mer snarlika än om de tillfrågas om möjligheterna med implementeringen. Även Ekberg & Gao (2018) beskriver att lärarna anser att tiden är den största utmaningen. De aspekter som nämndes var tid att planera lektionen, tid att lära sig program/verktyg och tiden det tar att hitta digitala resurser. Humble et al. (2019) nämner även att programmering är svårt att lära sig om man inte har någon tidigare erfarenhet samt att mycket material om programmering är på engelska vilket vissa lärare tyckte var en utmaning.

En del av lärarna nämnde en positiv aspekt i att programmering är roligt för eleverna och att det finns mycket material online. De flesta av lärarna som datan är baserad på använder programspråket Python när ren kodning ska göras och Scratch när blockprogrammering används. (Humble et al., 2019)

Något som påverkar implementeringen av programmering är att lärarna har tillgång till för få resurser. Lärarna har sina datorer och office-paket, men inte så mycket mer. De måste istället hitta program som är gratis online och lära sig dessa. Lärarna har inte stor tillit till teknik och måste därför planera en backup-plan om tekniken skulle strula på lektionen, t.ex genom att skriva ut papperskopior. Om något är fel med tekniken kan det inte heller alltid lösas av läraren, utan en elev kan behöva gå till IT-supporten med sin dator eller surfplatta. Tilltron till tekniken är ännu lägre för de äldre lärarna och de var mindre benägna att använda tekniska hjälpmedel i undervisningen. (Ekberg & Gao, 2018)

Götling & Löfwenhamn (2018) undersökte vilka utmaningar och styrkor som både lärare och elever uppfattade med införandet av programmering i matematikämnet på gymnasiet. En av de största brister de uppfattade var bristen på tid. Både lärare och elever tyckte att det kändes svårt att hinna med fler moment. Eleverna såg en utmaning i själva syntaxen och att det ibland var så att eleverna inte ville programmera i framtiden då syntaxen verkade så svår. En styrka som de såg i att använda programmering är att undervisningen blir mer varierad vilket i sin tur kan göra så att eleverna blir mer motiverade att lära sig matematik.

I en studie som Belgheis & Kamalludeen (2018) genomförde ingick 132 matematiklärare i Malaysia och deras resultat visade på att det finns ett starkt samband mellan hur kompetent och bekväm en lärare är med ett verktyg (i detta fall Geogebra) och hur troligt det är att läraren använder verktyget i undervisningen. En anledning som kan göra att lärarna väljer bort att använda ett verktyg är att de inte fått utbildning om verktyget.

Mannila, Nordén och Pears (2018) skriver om lärarnas förmåga att lära sig på egen hand. Att lära sig själva digitala verktygen, programmering i matematikens fall, är en del av lärandet.

Enligt författarna är det även viktigt att lärarna blir självsäkra och självständiga i att utforska hur tekniken förändras och hur digitala verktyg och kompetenser kan användas i undervisningen. De har analyserat svar från 530 lärare och kommer fram till att det finns stora utmaningar i denna självständighet. För att vi ska kunna ha undervisning som håller jämna steg med digitaliseringen så måste skolor och myndigheter ha kontinuerlig utbildning för lärarna så att de har verktygen för att lyckas i sin undervisning. I svaren från studien är programmering och hur programmering kan användas några av de kompetenser som lärarna har rankat som svårast. Ragonis, Gal-Ezer och Hazzan (2010) skriver att med tanke på vilka speciella egenskaper programmering och datorkunskap har så behövs specifik utbildning för lärarna och att en lärare ska inte få undervisa i datorrelaterade ämnen om läraren inte är utbildad i dessa.

För att eleverna ska kunna lära sig inom dagens teknik kräver det att lärarna fortbildar sig med ny kunskap eftersom tekniken förändras så snabbt. En lärares kompetens har stor påverkan på elevers prestationer och lärande. (Mannila et al, 2014) För att programmering och IT ska läras ut framgångsrikt i skolan krävs en balans mellan teori och praktik och att lärarna har kompetens och motivation att fortsätta lära sig (Dagiene & Stupurienė, 2016).

2.8. Stöd efter beslutet om förändring i kursplanen

Skolverket erbjuder alla som arbetar i skola olika typer av utbildningar och kurser. I skrivande stund så finns det 80 aktuella kurser och utbildningar på skolverkets hemsida varav 21 kurser är kopplade till digitalisering. I programmering så har de en kompetensutveckling som är uppdelad på tre nivåer. (Skolverket, n.d.d)

Inom programmering erbjuder skolverket en kompetensutveckling i tre steg som presenteras nedan (Skolverket, 2020):

Nivå 1 - Allmänbildande programmering

En webbkurs som alla som arbetar i skolan bör gå om de inte har tidigare kunskap om programmering. Tar ca 16 timmar att genomföra, och genomförs på webben.

Nivå 2 - Grundläggande programmering

Denna nivå är till för lärare i matematik och teknik på högstadiet samt gymnasiet. Kan läsas på distans eller på ett universitet.

Nivå 3 - Programmering och didaktik

I den sista nivån finns det olika val av kurser beroende på de kurser du undervisar i (matematik eller teknik).

På samma sätt som skolverket har en lista på utbildningar och kurser så finns det en lista på stödmaterial. Det finns 178 stödmaterial varav 19 är kopplade till digitalisering. (Skolverket, n.d.e)

2.9. Digitalisering i resten av världen

Implementering av digitalisering av skolan är inte ett nationellt fenomen utan en process som pågår runt om i världen. Nedan presenteras två av länder som valdes eftersom båda har implementerat digitaliseringen annorlunda jämfört med Sverige.

2.9.1. Storbritannien

I början av 2012 släppte "The Royal Society" en rapport som kallades "Shut down or restart? - The way forward for computing in UK schools" som hade undersökt hur datoranvändningen (eng. computing) användes i undervisningen. De kom fram till att dåvarande undervisningen inom datorkunskap var mycket otillfredsställande. Anledningen till detta var att det ämne som datorkunskap lärdes ut i (informations- och kommunikationsteknik (eng. ICT)) var skrivet väldigt brett vilket gjorde att skolan kunde välja att lägga undervisningen på en låg nivå, speciellt om den undervisande läraren själv inte hade mer kunskaper än excel och word, vilket många inte hade. (The royal society, 2012)

Förslagen som de kom fram till var att granska kursplanen för ICT och göra det tydligare vilka olika områden som ska ingå i kursen. Namnet på kursen ska också bytas ut då många hade en negativ syn på kursen. Fler lärare behöver utbildas inom datoranvändandet och ett förslag till detta var att deras regering skulle sätta ett mål för hur många lärare som skulle utbildas och sedan finansiera dessa utbildningar. Alla skolor måste ha tillgång till tillräckliga resurser för att undervisa i datorkunskap, t.ex programmeringsbara robotar. Det nämns att de tycker att precis som det ska vara självklart att man kan skriva och läsa när man går ut skolan, så ska varje elev vara digitalt kunnig (eng. digitally literate) och att eleverna ska ges möjlighet att lära sig grunder inom datavetenskap innan de väljer mer inriktad skolgång. (The royal society, 2012)

Akademien släppte ytterligare en rapport i slutet av 2017, som var en typ av uppföljning till tidigare nämnda rapport, med både vad som hade förändrats åt det positiva sedan 2012 men också förslag på vad som fortfarande behövde förändras. I England är det nu obligatoriskt med ämnet "computing", där datorvetenskap (eng. Computer science), digital kunnighet (eng. digital literacy) och IT är de områdena som berörs mest, för elever mellan fem till sexton år. Ett stort problem är dock fortfarande att det inte finns tillräckligt många lärare som har tillräckliga kunskaper för att lära ut i dessa datorämnen. De skriver att för att verkligen kunna nå målet att alla elever ska få möjligheter att lära sig maximalt inom datorämnena så måste lärarna ges utbildning så att de kan utveckla sin egen kunskap för att sedan kunna lära ut det till eleverna. (The royal society, 2017)

2.9.2. Finland

I Finland började nya kursplaner gälla hösten 2016 för grundskolan(år 1-6) och högre årskurser följde sedan gradvis efter. Syftet med förändringen var att förbereda eleverna inför ett mer komplext och digitalt samhälle. (Skolvärlden, 2015)

2019 färdigställdes den nya kursplanen för Finlands gymnasieutbildning och den ska tas i bruk hösten 2021. (Utbildningsstyrelsen, n.d.a)

De nya kursplanerna för gymnasiet talar mycket om digitala miljöer, digitala verktyg samt förändringen i samhället i och med digitaliseringen. Just programmering nämns dock endast två gånger, och det är i en valbar kurs i matematik, MAA11 Algoritmer och talteori. (Utbildningsstyrelsen, n.d.b)

Om man jämför kursplanen för gymnasium och kursplanen för grundskola i Finland så ser man att eleverna introduceras till programmering redan i de första årskurserna och i resterande grundskola så nämns programmering mycket i kursplanerna. Grundskolan har även sju kompetenser som ska ge eleverna en mångsidig kompetens, där digital kompetens är en av dem. (Åbo, n.d) På gymnasiet är denna mångsidiga kompetens uppdelad i sex olika kompetenser, men då är inte längre digital kompetens en av dem. Programmering nämns endast i en kurs i kursplanen för gymnasiet. (Utbildningsstyrelsen, n.d.b)

3. Metod

Detta kapitel beskriver de metoder som använts för denna undersökning. Kapitlet behandlar urvalet till intervjuerna, vilken typ av intervjuer som användes, hur analysen av resultatet gick till samt vilka etiska riktlinjer som undersökningen har följt.

3.1. Urval

Denna undersökning har utförts med ett kumulativt tillvägagångssätt gällande antal intervjuer. Det innebär att intervjuer kommer fortsätta tills dess att flera intervjuer inte kan tillföra ytterligare information till datan. (Denscombe, 2017) I praktiken innebar detta att sju stycken intervjuer bokades in. Efter initial genomlysning av dessa intervjuer så togs beslutet att inga fler intervjuer skulle planeras in då fler intervjuer inte förväntades bidra till nya resultat.

De som deltagit i intervjuerna har valts ut med hjälp av ett subjektivt urval, vilket innebär att ett kriterium har formulerats (Denscombe, 2017). I detta fall var kriteriet lärare som undervisar i c-spåret i matematik på gymnasiet och har gjort det både innan och efter implementeringen av programmering i kursplanen. Lärare kontaktades baserat på detta kriteriet. Det subjektiva urvalet har blandats med ett bekvämlighetsurval då tidigare kontakt funnits med lärare, inom kriterierna, under VFU-moment.

De intervjuade lärarna (7 st) har arbetat som matematiklärare mellan 5 och 23 år, fem av lärarna undervisade även i fysik. Sex av lärarna angav sig ha ingen eller begränsad kunskap inom programmering.

3.2. Intervjuer

En intervjuguide är en guide för intervjuaren som skapas med avseende att kunna styra förloppet av intervjun. Den kan vara mer eller mindre strukturerad och kan innehålla allt ifrån de ämnen som ska beröras till mer detaljerade frågor. För en halvstrukturerad intervju, som använts i denna studie, innehåller intervjuguiden både ämnen som ska beröras i intervjun och förslag på frågor. (Kvale & Brinkmann 2014) Intervjuguiden för denna rapport hittas i bilaga 8.1. För att få ut så mycket som möjligt av en halvstrukturerad intervju krävs att intervjuaren ställer andrafrågor, vilket kan liknas vid en följdfråga. Andrafrågor kan vara till exempel att intervjuaren ber informanten att förklara vad den menar med något den sade eller att den ska ge några exempel. För att få så nyanserade svar som möjligt från informanten krävs det aktivt lyssnande från intervjuaren så att andrafrågorna blir relevanta för intervjun. (Kvale & Brinkmann 2014)

Intervjuerna dokumenterades genom ljudinspelningar då detta ger en i princip fullständig dokumentation av intervjuerna. (Denscombe, 2017)

3.3. Analys

Enligt Björndal (2005) innebär analys att förenkla, klassificera och jämföra, kartlägga mönster samt förklara orsakerna till dessa mönster och sedan bedöma konsekvenserna av dessa mönster. Då metoden för datainsamling i denna studie är halvstrukturerade intervjuer består datan av ljudinspelningar. Dessa ljudinspelningar transkriberades för att lättare ha tillgång till datan utan att behöva lyssna om på inspelningarna. (Denscombe, 2017)

Tematisk analys innebär att datan kategoriseras och analyseras utifrån teman (Bryman, 2012). Teman valdes med avseende på hur relevant det är till undersökningens forskningsfrågor samt om något var återkommande, antingen om en lärare tog upp något flera gånger eller om flera olika lärare tog upp samma saker. Teman valdes ut baserat på ovanstående kriterier och eventuella teman såsom vilka uppgifter lärare hade arbetat med, hur lektioner kan se ut eller hur eleverna påverkats har utelämnats. Eftersom temana är starkt kopplade till forskningsfrågorna är analysen deduktiv. Analysen är semantisk då temana är identifierade från tydligt förekommande händelser i intervjuerna, temana är inte baserade på en tolkning av underliggande idéer av det som lärarna har sagt i intervjuerna. (Braun & Clarke, 2006)

3.4. Metoddiskussion

3.4.1. Etik

Undersökningen har följt de fyra huvudprinciperna som Denscombe (2017) skriver om:

- Deltagarnas intressen ska skyddas
- Deltagandet ska vara frivilligt och baserat på samtycke
- Forskare ska arbeta på ett öppet och ärligt sätt med hänsyn till undersökningen
- Forskningen ska följa den nationella lagstiftningen

Genom att anonymisera deltagarna och varken nämna var de jobbar, namn, kön eller andra personliga egenskaper skyddas deltagarnas intressen om känslig information hade kommit upp under intervjuerna. De intervjuer som har spelats in under undersökningen har endast varit tillgängliga för undersökaren. Innan intervjun informerades deltagarna om att deltagandet är frivilligt och information om undersökningen gavs. Ovanstående information om anonymisering nämns också för deltagarna. Deltagarna informerades även om vad undersökningen handlade om och vad intervjun skulle handla om. Undersökningen har följt svensk lagstiftning. Detta går i linje med Vetenskapsrådets (2017) forskningssed.

3.4.2. Reliabilitet och validitet

Om reliabiliteten är hög i en undersökning innebär det att resultatet kan förväntas vara snarlika om undersökningen görs om med samma ramar. Validitet mäter om resultatet är ett rimligt mätinstrument för undersökningens frågor. (Bryman, 2012)

Studien har använt ett kumulativt tillvägagångssätt vilket innebär att endast sju intervjuer hölls, då fler intervjuer inte förväntades ge nya resultat. Tack vare detta bör reliabiliteten vara hög då det kan förväntas att resultatet blivit snarlikt om samma intervjuer genomförts med liknande urval av lärare.

Intervjuguiden testades på en annan examensarbetare samt skickades till handledare innan den togs i bruk vilket ger en ökad validitet (Bryman, 2012). Analysen är semantisk vilket bidrar till att författaren inte behöver tolka underliggande meningar i datan och intervjuerna var halvstrukturerade vilket bidrar till att de som blivit intervjuade har kunnat svara friare än om intervjuerna hade varit strukturerade. Dessa två aspekter bidrar till ökad validitet.

4. Resultat & Analys

Rådatan transkriberades och lästes igenom flertalet gånger för att kunna formulera teman som ofta förekom i intervjuerna. Temana baserades på analys enligt avsnitt 3.3. Med hjälp av dessa teman lästes transkriberingen igenom ytterligare ett par gånger och datan delades upp enligt temana.

De teman som resulterade från analysen är:

- Kompetensutbildning för lärare
- Elevernas kunskaper inom programmering
- Tidsbrist
- Kursplaner
- Programspråken
- Digitala verktyg
- Kurslitteratur
- Förbereda eleverna för vidare studier
- Grundkurs i programmering

4.1. Kompetensutbildning för lärare

Detta tema beskriver vad för typ av utbildningar som lärarna gick innan implementeringen och hur de hade önskat att utbildningssituationen såg ut.

Under intervjuerna framkom att skolorna hanterat kompetensutbildning för lärarna inom programmering på tre olika sätt. Det första alternativet är att skolan inte hade haft någon utbildning. Lärarna som inte hade haft någon utbildning hade önskat att de hade blivit utbildade inom programmering, framför allt konkreta exempel på hur programmeringen kan användas i matematiken. Diskussioner, inför införandet av programmering i matematiken, mellan skolans lärare hade förekommit.

”Det har varit diskussioner fram och tillbaka, men inte stöd med konkreta exempel med ‘så har vi tänkt att vi ska göra’, ‘dit vill vi att eleverna ska nå’, ‘vad är målet vi ska vara’, inte riktigt.”

Andra alternativet är att skolan hade haft en lärarledd internutbildning där de lärare som hade erfarenhet av programmering lärde ut programmering samt med stöd av de matematiklärare

som hade programmeringserfarenhet tog fram konkreta exempel på hur programmering kan implementeras i matematikämnet.

”Vi hade en grupp här på skolan där vi lärde oss om programmering, åtminstone en gång i månaden. En var duktig på att programmera som höll lite kurser som helt enkelt lärde oss python.”

Ett tredje alternativ var externa utbildningar. En lärare berättade att utbildningsföretaget Academedia erbjöd sina skolor en webbkurs för lärare våren 2018. Skolverket har även dem arrangerat träffar för lärare, vilket nämndes av flera av lärarna.

”Vi har gått på Skolverkets, dem har haft några träffar, i år ingenting. Inför förändringen var det mycket prat, men nu detta läsår ingenting. Det har försvunnit.”

Gemensamt för ovanstående utbildningar är att de utfördes inför höstterminen 2018 eller under läsåret 2018/2019. Lärarna uttryckte att inga ytterligare utbildningar genomförts efter det första läsåret med programmering i matematikundervisningen. Det enda som kvarstår är diskussioner mellan lärare på ämnesmöten eller liknande.

Mer utbildning, men framför allt konkreta exempel på hur programmeringen kan användas i matematiken, önskas av lärarna. De menar att eftersom målen från Skolverket inte är tydliga är det viktigt att lärarna får en plattform där de kan få inspiration, stöd och möjlighet att utbyta erfarenheter. Diskussionerna online har också minskat sedan införandet, och en plattform för diskussioner är något som lärarna skulle vilja ha.

”En sådan utbildning skulle först och främst vara väldigt konkret. Presentera olika program. Presentera olika mål och hur olika metoder för att göra programmering.”

4.2. Elevernas kunskaper inom programmering

Detta tema behandlar elevernas kunskaper inom programmering. Deras kunskaper var i många fall bristfälliga vilket lärarna såg som ett stort problem.

Elever som läser naturvetenskapsprogrammet läser idag ingen obligatorisk programmering på gymnasiet förutom i matematikkurserna och elever på teknikprogrammet läser inte alltid programmering direkt vid gymnasiestart. Detta ser lärarna som en stor brist av flera anledningar. Elevernas bristande kunskaper inom programmering innebär att stora delar av matematiklektionerna går till att lära ut själva programmeringen, istället för att ägna sig åt matematikinnehållet. Det kan exempelvis handla om hur syntax fungerar.

”80-85% av tiden går till syntaxen, bara var ska kommatecknet sitta, var ska det vara mellanslag, var ska det vara indraget. Det är där det blir fel, där alla fastnar. Så att ibland så blir det som att, aha den här lektionen gick inte till någon matte egentligen, utan bara att försöka hantera programmet.”

En lärare säger att det är önskvärt att nå till en punkt där eleverna får en uppgift och själva kan komma fram till att den kan lösas med programmering. Då behövs inte hela lektioner med programmering, vilket är det som krävs idag. Det vill säga att eleverna ser programmering som ett digitalt verktyg som de kan plocka fram när de anser lämpligt.

”Om idén är att använda programmering för att lösa problem, så bygger det på att man kan programmera eller att man har problem som löses lätt med programmering. Elevernas bristfälliga kunskaper i programmering innebär att de måste lära sig mycket programmering innan det blir meningsfullt.”

Eleverna ska ha med sig kunskaper inom programmering från grundskolan, men så är inte fallet i dagsläget då eleverna som nu går på gymnasiet har läst max ett år på grundskola efter implementeringen av programmering.

“Ideen från skolverket är ju att eleverna ska ha kunskaper om programmering från grundskolan för att sedan kunna tillämpas på gymnasiet. Men eftersom det är långt från ett faktum så måste jag lära dom det. Att få en enhetlig grund från grundskolan är svårt eftersom de kommer från olika skolor, och det är inte ens tydliggjort vad det är egentligen man ska kunna från grundskolan.”

4.3. Tidsbrist

Temat tidsbrist beskriver att lärarna tycker att programmering i matematiken har inneburit att det blir mindre tid till matematiken och att planeringen tar längre tid.

Några av lärarna beskriver att planeringen för lektionstillfällena där programmering ska ingå tar lång tid på grund av deras bristande förkunskaper i programmeringen. Innan de kan planera för att använda programmering i undervisningen måste de själva lära sig språket.

”Ska du då lära dig python tar ju det tid, att lära dig programspråket, jag kan ju inte själv det något bra.”

“Jag skulle helst vilja slippa att undervisa om programmering, men ibland är det nödvändigt. “

Lärarna säger att matematikkurserna var sprängfyllda redan innan implementeringen av programmering. Eftersom ingenting togs bort när programmering lades till är kurserna nu ännu mer fyllda. Några av lärarna säger att det måste prioriteras att hinna med alla moment innan de nationella proven då de nationella proven ska väga extra tunga inför betygsättning efter lgy11. I och med att programmering inte i dagsläget ingår i de nationella proven så tycker lärarna att det är svårt att motivera att ta tid till programmering på lektionerna.

”Nationella ska ju väga extra tungt efter nya riktlinjerna, så därför vill man förbereda dem[eleverna] så mycket som möjligt inför de nationella proven, där programmering inte ingår som det sett ut hittills.”

”Det[attn undervisningen inte har förändrats mycket] är främst för tidsbrist, svårt att hinna med eftersom inget plockades bort. Lades bara till i kurser som redan nu är svåra att hinna.”

”Jag kan tycka att det är så pass avancerat att lära sig verktyget, det är liksom ett helt språk, att det låter sig inte göra inom ramarna av mattetimmarna.”

4.4. Kursplaner

Detta tema behandlar lärarnas tankar om skolverkets kursplaner inom matematik och oklarheter i dessa kursplaner.

De berörda kursplanerna för matematik på gymnasiet nämner som det ser ut idag programmering på ett ställe under rubriken problemlösning. Lärarna tycker att skolverkets riktlinjer är alldeles för svaga och icke-konkreta. De vill se ett tydligt mål med programmeringen med vad det är eleverna förväntas kunna inom programmering. De skulle även vilja se vilken slags programmering som ska läras ut och hur den förväntas läras ut.

”Ge mig tydligare vad det är jag förväntas lära ut till eleverna och vad de förväntas kunna, och hur mycket tid jag får för att göra detta. Får jag ingen tid för detta så är det inte seriöst, då kan vi lika gärna låta bli, det är precis vad som har hänt nu. Man tydliggör att det här ska göras, men man ger ingen tid. Det är verkligen bara mer, inget togs bort.”

Om programmering ska finnas kvar som det ser ut i dagsläget så vill lärarna att kurserna ska utökas så att programmering får en naturlig plats och inte behöver klämmas in.

”Eller så får man utöka kurstiderna för mattekurserna. Det blir princip en kurs i kursen, annars ser jag inte hur man ska packa in ytterligare ett moment i en kurs som redan nu är tajt tidsmässigt. Så jag tror inte att man kan trycka in, det är önsketänkande. Att bara säga ”nu kör vi programmering i matten”. Vad vi ska ta bort då är frågan faktiskt. Jag tror utökade kurser på matten isåfall.”

4.5. Förbereda eleverna för vidare studier

Detta tema behandlar sådant i intervjuerna där studenternas vidare studier diskuterades samt hur programmering i matematiken relaterar till detta.

I de intervjuer där fortsatta studier kommer upp i samtalet ser lärarna det som positivt att eleverna på naturvetenskapliga- och teknikprogrammet, som båda är högskoleförberedande, lär sig programmering. Många av eleverna ska efter gymnasiet vidare till tekniska utbildningar på universitet eller högskolor där programmering ingår i de flesta fall.

“Jag förstår tanken, när de går vidare från gymnasiet ska de ha en viss tanke vad programmering ser ut eftersom de ska programmera på universitetet också. Tanken är att de ska komma dit, de som ska läsa till ingenjörer måste definitivt kunna lära sig programmering.”

4.6. Programspråken

Detta tema behandlar lärarnas tankar om programmeringsspråk samt hur detta relaterar till elevernas eventuella kunskaper inom andra programmeringsspråk.

De lärare som pratar om vilket programmeringsspråk de har använt säger alla att de använder sig av Python. En lärare, som inte hade använt ren kodning i undervisningen, tyckte att de programmeringsspråk som finns idag inte är anpassade till matematiken och att de programspråk som är speciellt anpassade för matematik, matlab eller mathematica, är dyra vilket skolan inte har råd med.

“Det finns några programmeringsspråk ute, men jag tycker att det inte är riktigt anpassade till matematiken än. Till exempel matlab eller mathematica skulle kosta enorma pengar att köpa in som skolan inte har råd med.”

Vad gäller teknikelever som redan har programmerat uttryckte sig lärarna lite olika. En lärare sa att programmeringslärarna på skolan inte ville att eleverna skulle lära sig ett nytt språk på matematiken då eleverna skulle bli förvirrade vilken kod som hör till vad(då olika program har olika syntax).

“Vi har haft samtalet mellan matematiklärare och programmeringslärare vad dom tycker och säger. De vill helst inte att vi ska lära ut ytterligare ett språk för då kommer eleverna bli helt förvirrad vilken kod hör till vilket program. Men däremot kan vi ha ett samarbete med programmeringen.”

Några av lärarna beskrev att teknikeleverna programmerar på en högre nivå än det som lärs ut i matematiken eller på en högre nivå än matematikläraren, och att detta sågs som ett problem då dessa elever inte skulle uppskatta att programmera på matematiken då nivån skulle vara för låg.

“Jag tror att om jag skulle sätta igång att lära dem [teknikeleverna] något tror jag de skulle tycka det var löjligt då de redan kan mer avancerat än det jag lär ut.”

“Okej, dom som läser programmering på gymnasiet, dom kanske inte behöver den här programmeringen, det är för lätt nivå. De jobbar på väldigt hög nivå på programmering och det handlar om att vi slösar bort tid.”

En annan lärare sa att hen undervisar i python på matematiken men att eleverna annars kodar i java, vilket hen inte ansåg vara ett problem.

“Väldigt begränsat [erfarenhet inom programmering] från grundskolan, och även om dom har programmerat tidigare så kanske det inte är i python. Men däremot så läser mina nuvarande elever programmering, men då inte i python, utan java tror jag.”

4.7. Digitala verktyg

Detta tema behandlar andra typer av digitala verktyg än programmering. Digitala verktyg används mycket i matematikundervisningen och det finns många olika program att använda sig av.

Lärarna visar generellt upp en positiv attityd mot digitala verktyg och detta är något som alla tillfrågade lärare använder. Geogebra nämns mest, men även Excel, Google Sheets, Desmos och Pear deck används flitigt.

“Men däremot tror jag att eleverna kommer ha lättare att tillgodogöra sig matten tack vare programmeringen. Jag skulle gärna se att det finns simuleringar man kan göra, prövar vad som händer om man ändrar konstanter, prövar algoritmer och konvergens. Lite såna saker för att förstå teorin, men det räcker inte med programmering för att förstå teorin. Så jag ser mer programmeringen som ett hjälpmedel.”

Fördelarna med digitala verktyg som belyses är att geogebra hjälper eleverna att göra det abstrakta konkret genom att de kan rita ut kurvor och grafer. Genom att ändra variablerna kan eleverna se hur grafer förändras istället för att arbeta med grafer abstrakt.

“Om de löser bara uppgifter har de inte en graf framför sig. Det blir lite svårt och abstrakt annars. Här [med geogebra] blir det konkret, så här kommer grafen se ut om du tar bort c eller b [vid undersökning av en andragradsekvation].”

En skillnad mot programmering är att startsträckan inte är lika lång då dessa program är mer intuitiva än ett rent programmeringsspråk och eleverna inte behöver lika mycket grunder från tidigare. Med bättre grunder hos eleverna hoppas lärarna att i framtiden kunna använda programmering som ett verktyg som eleverna kan använda precis som de använder geogebra eller andra digitala verktyg idag.

“De [Geogebra & Excel] funkar bra, de är mer intuitiva, man kan starta på en gång. Inte så lång startsträcka.”

Några nämner att de inte vill använda programmering när i de fall då det finns bättre alternativ. Programmeringen vill de använda för de uppgifter som är relevanta och om ett program som geogebra eller desmos är mer lämpligt vill de använda dem.

“Sen att hitta uppgifter som känns meningsfulla, minst lika jobbigt. Det måste vara uppgifter där programmering känns relevant. Till exempel kan vi lösa $2+5$ med programmering, men varför ska vi göra det när vi kan lösa det på miniräknaren? [...] De övningar jag söker ska inte kunna göras utan digitala hjälpmedel, varken räknaren, geogebra, excel eller desmos ska vara lämpliga istället eller mer lämpliga.”

4.8. Grundkurs i programmering

Detta tema behandlar lärarnas tankar om en grundkurs i programmering då de anser att detta är något som eleverna skulle behöva.

Flera av lärarna nämner att det skulle behövas en grundkurs i programmering på gymnasiet, antingen 50p eller 100p. I den skulle eleverna lära sig grunderna i programmering och eventuellt program som google sheets. Kursen datorkunskap togs bort i och med lgy11 vilket vissa av lärarna är tveksamma till då elevernas generella datorkunskaper är låga jämfört med för 10 år sedan eftersom nuförtiden så använder dagens gymnasieelever mer plattor och smartphones jämfört med datorer.

“Det hade underlättat om de[naturelever] gjorde det [läst programmering], skulle jag säga. Och jag tror att det skulle gå åt mindre tid för mig som mattelärare om jag inte behövde ta det grundläggande i programmeringen, man skulle komma till skott fortare. En grundkurs i ettan så vi kan ha samma grund på hela skolan. Vi kan styra det mer.”

En kurs för alla elever inom programmering deras första termin på gymnasiet skulle innebära att lärarna vet vilken nivå eleverna förväntas ligga på inom programmering och att skolan kan styra och skapa en gemensam grund för t.ex vilket programmeringsspråk som ska användas på skolan. Programmering kan även appliceras på flera ämnen än matematik, t.ex fysik.

“Jag tycker det[programmering] ska vara en egen kurs, inte baka in i matematiken. Den är redan sprängfylld. En ren programmeringskurs så att alla får samma grunder, det kan appliceras i flera olika ämnen t.ex fysiken.”

“Skulle min naturklass få gå programmering 1 i ettan så har man ett helt annat utgångsläge, och det är kanske det som behövs för att de ska få lite grunder och att man ska kunna göra något mer meningsfullt.”

5. Diskussion

I detta kapitel kommer denna studies resultat diskuteras i relation till den tidigare forskning som presenterades i kapitel två. Resultatet kommer även diskuteras i relation till Finlands och Englands implementeringsstrategier.

Diskussionen utgår ifrån de två forskningsfrågorna:

Vilka möjligheter och utmaningar har gymnasielärare i matematikämnet upplevt sedan introduktionen av programmering för problemlösning på programmen naturvetenskapligt- och tekniskt program?

Vilka strategier för implementering och undervisning av programmering i matematikämnet har använts av lärarna?

Den första forskningsfrågan kopplas till delkapitlet 5.1 *Utmaningar* samt 5.2 *Möjligheter*. Den andra forskningsfrågan kopplas till delkapitlet 5.3 *Strategier*.

Tabell 3. Hur temana kopplas till nedanstående diskussion

Tema	Koppling till diskussionsdel
Kompetensutbildning för lärare	Strategier
Elevernas kunskaper inom programmering	Utmaningar
Tidsbrist	Utmaningar
Kursplaner	Utmaningar
Förbereda eleverna för vidare studier	Möjligheter
Programspråken	Utmaningar & möjligheter
Digitala verktyg	Möjligheter
Grundkurs i programmering	Möjligheter & strategier

5.1. Utmaningar

5.1.1. Kunskaper

Implementeringen av programmering i skolans kursplaner skedde samtidigt i alla årskurser vilket innebär att under dessa första år kommer eleverna börja gymnasiet med bristande eller inga kunskaper inom programmering. Detta eftersom programmering inte ingick i grundskolans kursplaner tidigare. I temat *elevernas kunskaper inom programmering* blir det tydligt att denna brist i programmeringskunskaper har inneburit att lärarna behövt lära ut grunderna inom programmering under matematiklektionerna för att kunna använda programmering på lektionerna.

Citatet från Heintz et al. (2017) i 2.1.6 sammanfattar detta problem:

“The progression in the revised curriculum is expressed as if the curriculum has already reached a steady state. This leaves local authorities throughout the country with the responsibility of implementing the changes and solving the bootstrapping challenge.”

Att förändra i kursplanen samtidigt i alla årskurser och formulera det som att eleverna behöver ha med sig kunskaper från tidigare årskurser innebär just att kunskaperna kommer vara haltande några år framöver. Exempelvis så kommer de elever som börjar gymnasiet hösten 2020 endast ha gått två år på grundskolan sedan kursplanerna uppdaterades och det är tveksamt om deras kunskaper inom programmering är tillräckliga för att lösa problem med hjälp av programmeringen på matematiklektionerna.

Något som nybörjare inom programmering har svårt med är, enligt Qian & Lehman (2017), syntax. Detta är även något som visas i tema *elevernas kunskaper inom programmering* då lärarna beskriver att stora delar av lektionerna ibland behöver gå till att lära ut syntaxen och sedan hjälpa eleverna när syntaxen blir fel. Malik & Nelson (2017) beskriver att en person måste lära sig både ytkunskaper och djupare kunskap för att kunna lösa problem med hjälp av programmering. Eleverna måste även lära sig att förstå helheten i ett program samt hur olika metoder och kodrader relaterar till varandra (Teague & Lister, 2014).

I början av inläring av programmering ligger mycket fokus på att lära sig ytkunskaperna men för att eleverna ska kunna lösa problem med programmering som ett verktyg behöver de även djupare kunskaper om programmering. Om eleverna då saknar grunder i programmering när den ska användas i matematikämnet kommer alltså problemlösning vara svårt att uppnå. Några av lärarna sa att målet är att nå till en punkt där eleverna kan använda programmering som ett verktyg för att lära lösa problem i matematiken. De såg de bristande kunskaperna hos eleverna som ett stort hinder och om de då istället hade haft elever som redan kunde själva programmeringen hade det underlättat rejält.

I motsats till att eleverna har bristande kunskaper så visar tema *programspråken* att flera av de intervjuade lärarna tycker att det är onödigt att de skulle lära ut programmering till teknikelever

som redan läser programmeringskurser eller att dessa elever inte behövde programmeringen i matematiken. Att se det som ett problem och inte en möjlighet att eleverna redan kan programmering är synd. Om eleverna redan kan programmera behöver inte läraren lära ut programmering till dem utan programmeringen kan då direkt börja användas som ett verktyg för problemlösning och då uppfylla programmerings syfte enligt kursplanen. Om läraren är bekväm med ett språk och eleverna är bekväma med ett annat språk så kan läraren låta eleverna använda det språk de vill eftersom fokuset inte bör ligga på själva programmeringen utan fokus bör ligga på matematiken och problemet som ska lösas. Själva programmeringen kanske ligger på en lägre nivå än vad eleverna är vana vid, men då programmeringen inte är fokuset under matematiklektionerna bör detta inte vara ett problem.

Programmering är i dagsläget inte obligatoriskt för varken natur eller teknik. Generellt så läser dock teknikprogrammet oftare programmering än natur. Som visas i tema *grundkurs i programmering* efterfrågar lärarna en grundkurs i programmering för alla elever som ska programmera i matematiken. Eftersom det finns en kurs (programmering 1) så behövs inte en ny kurs införas, men det krävs en förändring i programplanerna för natur och teknik där programmering 1 läggs in som obligatorisk. I programmering 1 lär sig eleverna grunderna i programmering och om denna kurs skulle införas som obligatorisk för alla som läser c-spåret i matematik skulle det underlätta för matematiklärarna avsevärt. Att alla elever skulle läsa en kurs i programmering är i likhet med hur England har gjort där alla elever läser ämnet "Computing" (Heintz et al, 2017; The royal society, 2017)

5.1.2. Tid

Om matematiklärarna behöver lära ut grunderna inom programmering för att kunna använda det till problemlösning går det åt ytterligare planeringstid och lektionstid. Denna tidsbrist är en av de stora utmaningarna som nämns i resultatet och som också beskrivs som ett problem av Humble et al. (2019) och Ekberg & Gao (2018). Som visas i temat *tidsbrist* tycker lärarna att kurserna redan innan implementeringen hade mycket innehåll som skulle hinnas med och att då lägga till programmering utan att ta bort något gör att kursernas innehåll blir svårare att hinna med. Om elevernas kunskaper är bristfälliga så att programmering måste läras ut under lektionstid i matematikämnet så tycker lärarna att kurserna behöver utökas tidsmässigt. Något som skolorna hade kunnat göra dessa första år efter förändringen på kursplanen är att tid skulle avsatts till att lärarna fick lära ut programmering till eleverna och tid för att planera för lektionerna där programmering skulle ingå.

För att en lärare ska kunna lära ut programmering så behöver läraren specifik utbildning och motivationen att lära sig om dagens snabbt fortskridande teknik. Om läraren inte har denna utbildning så kan det påverka elevernas lärande negativt. (Mannila et al, 2014. Gal-Ezer & Hazzan, 2010. Dagiene & Stupurienė, 2016) Dessa utbildningar tar även de tid och det kan vara så att utbildningarna ligger utanför arbetstid vilket problematiserar det hela.

Nationella proven är något som lyftes upp under intervjuerna. I dagsläget så ingår inte programmering i nationella provet, vilket innebär att programmering ibland kan prioriteras bort.

Detta eftersom nationella provet ska väga tyngre sedan lgy11 och lärarna vill fokusera på att lära ut det som eleverna kommer ha nytta av för att klara de nationella proven. Det finns en tydlig motsättning i lärarnas vilja att lära ut allt i kursplanen och att fokusera på de nationella provens innehåll. Om programmering skulle ingå i de nationella proven skulle den här motsättningen minskas eller inte finnas alls.

5.1.3. Kursplaner

I kursplanen för matematik på gymnasiet står det att programmering ska användas för problemlösning. Det står dock inte på vilket sätt eller hur mycket. Som visas i tema *kursplaner* efterfrågar lärarna tydligare riktlinjer om vilken typ av programmering, vilket språk, hur mycket och vad målet är med programmeringen. Heintz et al. (2017) skriver att otydliga kursplaner kan vara bra i början då detta ger mer frihet för läraren. Detta är något som de intervjuade lärarna i denna studie inte håller med om då resultatet visar att lärarna tycker att kursplanen har för svaga riktlinjer och är icke-konkret.

Om man jämför vad det står om programmering jämfört med vad det står om andra digitala verktyg i kursplanen så står det inte detaljerat hur varken programmering eller andra digitala verktyg ska användas i undervisningen. Trots det så har lärarna använt digitala verktyg som Geogebra och Desmos redan innan implementeringen. Med detta i åtanke så kanske inte kursplanen är den plats där användandet av verktygen ska beskrivas på ett detaljerat sätt. En stor skillnad mellan program som Geogebra och ren programkod är att Geogebra är mer intuitiv och användarvänligt, vilket lärarna beskriver som att startsträckan är mycket kortare. Detta tillsammans med att programmering är någonting nytt för lärarna och eleverna är det som bidrar till att tydligare riktlinjer efterfrågas. Så istället för att uppdatera kursplanerna att ha tydligare riktlinjer så är nog det som behövs konkreta exempel på hur programmeringen kan användas i undervisningen så att lärarna kan bygga upp en likadan grund som de idag har i Geogebra.

Denna upplevda otydlighet i kursplaner hos lärarna i denna studie samt att de inte tidigare är utbildade i programmering tyder på en osäkerhet i ämnet. Eftersom utbildningarna som lärarna har gått inför implementeringen har använt python så verkar det finnas en tendens hos lärarna att automatiskt välja detta programmeringsspråk. Något som vore bra är att lärarna fick underlag för flera olika typer av programmering så att läraren kan göra val baserat på den elevgrupp som undervisas. Genom att ha en bredare förståelse för programmering som begrepp kan vissa av de belysta problemen underlättas. Istället för att fokusera mycket på koden kan fokus skifta till själva arbetssättet och programmeringens strategier. Exempelvis kan det ha varit bättre att använda scratch eller excel för att lösa problem inom matematiken då dessa inte kräver att eleverna skriver kod, speciellt nu i början av implementeringen då elevernas förkunskaper är bristfälliga.

5.2. Möjligheter

Ekberg & Gao (2018) skriver att lärarna i deras studie har en backup-plan ifall tekniken strular, eftersom tilltron till tekniken inte finns hos dessa lärare. Speciellt de äldre lärarna är mindre benägna att använda teknik i sin undervisning. Tvärtemot detta visar resultatet i denna studie att även om lärarna ser programmering som problematiskt moment så har de en generell positiv syn på teknik och digitala verktyg och gärna använder dessa i sin undervisning. Program som Geogebra och Desmos används och har använts flitigt även innan implementeringen av programmering. En stor fördel med program som dessa är att de kan göra matematiken mindre abstrakt då eleverna till exempel kan rita upp grafer och genom att ändra koefficienter i en funktion se hur grafen förändras.

Programmering ger eleverna möjligheter att lösa problem som är svåra att lösa på andra sätt. Exempelvis uppgiften att hitta de 1000 första primtalen tar lång tid att lösa för hand och är rätt meningslös, medan det är en ypperlig uppgift för en elev att lösa med hjälp av programmering. Detta exempel är hämtat ur en av intervjuerna och läraren tryckte då extra på att det krävs relevanta uppgifter där programmering känns som lämpligt och optimalt verktyg för att lösa problemet. Ökade krav på komplex problemlösning är något som programmeringen har möjlighet att lösa. Psycharis och Kallia (2017) skriver att programmering bidrar till flera positiva aspekter för eleverna, där problemlösning är en av dessa.

Det finns mycket material om programmering online (Humble et al., 2019). Lärarna efterfrågar dock mer konkreta exempel på hur programmering kan användas i matematiken samt ett forum där diskussioner med andra lärare kan hållas. Eftersom implementeringen skedde nyligen så kan det dröja innan det finns mycket material online om programmering i matematiken men ett diskussionsforum har potentialen att snabbare förmedla mer vardaglig inspiration.

Att eleverna som läser natur och teknik på gymnasiet får möjlighet att använda programmering i matematiken är något som kommer hjälpa dem om de väljer att studera vidare på en teknisk högskola då programmering ingår i nästan alla civilingenjörsprogram. I temat *förbereda eleverna för vidare studier* beskriver lärarna det positiva med att förbereda eleverna för kommande studier. Programmering och problemlösning är en central egenskap hos ingenjörer (CDIO, 2011), vilket tyder på att lärarnas vision om att förbereda sina elever är välgrundad.

5.3. Strategier

I Finland så skedde implementeringen stegvis vilket gjorde att kunskapsglappet inte finns på samma sätt som det finns här i Sverige (Skolvärlden, 2015). Att implementera stegvis innebär att endast de elever som hade lärt sig programmering från årskurserna under har programmering i sina kursplaner och de elever som inte har läst programmering i årskurserna under går vidare med sin gamla kursplan. Med Finlands implementeringsstrategi förhindrar problemet som visas i tema *elevernas kunskaper inom programmering* där lärarna i denna studie upplever att eleverna inte har med sig grundkunskaper i programmering från grundskolan.

I England har de ett separat ämne där programmering och datorkunskaper ska läras ut. Detta är likt ämnet "datorkunskap" som fanns på gymnasiet innan lgy11. Att ha en kurs på gymnasiet där programmering lärs ut nämns som önskvärt av de lärarna i denna studie eftersom eleverna då skulle få en grund i programmering så att de både kan mer och att matematiklektionerna kan fokusera på att lära sig matematik istället för att lektionstid ska gå åt att lära sig programmering.

Skolverket har erbjudit utbildning och erbjuder fortfarande kurser i tre steg. Under intervjuerna nämndes det att skolverket hade anordnat träffar, men dessa utbildningar nämns ej. Lärarna tyckte att det är otydligt vad som förväntas med användandet av programmering i matematiken och något som hade behövts inför implementeringen är att skolverket skulle haft tydligare riktlinjer som eventuellt hade förmedlats på deras utbildningar.

Skolorna som lärarna i studien arbetar på hade valt att lägga upp sina utbildningar inom programmering på olika sätt. Lärare som hade erfarenhet inom programmering är en stor resurs som användes som utbildare på några av de skolorna där de intervjuade lärarna arbetar på. Oavsett vilken typ av programmeringsutbildning som lärarna i denna studie hade fått så kände de att kompetensutbildningar med mer konkreta exempel på hur de som lärare kan använda programmeringen i matematiken var något som efterfrågades. Om utbildning hade skett så skedde den inför eller under läsåret 2018/2019 och har detta läsår upphört. Med ökade krav på digital kompetens hos lärarna samt ett samhälle som blir mer och mer digitaliserat behöver lärare ha tillgång till en kompetensutveckling som de efterfrågar. Idag är så inte fallet och för att skolan ska kunna utvecklas i samma takt som digitaliseringen krävs det att kompetensutveckling finns och är på hög nivå för skolans lärare.

6. Slutsats

Vilka möjligheter och utmaningar har gymnasielärare i matematikämnet upplevt sedan introduktionen av programmering på programmen naturvetenskapligt- och tekniskt program?

Lärarna i denna studie har upplevt mer utmaningar än möjligheter med införandet av programmering i matematiken. Det största problemet är att eleverna inte har grundkunskaper i programmering med sig från grundskolan vilket i sin tur innebär att lärarna måste lägga ner planerings- och lektionstid på att lära ut programmering till eleverna.

Lärarna i denna studie ser generellt digitala verktyg som någonting bra, men de verktyg som de använder mest (till exempel Geogebra) har kortare startsträcka än ren kodning vilket gör att de inte behöver lägga massa lektionstid på att lära eleverna verktyget.

Vilka strategier för implementering och undervisning av programmering i matematikämnet har använts?

Lärarna i denna studie har fått olika typer av utbildningar inför implementeringen av programmering i matematiken. Något som lärarna har saknat är konkreta exempel på hur programmeringen ska användas i matematikämnet. Utbildningarna har haft fokus på att lära ut själva språket istället för att lära ut arbetssätt och strategier inom programmering. Att utbildningarna har lärt ut python har gjort att python är det språk som lärarna har valt att använda även om det finns andra språk som har kortare startsträckor.

Sverige har genomfört förändringen i kursplanen annorlunda jämfört med de två andra länder som har jämförts i denna studie. England har lagt till ett ämne, "Computing", som är likt ämnet datorkunskap som fanns på svenska gymnasiet innan lgy11. Finland har gjort en stegvis implementering vilket betyder att implementeringen skett en årskurs i taget. Detta har inneburit att eleverna har grundkunskaper i programmering med sig till äldre årskurser och de som redan går i högre årskurser förväntas inte kunna programmering utan de fortsätter på sina ursprungliga kursplaner.

Som det ser ut nu så väger utmaningarna tyngre än möjligheterna med programmering i matematiken. För att programmering i matematiken ska fungera krävs bättre kunskaper hos eleverna. Detta kan eventuellt komma att lösas automatiskt då eleverna kommer ha med sig mer och mer kunskaper från grundskolan över de kommande åren. Om problemet inte löser sig automatiskt behövs antingen programmering 1 göras obligatorisk för alla som läser c-spåret i matematik eller att matematikkurserna där programmering ska ingå utökas. Elevernas förkunskaper måste även tas i åtanke när programmering i matematiken planeras då t.ex Python inte alltid är det bästa språket att använda.

Förberedelserna fokuserade på att lärarna skulle lära sig programmera istället för att lära sig strategier för problemlösning med hjälp av programmering. Lärarna måste få fortsatt kompetensutbildning för att kunna inkludera programmering i matematikämnet och kunna göra olika val för hur programmeringen ska användas. Om det rör sig om en elevgrupp som inte har förkunskaper inom programmering ska läraren ha fått utbildning inom t.ex. scratch eller excel för att kunna välja det som programmeringsverktyg medan en elevgrupp som är erfarna inom programmering är mer lämpade att skriva kod. Oavsett vilket språk som används så för programmeringen med sig möjligheter att lösa uppgifter som annars vore olämpliga att lösa och det förbereder eleverna för framtida tekniska studier.

Baserat på ovanstående är mina rekommendationer att skolverket bör ta fram ett bredare utbildningsmaterial för lärare där programmeringsspråk av olika nivåer inkluderas samt att det följs upp hur elevernas förkunskaper utvecklas de kommande åren. Lärarna måste bli bekväma i olika former av programmering för att kunna göra val i sin undervisning baserat på vilken elevgrupp som undervisas. En bredare begreppsförståelse skulle hjälpa lärarna och eleverna att se programmering som ett verktyg istället för ett delmoment i kurserna i matematik.

Som avslutning kommer här några frågor som kan hjälpa lärare inför start av en matematikkurs i c-spåret:

- Kan eleverna programmering sedan tidigare eller behöver du som lärare lära ut programmering för att kunna använda det för problemlösning?
- Har eleverna programmerat i matematiken tidigare?
- Vilka programmeringsspråk är du och eleverna bekväma i?
- Vilka delar av kursen kan programmering användas i?
- Vad för typ av uppgifter kan användas i denna kurs?
- Ska hela lektioner tillägnas åt programmering eller ska programmering förekomma i kortare segment?

7. Referenser

Åbo. (n.d). Läroplan.

<https://www.turku.fi/sv/dagvard-och-utbildning/grundskolan-1-9/undervisning/laroplan>

Belgheis, S & Kamalludeen, R. (2018). The Intention to Use GeoGebra in the Teaching of Mathematics Among Malaysian Teachers. Malaysian Online Journal of Educational Technology, 6(1), 109-115.

Bjørndal, C., & Nilsson, B. (2005). Det värderande ögat : Observation, utvärdering och utveckling i undervisning och handledning (1. uppl.. ed.). Stockholm: Liber.

Braun, V., & Clarke, V. (2006). Using thematic analysis in psychology. Qualitative Research in Psychology, 3(2), 77-101.

Bryman, A. (2012). Social research methods (4.th ed.). Oxford: Oxford University Press.

Calder, N. (2019). Using Scratch to Facilitate Mathematical Thinking. Waikato Journal of Education, 23(2), 43-58.

CDIO. (2011). The CDIO Syllabus v2.0. http://cdio.org/files/project/file/cdio_syllabus_v2.pdf

Dagiene, V. Stupurienė, G. (2016). Informatics Concepts and Computational Thinking in K-12 Education: A Lithuanian Perspective

Denscombe, M. (2016). Forskningshandboken : För småskaliga forskningsprojekt inom samhällsvetenskaperna (3., rev. och uppdaterade uppl.. ed.). Lund: Studentlitteratur.

Digitaliseringskommissionen. (2014). En digital agenda i människans tjänst – en ljusnande framtid kan bli vår.

Digitaliseringsrådet. (n.d). Globala målen.

<https://digitaliseringsradet.se/vaerldens-digitala-agenda/globala-maalen/>

Ekberg, S., & Gao, S. (2018). Understanding challenges of using ICT in secondary schools in Sweden from teachers' perspective. The International Journal of Information and Learning Technology, 35(1), 43-55.

Götling, S., & Löfwenhamn, O. (2018). Programmering i matematikundervisningen: En fallstudie om utmaningar och styrkor med att programmering ska integreras i matematikundervisningen på gymnasiet.

Hansson, Å. (2019). Problemlösning i matematik.

<https://larportalen.skolverket.se/LarportalenAPI/api-v2/document/path/larportalen/material/inrikt>

ningar/1-matematik/Grunds%C3%A4rskola/461_didaktiskaperspektivpamatematikundervisning
en2_SAR/5_problemlösning/material/flikmeny/tabA/Artiklar/SK2_05A_01_problemlösning.docx

Heintz, F, Mannila, L, Nordén, L-Å, Parnes, P, & Regnell, B. (2017). Introducing Programming and Digital Competence in Swedish K-9 Education. *Informatics In Schools*, 10696 LNCS, 117-128.

Högfeldt, Anna-Karin. Kann Viggo. (2012). CDIO och ingenjörskräfverigheter.
<https://www.kth.se/social/files/5702de51f276542120e47fa1/cdio-fardigheter.pdf>

Humble, N, Mozellus, P, & Sällvin, L. (2019). Teacher Challenges And Choice Of Programming Tools For Teaching K-12 Technology And Mathematics. *Education And New Developments (End2019)*, 431-435.

KTH. (n.d). Hitta din utbildning på KTH.
<https://www.kth.se/utbildning/hitta?Utbildningar=Civilingenjoer>

Kvale, S., & Brinkmann, S. (2014). Den kvalitativa forskningsintervjun (3. [rev.] uppl.. ed.). Lund: Studentlitteratur.

Liang, S. (2016). Teaching the concept of limit by using conceptual conflict strategy and Desmos graphing calculator. *International Journal of Research in Education and Science (IJRES)*, 2(1), 35-48.

Malik, S., & Coldwell-Neilson, I. (2017). A model for teaching an introductory programming course using ADRI. *Education and Information Technologies*, 22(3), 1089-1120.

Mannila, L. Dagiene, V. Demo, B. Grgurina, N. Mirolo, C. Rolandsson, L. Settle, Amber . (2014). Computational Thinking in K-9 Education.

Mannila, L. Nordén, L-Å. Pears, A. (2018). Digital Competence, Teacher Self-Efficacy and Training Needs

Palumbo, D. B. (1990). Programming language - problem-solving research. *Review of Educational Research*, 60(1), 65-89.

Pears, A. Seidman, S. Malmi, L. Mannila, L. Adams, E. Bennedsen, J. Devlin, M. Paterson, J. (2007). A Survey of Literature on the Teaching of Introductory Programming. *ACM SIGCSE Bulletin*. 39. 10.1145/1345375.1345441.

Psycharis, S., & Kallia, M. (2017). The effects of computer programming on high school students' reasoning skills and mathematical self-efficacy and problem solving. *Instructional Science*, 45(5), 583-602.

Qian, Y., & Lehman, J. (2017). Students' Misconceptions and Other Difficulties in Introductory Programming: A Literature Review. *ACM Transactions on Computing Education (TOCE)*,

18(1), 1-24.

Ragonis, N. Gal-Ezer, J. Hazzan, O. (2010). A survey of computer science teacher preparation programs in Israel tells us: computer science deserves a designated high school teacher preparation!

Regeringen. (2012). Digitaliseringskommissionen – en kommission för den digitala agendan. <https://www.regeringen.se/49bba8/contentassets/538820854ece4bf2842e139f84b4b723/digitaliseringskommissionen---en-kommission-for-den-digitala-agendan-dir.-201261>

Regeringskansliet. (2017). Stärkt digital kompetens i skolans styrdokument. <https://www.regeringen.se/pressmeddelanden/2017/03/starkt-digital-kompetens-i-laroplaner-och-kursplaner/>

Robins, A. Rountree, J. Rountree, N. (2003). Learning and Teaching Programming: A Review and Discussion. *Computer Science Education*, 13(2), 137-172.

Sangwin, C., & O'Toole, C. (2017). Computer programming in the UK undergraduate mathematics curriculum. *International Journal of Mathematical Education in Science and Technology*, 48(8), 1133-1152.

Skolvärlden. (2015). Den finska skolan satsar digitalt. <https://skolvarlden.se/artiklar/den-finska-skolan-satsar-digitalt>

Skolverket. (2011). Läroplan för gymnasieskolan 2011 , examensmål och gymnasiegemensamma ämnen.

Skolverket. (2016). Redovisning av uppdraget om att föreslå nationella it-strategier för skolväsendet – förändringar i läroplaner, kursplaner, ämnesplaner och examensmål. <https://www.skolverket.se/getFile?file=3668> / <https://www.skolverket.se/publikationsserier/regeringsuppdrag/2016/uppdrag-om-nationella-it-strategier-for-skolasendet>

Skolverket. (2018). Problemlösning. [https://larportalen.skolverket.se/LarportalenAPI/api-v2/document/path/larportalen/material/inriktningar/1-matematik/Grundskola/435_problemlosning%20%C3%A5k7-9/1_matematikundervisninggenomproblemlosning/pdf_prob_ak7-9_del1%20\(1\).pdf](https://larportalen.skolverket.se/LarportalenAPI/api-v2/document/path/larportalen/material/inriktningar/1-matematik/Grundskola/435_problemlosning%20%C3%A5k7-9/1_matematikundervisninggenomproblemlosning/pdf_prob_ak7-9_del1%20(1).pdf)

Skolverket. (n.d.a). Gymnasieprogrammen. <https://www.skolverket.se/undervisning/gymnasieskolan/laroplan-program-och-amnen-i-gymnasieskolan/gymnasieprogrammen>

Skolverket. (n.d.b). Ämne - Matematik. <https://www.skolverket.se/undervisning/gymnasieskolan/laroplan-program-och-amnen-i-gymnasieskolan/gymnasieprogrammen/amne?url=1530314731%2Fsyllabuscw%2Fjsp%2Fsubject.htm%3FsubjectCode%3DMAT%26tos%3Dgy&sv.url=12.5dfee44715d35a5cdfa92a3>

Skolverket. (n.d.c). Ämne - Programmering.

https://www.skolverket.se/undervisning/gymnasieskolan/laroplan-program-och-amnen-i-gymnasieskolan/gymnasieprogrammen/amne?url=1530314731%2Fsyllabuscw%2Fjsp%2Fsubject.htm%3FsubjectCode%3DPRR%26courseCode%3DPRRPRR01%26tos%3Dgy&sv.url=12.5dfee44715d35a5cdfa92a3#anchor_PRRPRR01

Skolverket. (n.d.d). Kurser och utbildningar.

<https://www.skolverket.se/skolutveckling/kurser-och-utbildningar#/>

Skolverket. (n.d.e). Stöd i arbetet.

<https://www.skolverket.se/skolutveckling/inspiration-och-stod-i-arbetet/stod-i-arbetet>

Taflin, E. (2007). Matematikproblem i skolan: För att skapa tillfällen till lärande.

Teague, D. Lister, R. (2014). Blinded by their Plight: Tracing and the Preoperational Programmer .

The royal society. (2012). Shut down or restart? The way forward for computing in UK schools. <https://royalsociety.org/~media/education/computing-in-schools/2012-01-12-computing-in-schools.pdf>

The royal society. (2017). After the reboot: computing education in UK schools. <https://royalsociety.org/~media/policy/projects/computing-education/computing-education-report.pdf>

UNDP. (n.d) Globala Målen. <https://www.globalamalen.se/>

Utbildningsdepartementet. (2015). Uppdrag att föreslå nationella it-strategier för skolväsendet. <http://www.regeringen.se/4a80e6/contentassets/a22b7decc51047a790f68d63c64920cb/uppdrag-till-skolverket-att-foresla-nationella-it-strategier-for-skolvasendet.pdf>

Utbildningsstyrelsen. (n.d.a). Grunderna för gymnasiets läroplan förnyas.

<https://www.oph.fi/sv/utbildning-och-examina/grunderna-gymnasiets-laroplan-fornyas>

Utbildningsstyrelsen. (n.d.b). Grunderna för gymnasiets läroplan 2019.

https://www.oph.fi/sites/default/files/documents/grunderna_for_gymnasiets_laroplan_2019.pdf

Vetenskapsrådet. (2017) God forskningsed.

Zulnaidi, H., Hidayat, R., & Oktavika, E. (2020). Effect of use of GeoGebra on achievement of high school mathematics students. Education and Information Technologies, 25(1), 51-72.

8. Bilagor

8.1. Intervjuguide

Bakgrundsinfo

- Är som matematiklärare
- Andra behörigheter
- Erfarenhet inom programmering
- Vilka mattekurser?

Lärarens bild

- Hur har din undervisning förändrats?
- Vilka svårigheter har du upplevt för dig som lärare?
- Finns det några sätt du skulle vilja implementera programmering men som du inte har gjort?
- Vidareutbildning?
 - Har du gått någon kurs etc?
- Har du använt någon speciell utrustning?
 - Är det något du saknar?

Undervisningen

- Vilket centralt innehåll har du använt programmering i?
- Vilka kurser?
- Har det blivit någon skillnad i kurslitteratur?
- Har du gjort någon skillnad på de elever som har erfarenhet inom programmering jämfört med de som aldrig tidigare har programmerat?
- Hur kan en typisk lektion se ut?
- Jobbar eleverna enskilt eller i par/grupp?

Stöd

- Har det funnits stödmaterial från dig som lärare?
 - Från skolverket eller från annat håll?
- Vad skulle du vilja ha för stöd?
 - Vad skulle du behöva hjälp för att nå längre om du skulle vilja göra mer?
- Diskussioner inom arbetslagen/skolan?

TRITA-ITM-EX
2020:476