



**MÄLARDALENS HÖGSKOLA
ESKILSTUNA VÄSTERÅS**

ELEVUPPFATTNINGAR OM SAMBAND MELLAN PROGRAMMERING OCH MATEMATIK

En fenomenografisk studie om några gymnasieelevers uppfattningar om samband mellan programmering och matematik med fokus på likheter, skillnader, möjligheter och begränsningar

DANIEL ÖSTBERG

Akademien för utbildning, kultur och kommunikation
Teknikdidaktik
Självständigt arbete i teknik för ämneslärare, 15 hp

Handledare: Roger Andersson

Examinator: Gunnar Jonsson

Termin 9, 2019



MÄLARDALENS HÖGSKOLA
ESKILSTUNA VÄSTERÅS

Akademien för utbildning
kultur och kommunikation

SJÄLVSTÄNDIGT ARBETE

OAU253

15 hp

Daniel Östberg

Elevuppfattningar om samband mellan programmering och matematik

En fenomenografisk studie om några gymnasieelevers uppfattningar om samband mellan programmering och matematik med fokus på likheter, skillnader, möjligheter och begränsningar

Student conceptions of the relations between programming and mathematics

HT, 2019

Antal sidor: 32

SAMMANFATTNING

Studien syftar till att undersöka några gymnasieelevers uppfattningar om samband mellan programmering och matematik med fokus på likheter, skillnader, möjligheter och begränsningar. Metoden utgår från en fenomenografisk ansats med intervjuer som transkriberats och analyseras via en kategoriseringsmetod i sju steg. Resultaten visar att eleverna kan se flera samband mellan begrepp (variabler), procedurer (räknesätt, algoritmer) och arbetssätt (problemlösning). Slutsatsen som kan dras är att eleverna uppfattar flera likheter och skillnader ämnena emellan, samt att dessa både bidrar till möjligheter och begränsningar för utveckling av kunskaper och förmågor inom båda ämnen. Exempelvis synliggör programmeringens öppna förhållningssätt till problemlösning att eleverna i matematiken får flera sätt att lösa problem på. Begränsningar förklaras främst i de större skillnader som finns mellan ämnena, exempelvis att programmeringen inte alltid innehåller inslag av matematik eller att programmeringens betygskriterier skiljer sig från matematikens.

Nyckelord: Programmering, matematik, elevers uppfattningar, gymnasieskolan, fenomenografi

ABSTRACT

The study aims to investigate some high school students' perceptions of the relationship between programming and mathematics with a focus on similarities, differences, opportunities and limitations. The method is based on a phenomenographic approach with interviews that have been transcribed and analyzed via a seven-step categorization method. The results show that students can make several connections between concepts, procedures and working methods and that they both see similarities and differences in them. The conclusion that can be drawn is that the students perceive several similarities and differences between the topics, and that they both contribute to opportunities and limitations. For example, open programming approaches to problem solving reveal that students also in mathematics are given several ways to solve problems. Limitations are mainly explained by the major differences that exist between the subjects, for example that the programming does not always include elements of mathematics, or that the programming subjects criteria differ from those of the mathematics' subject.

Innehållsförteckning

1	Inledning.....	1
1.1	Problemområde, syfte, forskningsfrågor och avgränsningar	1
1.1.1	Problemområde	1
1.1.2	Syfte.....	2
1.1.3	Forskningsfrågor.....	2
1.1.4	Avgränsningar	2
2	Bakgrund.....	3
2.1	Definitioner.....	3
2.2	Aktuell debatt.....	4
2.2.1	Pågående digitalisering.....	4
2.2.2	Programmering central i ändringen av läroplaner	4
2.3	Tidigare forskning	5
2.3.1	Elever och lärares uppfattningar om programmering i matematiken 5	
2.3.2	Matematik och programmeringens sammanhängande	6
3	Metod	8
3.1	Metodologi.....	8
3.2	Genomförande	8
3.2.1	Datainsamlingsmetod.....	8
3.2.2	Urval.....	8
3.2.3	Databearbetning	9
3.2.4	Etiska överväganden.....	13
4	Resultat	15
4.1	Likheter och skillnader mellan programmering och matematik.....	15
4.1.1	Likheter	15
4.1.2	Skillnader	17
4.2	Matematikens roll för programmering och programmeringens roll för matematiken	19
4.2.1	Matematikens roll för programmering.....	19
4.2.2	Programmeringens roll för matematiken.....	20

4.3	Uppkomsten- och utvecklingen av att se samband mellan programmering och matematik	20
4.4	Resultatsammanfattning	22
5	Diskussion	23
5.1	Resultatdiskussion	23
5.1.1	Resultatens betydelse	23
5.1.2	Resultatens rimlighet	24
5.1.3	Resultatens begränsningar	28
5.1.4	Resultaten i förhållande till syftet	28
5.2	Slutsats	29
5.3	Metoddiskussion	29
5.4	Framtida forskningsfrågor	32
	Referenser	33
	Informationsbrev om deltagande i forskningsstudie	36
	Intervjuguide	37

1 Inledning

1.1 Problemområde, syfte, forskningsfrågor och avgränsningar

I detta kapitel presenteras problemområdet som ligger till grund för studiens syfte och forskningsfrågor.

1.1.1 Problemområde

I hög takt digitaliseras samhället vilket ställer nya krav på världsmedborgare att utveckla kunskaper om digital teknik för att agera i- och förstå samhället vilket motiverat satsningar på programmeringsundervisning för utveckling av digital kompetens (Europaparlamentet, 2006; Computational Thinking, 2016). Den digitala tekniken är skapad av människan för att tillgodose människors behov och styrs genom maskinprogram. Maskinprogrammen i sig är skapta genom programmering som kan definieras på olika sätt men i korthet beskrivas av Svenska Akademiens ordlista (SAOL 14, 2015a) som att ”analysera ett problem och omforma det till en arbetsinstruktion för en dator” eller enligt Collins ordbok (Collins Dictionary, 2020) “att skriva en sekvens av kodade instruktioner som inmatas i en dator” [egen översättning].

Blom (2019) lyfter dock flera definitioner av programmering och menar att programmering är mer än bara kommandon som datorn ska utföra via given kod. Programmering är också att lösa problem med hjälp av datoranvändning, där Blom poängterar att just problemlösningsmetodiken att gå från problem till lösning är den svåraste biten.

Som följd av behovet av kunskapsutveckling om digital teknik har Regeringen gett Skolverket i uppdrag att ändra läroplaner och kursplaner till att innehålla programmering (Regeringen, 2015). Ändringarna syftar till att utveckla elevernas förmågor att formulera- och lösa problem genom kreativa-, logiska- och strukturerade arbetssätt som ett medel att stärka den digitala kompetensen hos eleverna (Fridolin, Hadzialic & Kaplan, 2015; Fridolin & Damberg, 2016). De förmågor som programmeringen syftar till att utveckla överensstämmer väl med de förmågor matematikämnet syftar till att utveckla (Faag, 2018) vilket bland annat syns i de nya kursplanerna för matematik 1-5 inom den svenska gymnasieskolan.

I och med att ändringen i läroplanen är relativt ny och implementerades först till höstterminen 2018 finns ett begränsat antal studier om resultatet av implementeringen. Det som finns att tillgå är undersökningar om lärares uppfattning- och syn på programmeringen, programmeringens för- och nackdelar vid implementation i matematikundervisningen (Sandell, 2017; Wärnå, 2019; Arkå-Nilsson, 2018). Gällande elevers uppfattning om programmeringens implementering i matematiken finns det inte lika mycket forskning. Faag (2018) undersöker grundskoleelevers uppfattningar om programmering i årskurs 1-4. Vidare finns en fallstudie om både lärare- och elevers uppfattningar om utmaningar och styrkor med programmeringens integration i matematikundervisningen på gymnasiet (Götling & Löfwenhamn, 2018).

Då tidigare studier främst undersökt lärares uppfattningar i frågan samt få undersökningar berört elevers uppfattningar motiveras en undersökning som utgår från elevernas erfarenheter av undervisningen. Då tidigare undersökning inriktat sig på specifika fall samt på grundskolan och elever i årskurs 1-4, riktar sig denna till elever i gymnasieskolan i den

allmänna undervisningen för att få kunskap om elevernas uppfattningar om hur programmeringen och matematiken kan hänga samman.

Då de tidigare studierna (Sandell, 2017; Arkå-Nilsson, 2018; Wärnå, 2019) visat att lärare haft flera svårigheter i att implementera programmering i matematiken, genomförs undersökningen där elever väljs ut som har en matematiklärare med goda programmeringskunskaper, där eleverna också läst någon programmeringskurs för att öka chanserna att eleverna fått en uppfattning av både matematiken och programmeringen.

Då studien avser att ge förståelse för hur eleverna uppfattar programmerings- och matematikundervisning, samt ämnena i sig- och sambanden mellan dem, är en fenomenografisk ansats att föredra (Denscombe 2017; Fejes & Thornberg, 2016). Genom denna metod fås möjlighet att utveckla och analysera data. Som metod för datainsamling rekommenderas intervjuer vilket används i studien för att få kunskap om elevernas upplevelser för att svara på forskningsfrågorna (Denscombe 2017; Fejes & Thornberg, 2016).

1.1.2 Syfte

Detta arbete syftar till att ge kunskap om några gymnasieelevers uppfattningar om programmering och matematik samt undervisning som behandlar båda ämnena. Intressant för studien är hur eleverna ser samband mellan programmering och matematik samt hur de ser på likheter och skillnader mellan programmering och matematik. Även uppkomst- och utvecklingen av sambanden de upplever är intressanta, med andra ord när-, var- och hur sambanden uppstått och utvecklas. I fokus ligger att undersöka skolans bidrag till elevernas uppfattningar om programmering- och matematik då skolan enligt läroplan syftar till att utveckla elevernas digitala kompetens genom just matematik- och programmeringsundervisning samt undervisning som samtidigt har inslag av både programmering och matematik.

1.1.3 Forskningsfrågor

1. Vilka samband mellan matematik och programmering finns utifrån elevernas upplevelser?
2. Hur upplever eleverna att deras
 - a. matematikkunskaper och förmågor utvecklas genom deras programmeringsundervisning?
 - b. Programmeringskunskaper- och förmågor utvecklas genom deras matematikundervisning?
3. Var-, när- och hur har förståelsen för dessa samband uppstått och utvecklats enligt eleverna?

1.1.4 Avgränsningar

Studien har avgränsats till att undersöka hur några svenska gymnasieelever uppfattar relationen mellan matematik och programmering. För att öka chanserna för att eleverna ska ha uppfattningar i frågan söktes elever som både läst- eller läser programmering samt matematik. Ytterligare avgränsas studien att undersöka elever som har en matematiklärare som är förtrogen med programmering då dessa elever sannolikt har större chanser att delta i undervisning där matematikundervisningen har inslag av programmering.

2 Bakgrund

2.1 Definitioner

Nedan följer några definitioner av ord som förekommer ofta som kan ha en bred betydelse, men som i denna studie kommer att ha betydelse enligt följande lista.

Programmering

Programmering utifrån elevernas perspektiv och definition vilken kan variera, men används främst som samlingsnamn för den programmering eleven stöter på inom skolkontexten.

Allmän definition enligt SAOL 14 (2015a) innebär programmering att *analysera ett problem och omforma det till en arbetsinstruktion för en dator*.

Programmeringsundervisning

Undervisning i programmering i skolkontexten.

Matematikundervisning

Undervisning i matematik i skolkontexten.

Matematiska förmågor

De matematiska förmågorna enligt läroplanen för svensk gymnasieskola (Skolverket, 2017); *Begreppsförmåga, Procedurförmåga, Problemlösningsförmåga, Modelleringsförmåga, Kommunikationsförmåga, Resonemangsförmåga och Relevansförmåga*.

Kunskaper och förmågor inom programmering

Nedan följer kunskaper och förmågor inom programmering enligt Skolverket (2017) vilka hänvisas till som programmeringskunskaper- och förmågor.

1. Kunskaper om *programmerbara system, programmeringsparadigm, gränssnitt mot filer, filsystem, databashanterare och internet, färdigheter i objektorienterad programmering och om samspelet mellan program, exekveringsmiljö, operativsystem och hårdvara*.
2. Förmåga att *formulera och planera programmeringsuppgifter med pseudokod och diagramteknik*.
3. Förmåga att *skriva, läsa, strukturera, analysera, dokumentera och kommentera programkod*.
4. Färdigheter i att *skapa program med ett givet syfte och för en avsedd användare*.
5. Förmåga att *finna, analysera, åtgärda och förhindra syntaxfel, programkörningsfel och programmeringslogiska fel*.
6. Förståelse av och färdigheter i att använda datavetenskapliga begrepp och principer.

Samband och koppling

Orden *samband* och *koppling* används i denna rapport som synonymer och definieras som *något som anknyter eller förbinder vissa företeelser vanligen tämligen abstrakt om förbindelser som kan vara svårupptäckta.* (SAOL 14, 2015b).

2.2 Aktuell debatt

I presentationen av problemområdet nämns digitaliseringssatsningar i relation till programmering och matematik. Nedan presenteras några inslag från den aktuella debatten från dessa satsningar.

2.2.1 Pågående digitalisering

För att förbereda EU-medborgare att kunna handla i ett allt mer krävande samhälle har Europaparlamentet och Europeiska rådet tagit fram åtta nyckelkompetenser nödvändiga för medborgare att utveckla. En av kompetenserna är *digital kompetens* som syftar till att förbereda EU-medborgare att på ett säkert och kritiskt sätt kunna använda informationssamhällets teknik i vardagslivet (Europaparlamentet, 2006).

Digital kompetens har därför fått stor betydelse för Sverige med tydliga mål om att ligga i topp gällande användningen av digitaliseringens möjligheter och därför utvecklat Digitaliseringskommissionen för ändamålet. Digitaliseringskommissionen definierar i sin tur digital kompetens som ”Digital kompetens utgörs av i vilken utsträckning man är förtrogen med digitala verktyg och tjänster samt har förmåga att följa med i den digitala utvecklingen och dess påverkan på ens liv” (SOU 2015:28, s. 102). Förtydligat innefattar den digitala kompetensen ”kunskaper att söka information, kommunicera, interagera och producera digitalt, färdigheter att använda digitala verktyg och tjänster, förståelse för den transformering som digitaliseringen innebär i samhället med dess möjligheter och risker” samt ”motivation att delta i utvecklingen” (SOU 2015:28, s. 103. Vidare har Regeringen gett Skolverket i uppdrag att genom ändring av styrdokument, skapa förutsättningar för skolan och dess elever att utveckla elevernas digitala kompetens (Regeringen, 2015).

2.2.2 Programmering central i ändringen av läroplaner

I och med de nya målen i EU gällande digital kompetens (Europaparlamentet, 2006) följer initiativ vars fokus ligger i att utveckla elevers digitala kompetens genom datalogiskt tänkande genom programmeringsundervisning- och inslag av programmering i de vanliga skolämnena (Computational thinking, 2016).

Likt andra länder inom EU (European schoolnet, 2015) satsar också Sverige på att införa inslag av programmering i olika ämnen (Regeringskansliet, 2017). Främst införs programmeringen i teknik- och matematikämnet, där det tydligt framgår att med programmering menas inte enbart att skriva kod, utan också att utveckla andra förmågor och kunskaper (Fridolin, Hadzialic & Kaplan, 2015). Med detta menas att eleverna genom införandet av programmeringens inslag i matematiken, ska få möjlighet att utveckla förmågor som att lösa- och formulera problem genom kreativa-, logiska- och strukturerade arbetssätt som en del av att stärka elevernas digitala kompetens.

Skolverket presenterar 2016 (Skolverket, 2016) ”Redovisning av uppdraget om att föreslå nationella it-strategier för skolväsendet – förändringar i läroplaner, kursplaner, ämnesplaner och examensmål”. Här framgår det att förändringar främst

kommer att ske i syftesbeskrivningar och de centrala innehållet i kursplaner. Matematikens- och teknikämnen kursplaner får de största förändringarna. Enligt den reviderade läroplanen för gymnasieskola (Skolverket, 2017a) står framskrivet att elever efter examen ska kunna ”använda såväl digitala som andra verktyg och medier för kunskapssökande, informationsbearbetning, problemlösning, skapande, kommunikation och lärande” (Skolverket, 2017a, s. 1). Samt i kursplaner för matematik 1c, 2c, 3c, 4 och 5 att undervisningen ska behandla strategier för problemlösning inklusive modellering både med och utan digitala verktyg och programmering (Skolverket, 2017b).

2.3 Tidigare forskning

I och med den aktuella debatten och satsningen på utveckling av digital kompetens genom programmering, har några studier behandlat just detta. Nedan lyfts några för studien relevanta forskningsstudier med deras syfte, resultat och betydelse inom problemområdet.

2.3.1 Elever och lärares uppfattningar om programmering i matematiken

I en undersökning (Faag, 2018) undersöktes hur några elever i grundskolans år 1-4 uppfattade matematiska aspekter av programmering. Resultatet visar att elevernas uppfattningar kan förklaras i kategorierna *styrning*, *matematik*, *arbetsätt* och *stegvis lösningsprocess*. Studien visar att ett samband mellan programmering och matematik kan uppfattas av eleverna, samtidigt som eleverna visar svårigheter i att definiera syftet med programmeringen.

I en annan studie (Wärnå, 2019) med syfte att undersöka hur lärare och elever på gymnasiet upplever implementeringen av- samt arbetet med olika programmeringsprogram gällande svårighetsgrad, programmeringens nytta för matematiken samt undervisningens påverkan på intresse för programmering. Studie visar att det är många elever som både uppfattar programmeringen som svår, men också anser att den inte underlättar förståelsen för matematik.

Götling och Löfwenhamn (2018) presenterar likt Faag (2018) och Wärnå (2019) programmeringsimplementationens styrkor och utmaningar i matematiken ur elevernas perspektiv. Resultatet tyder dels på att eleverna genom programmeringen får möjlighet att upptäcka matematikens tillämpningsområden, att programmeringsarbetet i matematiken bidrar till omväxling i undervisningen samt att de genom programmering får en grund för framtidens digitaliserade samhälle. Dock är flera elever skeptiska till datoranvändandet, att det är ovant och distraherande i undervisningen.

Sammanfattat kan man utifrån dessa studier säga att eleverna uppfattar både positiva- och negativa effekter av programmeringens inslag i matematikundervisningen.

Även lärarnas uppfattning om vad programmering är- samt hur den används i undervisningen varierar enligt flertalet studier (Andersson, 2018; Wärnå, 2019; Arkå-Nilsson, 2018; Sjöberg, 2019). I en studie där grundskolelärares definitioner av programmering samt den betydelse lärarens definitioner har för deras undervisning,

visar att definitionerna av ämnet varierar mellan lärare. Denna variation bidrar till att utformningen av lärarnas undervisning varierar (Andersson, 2018).

Gällande implementeringen av programmeringen i matematikkurser på gymnasiet, visar studier (Arkå-Nilsson, 2018) att lärarna är osäkra på hur själva implementeringen ska gå till. Lärarna ser även positivt på de möjligheter programmeringen medför så som att utveckla matematiska förmågor som problemlösningsförmågan. De negativa inställningarna i studien kopplas till brist på tid, att det redan finns många moment som ska ingå i matematikkurserna samt att kompetensen hos lärarna brister. Även här visar resultatet att lärarna har skilda uppfattningar om begreppet programmering i sig (Arkå-Nilsson, 2018), likt resultatet i undersökningen (Andersson, 2018) på grundskolan.

Även Sandell (2017) presenterar liknande resultat i en studie där införandet av programmeringen i matematikens kursmål påverkar elevers lärande i matematik. Lärarna i denna undersökning anser dels att programmeringen bidrar på flera olika sätt, exempelvis genom att fungera som ett verktyg för att förstå matematik, utveckla elevernas matematiska förmågor eller för att skapa mer intresse för matematiken. Dock menar lärarna också att varje enskild lärares åsikter och inställning till implementeringen påverkar dennes undervisning.

Vidare visar andra resultat (Sjöberg, 2019) på ett missnöje hos lärarna gällande hur snabbt implementeringen skett, samt att lärarna inte funnit något stöd för att ändra sin undervisning. Lärarna efterfrågar också stöd i form av vidareutbildning samt att läromedel ska innehålla inslag av programmering för att underlätta för lärarna.

Sjölander (2018) lyfter även att problemen ser liknande ut i andra länder enligt Sentance & Csizmadia (2015) där likheten finns gällande lärares bristande kunskap i ämnet och även elevernas svårigheter i att förstå programmeringsämnet i sig.

Sammanfattningsvis finns två tydliga bilder lärarna ger av implementeringen; dels att lärarna ser möjligheter i och med programmeringens implementering i matematikkurserna, men också att denna implementering är svår att möta för lärare som saknar kunskaper om hur detta ska göras. I och med dessa svårigheter, som varierar mellan lärare, får också undervisningen olika utseende och programmeringens inslag i matematiken varierar också därefter (Andersson, 2018; Wärnå, 2019; Arkå-Nilsson, 2018; Sjöberg, 2019).

Elevernas svar i undersökningen av Wärnå om huruvida de tycker programmering är svårt samt om programmeringen bidrar till bättre förståelse för matematik, tycks hänga samman med den undervisning eleverna fått (Wärnå, 2019). Med andra ord påverkar lärarens kunskaper i programmering hur läraren utformar undervisningen (Andersson, 2018; Wärnå, 2019), och där igenom hur eleverna uppfattar programmering i sig, men också hur programmeringen förbättrar deras förståelse för matematik (Wärnå, 2019).

2.3.2 Matematik och programmeringens sammanhängande

Gällande matematiken och programmeringens sammanhängande finns delade röster. Enligt en undersökning (Sjölander, 2018) tycker grundskolelärare se tydliga samband, dels mellan alla fem matematiska förmågor och programmering men också till samtliga matematiska områden. Även Faag (2018) menar att programmeringen påminner om matematik, samt att eleverna i grundskolan uppfattar ett samband

mellan programmering och matematik. Dock lyfter Sjölander (2018) i sin studie flertalet andra studier (Lewish & Shah, 2012; Sung et al., 2017; Gülbahar & Kaleliouglu, 2014; Calao, Moreno-León, Correa & Robles, 2015; Benton, Saunders, Kalas, Hoyles, & Noss, 2017) vars resultat är motsägelsefulla till utveckling inom programmering bidrar till utveckling i matematiken. Sammanfattningsvis kan både lärare och elever se tydliga samband mellan programmering och matematiken, medan forskning inte lika tydligt visar att utvecklandet av matematiska förmågor måste ske genom programmering och vice versa.

3 Metod

Med tidigare syftesbeskrivning presenteras i detta kapitel metoden som valts för att svara till syftet och de forskningsfrågor som syftet preciserats i. Först presenteras och motiveras vald metod, datainsamlingsmetod och urval. Därefter presenteras och förklaras databearbetningsmetoden, det vill säga analysmetoden.

3.1 Metodologi

Valet av metod styrs av karaktären på syftet och de frågeställningar som är centrala för studien. Dessa syftar till att ge kunskap om elevers ”... sätt att förstå något eller ett sätt att erfara något” (Fejes & Thornberg, 2016, s. 162) gällande undervisning i matematik- och programmering samt elevernas uppfattning om ämnenas samhörighet. Enligt Fejes och Thornberg (2016) lämpar sig en fenomenografisk eller fenomenologisk metodansats där data samlas in via intervjuer, vilka sedan analyseras och utvecklas. Anledningen till att en fenomenografisk ansats väljs är på grund av att denna studie fokuserar mer på skillnader i informanternas uppfattningar än likheterna. Som tidigare presenterats skiljer sig lärarnas uppfattningar i frågan om definitionen av programmering (Andersson, 2018) vilket påverkar lärarens undervisning (Andersson, 2018; Wärnå, 2019; Arkå-Nilsson, 2018; Sjöberg, 2019) vilket i sin tur tyder på att elevernas uppfattningar också bör variera (Andersson, 2018; Wärnå, 2019), därför styrkes en fenomenografisk ansats enligt Fejes och Thornberg (2016).

3.2 Genomförande

Nedan presenteras hur den valda metodansatsen utvecklas och används.

3.2.1 Datainsamlingsmetod

För att analysen i en fenomenografisk studie ska bli framgångsrik krävs att man så gott det går kommer åt försökspersonernas egna sätt att uppfatta fenomenet (Fejes & Thornberg, 2016). Därför bör man i sina intervjuer endast leda in informanterna på området, för att sedan låta informanterna delge sina uppfattningar. Enligt Fejes och Thornberg (2016) görs detta bäst genom halvstrukturerade intervjuer som är tematiska, som får följa informantens utsagor med hjälp av följdfrågor som håller informanten till temat (även kallas *probing*). Samma metod beskrivs och motiveras av Denscombe (2017) men kallas då för semistrukturerad intervju. För att underlätta vidare analys och spelas också intervjuerna in enligt författarens rekommendationer (Fejes & Thornberg, 2016).

3.2.2 Urval

För studien eftersökes informanter som både är förtrogna med matematik och programmering med motivering att dessa elever sannolikt har mer att säga om ämnena än elever som ännu inte har genomgått dessa typer av kurser. Därför avgränsas studien att undersöka elever som har en matematiklärare som är förtrogen med programmering då dessa elever sannolikt har större chanser att delta i undervisning där matematikundervisningen har inslag av programmering. Ett bekvämlighetsurval görs för att få snabb och smidig tillgång till informanter där redan etablerad kontakt till lärare inom programmering och matematik finns. Detta

för att forskaren ska få större möjligheter att nå både lärare och elever (Denscombe, 2017). Då forskningsansatsen kräver elevernas riktiga beskrivningar av uppfattningar efterfrågas elever som är villiga att dela med sig av sina uppfattningar. Genom lärarna fås information om vilka elever som eventuellt kan tänkas ställa upp som uppfyller kraven om kunskaper i både matematik och programmering. Lärarens medverkan i urvalet kan också medföra att informanter medverkar som inte har något emot att uttrycka sig i studien, vilket också förtydligas med beskrivningar av frivillighetskraven enligt Vetenskapsrådet (2017) både muntligen och skriftligen i informationsbrev. Då forskningsansatsen enligt Fejes och Thornberg (2016) eftersöker skillnader i elevernas uppfattningar, sökes variation mellan informanterna som kan ha betydelse för hur de svarar, därför tillfrågas elever i två olika programmeringsgrupper med olika lärare att medverka i studien. Eleverna går främst på teknik- eller naturprogrammet men informanter från andra program förekommer också.

Informantantalet och intervjuernas längd anpassades till studiens omfattning, vilken främst begränsas av given tid för arbetet. Informantantalet är 10 och intervjuernas längd varierade mellan fem och femton minuter. Studien är genomförd i en medelstor stad.

3.2.3 Databearbetning

För analys av data kommer Dahlgren och Fallsbergs (1991) analys i sju steg följas som en grund för analys enligt Fejes och Thornberg (2016) med motivering att höja validiteten av analysen. Stegen i modellen kommer presenteras nedan i ordningen 1-7 med exempel på hur stegen genomförts i bearbetningen av data från intervjuerna samt resultaten av stegen var för sig.

Steg 1 – att bekanta sig med materialet

För att bli bekant med materialet lyssnades de ljudinspelade intervjuerna igenom flera gånger. Sedan transkriberades dessa samt lyssnades igenom igen samtidigt som de lästes igenom. Detta gjordes för att säkerställa att informanternas uttalanden uppfattats på ett riktigt sätt, vilket säkerställs genom att läsa texten samtidigt som man lyssnar på ljudfilen för att snappa upp sådant som är svårt att få med i skrift. Genom detta blir transkriberingen genomförd på korrekt sätt. Därefter raderades ljudfilerna enligt överenskommelse med informanter och som en del av uppfyllandet av konfidentialitetskriteriet (Vetenskapsrådet, 2017).

Steg 2 – kondensation

Analysen startas genom att betydelsefulla och signifikanta delar av transkriptet plockades ut vilka gav representativa bilder av de intervjuade fenomenen. Här var allt intressant som hade med elevers uppfattningar och undervisning att göra som kunde kopplas till matematik eller programmering samt ämnenas innehåll och förmågor. Delarna som plockades ut kallas för *passager* och själva steget för *kondensation*. För att enkelt hålla samman passager med informanter fick varje informant en unik färg. Passagerna numrerades också i varje transskript för att kunna hitta tillbaka till uttalandet i det ursprungliga transkriptet. Således, enligt exemplet nedan, står en turkos tvåa för *passage nummer två* av *informant turkos*. Egna kommentarer är skrivna i orange och kursivt för att skilja på informanternas uttalanden och mina egna tankar. Text i lila är *passagerna*, som försöker fånga upp det betydelsefulla och

signifikanta i varje uttalande av informanterna. Nedan följer ett exempel ur ett transskript där noteringar finns.

[Informant 2] Jo det kan ju vara mycket att man ska göra uppgifter som nått spel till exempel, och sen så måste man göra beräkningar i själva kodningen, och att det håller ihop på själva det sättet, så då kan ja se då, det är ju matte överallt.

3 *Kan se att det hänger ihop i själva uppgifter med beräkningar, men också att "det är matte överallt".*

3 *Lösa uppgifter:* Jo det kan ju vara mycket att man ska göra uppgifter som nått spel till exempel

3 *Beräkningar som exempel på matte:* sen så måste man göra beräkningar i själva kodningen

3 *Få programmet att funka genom kod som innehåller matematik:* så måste man göra beräkningar i själva kodningen och att det håller ihop på själva det sättet

3 *Det är matte överallt:* Jo det kan ju vara mycket att man ska göra uppgifter som nått spel till exempel och sen så måste man göra beräkningar i själva kodningen och att det håller ihop på själva det sättet, så då kan ja se då, det är ju matte överallt

En tanke var att redan i detta stadie försöka se både ytliga och djupa betydelser i informanternas utsagor för att i ett tidigt skede bekanta mig med- och väcka tankar som kan vara relevanta för framtida jämförelser i kommande steg. I nästa steg 3 ges exempel på en jämförelse som både har en ytlig- och djupare betydelse för att exemplifiera detta.

Steg 3 – jämförelse

Olika passager från steg två jämfördes för att hitta likheter och skillnader. I baktanke låg fenomenografins mål att finna skillnader, men att vi genom att hitta likheter även kan finna skillnader. För att kunna jämföra en större mängd passager för att hitta likheter och skillnader klipptes passagera ut. Viktigt var att se till både ytliga och djupa likheter och skillnader för att kunna skapa grupper att placera passagera i. Som exempel lyfts två passager där två informanter nämner att problemlösning är något som finns med i både programmeringen och matematiken. Den ena informanten tillägger också att problemlösning i programmeringen är öppen för flera lösningsalternativ, medan man inom matematiken är fäst vid en lösningsmetod som anses vara "den rätta". I fallet finns både en likhet, det vill säga *problemlösning*, men också en skillnad i *vad problemlösning innebär* i de olika ämnena.

Likheter – Problemlösning

Skillnader – Vad problemlösning innebär

Därför skapades i detta steg en grupp för *likheter mellan programmering och matematik* och en för *skillnader mellan programmering och matematik*, där *problemlösning* hamnade i *likheter* och *vad problemlösningen innebär* hamnade i *skillnader*. På detta sätt togs hänsyn till passagers olika innebörd.

Steg 4 – gruppering

Funna likheter och skillnader från steg tre grupperades. Passagerna från steg två lades i grupper där likheter och skillnader styrde grupptillhörigheten. Åter igen var det viktigt att se till både skillnader och likheter mellan passager. I och med att redan i steg 1 försöka vara uppmärksam på både likheter och skillnader samt ytliga och djupa betydelser fanns ibland två passager som båda innehöll urklipp av samma informantuttalande. Detta exemplifieras nedan där en informant svarar på frågan om vilka kurser hen läst som haft inslag av programmering.

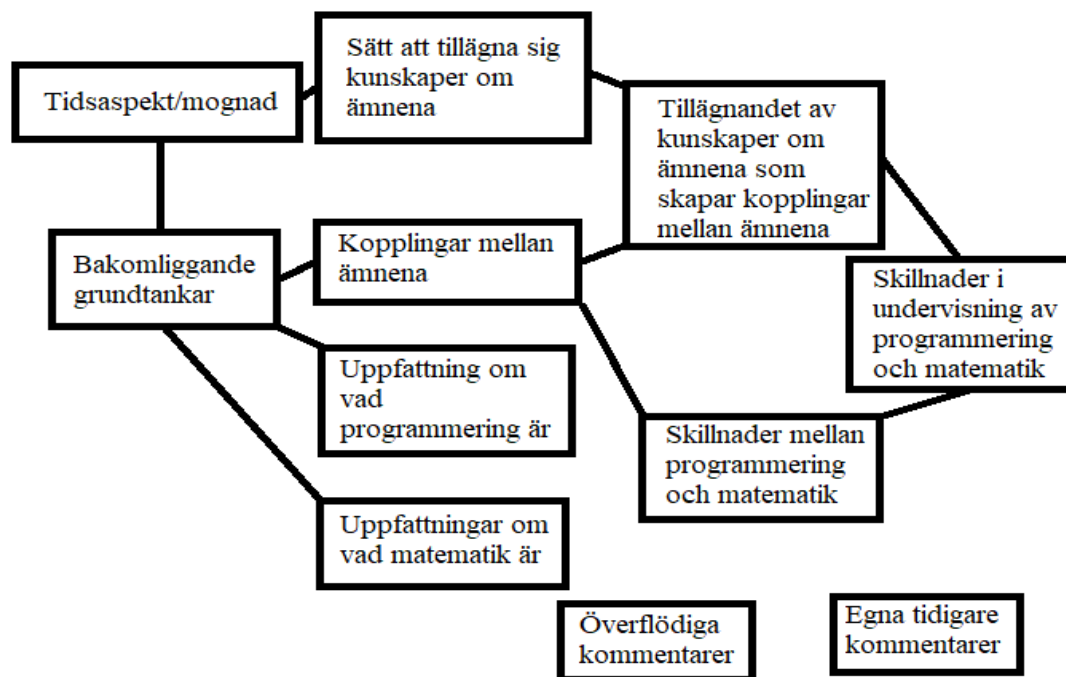
Uppfattning om vad programmering är: Ja, jag har läst programmering 1, sen har jag läst webbutveckling nu och det är bara där programmering ingår.

Nämner inte matematik med inslag av programmering: Ja, jag har läst programmering 1, sen har jag läst webbutveckling nu och det är bara där programmering ingår.

På detta sätt kunde flera betydelser av samma uttalande fångas upp utan att bli bortglömt i grupperingen och på så sätt säkerställa att flera dimensioner av samma uttalande fick sin särskilda plats i grupperingen. I exemplet var det viktigt att dels se vad informanten ansåg ha inslag av programmering, men också att se vad informanten inte ansåg ha inslag av programmering.

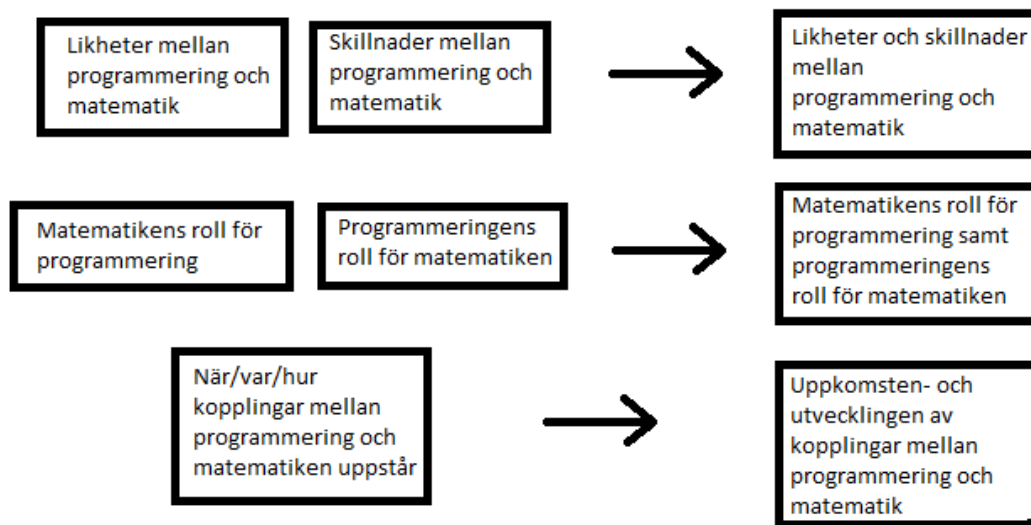
Steg 5 – artikulera kategorierna

I detta steg 5 *artikulerades* kategorierna genom att låta likheterna mellan två passager stå i fokus. Det som eftersöktes var kärnan av likheterna för att låta antydning till kategorierna växa fram. Noterat var att det kan vara oklart när en ny kategori bör bildas, det vill säga att avgöra när två likheter är tillräckligt olika för att delas upp i två kategorier- eller tillräckligt lika för att slås ihop. Med en stor mängd passager bildades till en början 10 grupper med innehåll styrt efter kärnan av likheterna mellan passagerna, även samband mellan de olika grupperingarna noterades genom streck som sammanbinder grupperna, se figur 1 nedan.



Figur 1, de elva grupper i första gruppbildningen utifrån passagers likheter.

Från denna uppdelning kunde likheter och skillnader mellan grupperna urskiljas för att slå samman grupper vars kärna hade likheter. Exempelvis svarar gruppen *tidsaspekt/mognad* på *när* och *hur* elevernas sätt att koppla samman programmering och matematik vuxit fram samt utvecklas. Samtidigt ger gruppen *Sätt att tillägna sig kunskaper om ämnena* en bild av *hur* eleverna får kunskaper om båda ämnena. Därmed har de starka likheter vilket motiverar att de slås samman. På detta sätt arbetades nya grupper fram med fokus på de likheter som fanns mellan grupperna. Därefter gick alla passager i samtliga elva grupper igenom för att se om de fortfarande passade in i den sammanslagna gruppen, om passagen behövde byta grupp eller om en ny grupp behövde bildas. Vidare genomfördes denna process igen utifrån samma arbetsgång för att skapa tre grupper utifrån de fem grupper som vuxit fram ur de elva initiala grupperna. Figur 2 visar hur grupperna ändrats från fem till tre grupper.



Figur 2, de fem och senare tre grupperna i andra- och tredje gruppbildningen utifrån gruppernas- och passageras likheter och olikheter

Steg 6 – namnge kategorierna

I figur två har de tre kategorierna vuxit fram ur de grupper som bildats genom de tidigare stegen. Kategorierna får namn enligt figur 2 vars innehåll beskrivs utförligt i kommande resultatavsnitt. Namnen syftar till att kort och koncist ge information om dess innehåll.

Steg 7 – kontrastiv fas

I denna slutliga fas granskades samtliga passager mot de tre kategorierna som vuxit fram, med andra ord gick samtliga passager igenom för att se att de fortfarande hörde hemma i kategorin, samt att alternativa betydelser också fanns noterade i de andra kategorierna. Som exempel kan åter igen problemlösning tas, som tillhör både *Likheter och skillnader mellan programmering och matematik* och *Matematikens roll för programmering samt programmeringens roll för matematiken*. Dels har ämnena problemlösning som gemensamt arbetssätt, men samtidigt skiljer sig problemlösningens innebörd ämnena emellan. Programmeringens problemlösning har också en roll att utveckla matematikens problemlösning genom synliggörandet av att finna flera lösningar till ett problem. De slutliga kategorierna som kvarstod är resultatet av analysen och utgör *utfallsrummet*. Kategorierna är nu skapta ur passagera som reducerats till korta kärnfulla citat.

3.2.4 Etiska överväganden

Under studiens gång har informanter informerats både muntligen och skriftligen via brev (se bilaga Informationsbrev) om studiens mening och användning samt att deltagande i studien är frivillig och att deras identiteter eller deltagande inte går att spåra enligt vetenskapsrådets etiska principer (Vetenskapsrådet, 2017) för att uppfylla informations-, frivillighets-, nyttjande- och konfidentialitetskrav (Tivenius, 2015). I den mån det går har elevernas uttalanden återanvänts ordagrant utan att avslöja elevernas identitet. Där det har behövts har exempelvis upprepningar och

pauser tagits bort med kravet på att de inte påverkar innebörden av dess originella mening.

4 Resultat

I detta avsnitt presenteras resultatet av analysen i föregående avsnitt genom att presentera de tre kategorierna som utgör utfallsrummet. De kategorierna har gemensamt är kopplingen till matematik, programmering eller både matematik och programmering, samt att samtliga kategorier bygger på elevernas uppfattningar. I enighet med syftet i studien lyfter resultatet de samband eleverna upplever mellan programmering och matematik. Kategorierna är uppdelade i tre delar där den första lyfter upplevelser om likheter och skillnader mellan programmering och matematik.

Den tredje kategorin innehåller elevernas upplevelser om hur programmeringen spelar roll i matematiken samt hur matematiken spelar roll i programmeringen. Denna kategori är kopplad till den första kategorin då den bygger på de likheter och skillnader eleverna kan se mellan matematik och programmering, men fokuserar mer på effekter av likheter och skillnader snarare än redogör för specifika likheter och skillnader (vilket den första kategorin istället gör).

Den andra kategorin lyfter elevernas upplevelser om när-, var- och hur samband uppstår mellan programmering och matematik. Denna kategori beskriver alltså uppkomsten- och utvecklingen av elevernas upplevelser om när-, var- och hur eleverna uppfattat likheter, skillnader samt ämnenas olika roller för varandra och hänger på så sätt ihop med de tidigare två kategorierna.

Kategorierna svarar i sig själva- och tillsammans med varandra på de forskningsfrågor som är aktuella för studien. Genom innehållet i kategorierna presenteras elevernas upplevelser om samband mellan programmering och matematik i form av likheter, skillnader, samband samt frågor om sambandens uppkomst- och utveckling.

4.1 Likheter och skillnader mellan programmering och matematik

Eleverna upplever flertalet likheter och skillnader mellan programmering och matematik både i skolkontexten och i vardagslivet. Till att börja med presenteras övergripande likheter som eleverna upplever mellan programmering och matematik varefter konkreta likheter presenteras. Därefter lyfts skillnader mellan ämnena utifrån elevernas uppfattningar.

4.1.1 Likheter

Bland de övergripande likheterna eleverna upplever mellan programmering och matematik är ämnenas syfte och karaktär. Syftet med ämnena är enligt eleverna att lösa problem, dels för att leverera resultat men också för att i framtiden kunna lösa verkliga problem och förberedas för framtiden.

Informant 10 - Det (inom ämnena) följer mycket av samma tankesätt ibland, kan jag säga, men jag tror att dom båda handlar mycket mer om logik än själva matte och programmering (som ämnen) och att båda tar mycket mer inflytande från bara det. Men visst man märker ju hela tiden, att man använder samma typ av sätt för att lösa uppgifter, det är det båda handlar om helt enkelt.

Informant 7 - Det beror på vad man ska plugga vidare till och vad man ska utbilda sig till sen, matten har du väl i det mesta i alla fall om du ska vidare till högskola, och programmeringen är mer specifik om du ha nån teknisk utbildning.

Båda ämnena karaktäriseras av ett logiskt tankesätt och bygger på utforskande, laborativa och experimenterande metoder. Eleverna upplever också att det finns tydliga likheter gällande uppgifter i ämnesundervisningen inom programmering och matematik, samt att det i båda ämnen mestadels handlar om att hantera siffror.

Informant 10 - Jag får ju utnyttja de matematiska kunskaperna praktiskt här (i programmeringen), det märks ju, och det är ju för att lösa mina egna problem som jag då har skapat med programmeringsprogrammet och sånt, för programmering handlar ju nämligen om att vi försöker få fram någonting, men man måste ju skriva det på något sätt, och då måste man lösa problemet också för hur man skriver det och då kommer ju alltid matematiken fram, det blir ju så man måste tänka när man-, speciellt med såna konditioner som upprepar sig, man tänker ut hur många gånger den ska upprepa sig, och då kommer det handla om matematiken också.

Mer konkreta likheter utgörs av begrepp och procedurer. Många av begreppen eleverna stöter på i programmeringen kan de koppla till matematiken då de har liknande namn. Det engelska språket som genomsyrar utvecklingsmiljöerna inom programmering hjälper eleverna att se samband mellan begrepp genom översättning till svenskan, och genom dessa upptäcker eleverna samband mellan programmeringen och matematiken.

Informant 4 - Jag förstår det själv mest genom att jämföra med engelskan, så det hjälper mycket för att språket är i engelska.

Några exempel på sådana begrepp är *integer*, *double* och *variable* som eleverna kopplar till *heltal*, *decimaltal* och *variabel* genom översättning. Eleverna känner också igen flertalet av begrepp från matematiken, exempelvis *modulus* och *binära tal*, vilka eleverna använt inom både programmering och matematik.

Vidare kopplar eleverna ihop programmering och matematik genom de beräkningar de utför samt räkneoperationer de använder, det vill säga matematikens *aritmetik*.

Informant 5 -... eftersom båda handlar om siffror, hur man ska koppla dom tillsammans med algoritmer och att allt det här, och också vi har lärt oss att, hur man ska använda beräkningar inom IT, alltså programmering, men ibland så kan man inte se själva matte i programmering, för vissa tillfällen, men när det kommer till beräkningar och så, och speciellt vissa områden så är, matte är där liksom.

Eleverna kommenterar de fyra räknesätten samt inbyggda funktioner de stött på inom programmeringen som exempelvis *sqrt* och *pow*, (det vill säga *roten ur* och *upphöjt till*) som exempel på ovan beskrivna likheter.

Eleverna ser också tydliga samband mellan de matematiska idéer och samband de använder inom programmeringen. Som exempel lyfts samband, formler och algoritmer kopplade till *Pythagoras sats* och *pq-formeln* men också metoder som att *deklarera variabler*, *iterera* (eller inom programmering *loopa*) samt ställa upp villkor med beslut i form av *true or false-satser* som steg i en lösningsprocess.

Gällande arbetssättet ser eleverna tydliga likheter i att gå en viss väg för att lösa problem.

Informant 10 - Ja man kan ju säga att, när man löser en uppgift, då gör man ju en slags formel då och det är ju nästan samma om man skulle skriva in den på programmering, att skriva in en uppgift, man kan ju skriva in hur man räknar ut något, med programmering, ”det är ju bara att gånga och dividera med det här eller det här” så får man ju samma sak som om man skulle göra det direkt i matten.

Informant 2 - Mm jo, alltså ... För att om jag till exempel skulle lösa ehm, en, lösa ut c kanske, då har jag ju skrivit i kodningen att räkna ut vad a är och höja upp det i två och sen b i två och sen vad blir c då? Och sen c² är det här och sen roten ur de. Och det är ju samma sak i matten och i programmeringen, liksom det är båda samma sätt, och det är samma sak om man skulle räkna ut vinkeln till exempel utifrån den här a delat med b är lika med cos vinkel

Som exempel på detta nämner eleverna att man vid lösning av ett visst matematiskt problem går en viss väg via bestämda steg. Inom programmeringen menar de att samma problem hade haft samma lösningssteg, men att man hade varit tvungen att översätta problemet till kod så att datorn kan förstå det. I båda ämnen delas problemet upp i mindre steg vilka löses i tur ordning, där det är viktigt att varje steg löses korrekt.

Informant 5 - Ja, vi har mest med hur man ska lösa problemet i programmering matematiskt, typ att hur ska vi behandla algoritmer på ett speciellt sätt, eftersom du inte kan skriva en algoritm med 1, 3 sen du går tillbaka till 2 sen kör 5, nej, du måste ha 1, 2, 3 och sen du måste också göra det säkert, att du gör steg 1, steg 2, du måste göra det säkert att steg 2 är rätt. I den där steg två du har typ ifall den och ifall den här, samma sak med matte, när du är klar med att lösa problemet så får du ta resultatet sen och testa den så du ser ifall ditt resultat är rätt eller fel

Informant 2 - Ja jo, man... man har ju, man börjar ju liksom från en viss, man måste ju börja på nått sätt, och utifrån det börjar man, en viss väg, det gör man i programmeringen och i matten också. Jag kan ju se programmeringen i matte också.

Informant 9 - Programmering, alltså som programmerare, man behöver dela upp problemet, sen lösa det del för del, alltså steg för steg, även i matte, man behöver dela problemet för att lösa det lättare, det blir lättare om du tar det steg för steg och löser

4.1.2 Skillnader

Eleverna upplever flera skillnader mellan programmering och matematik, främst inriktat på undervisningens utformning med bakomliggande betygskriterier. Nedan presenteras ett antal skillnader eleverna upplever finnas mellan ämnena och undervisning i ämnena. Först presenteras en skillnad i svårighetsnivå mellan ämnena följt av tre andra skillnader några av elever upplever.

Av de skillnader som majoriteten av eleverna nämner är nivån på programmeringen i jämförelse med matematiken. Eleverna upplever att deras programmeringskunskaper och den undervisning de deltar i är på en grundläggande nivå samtidigt som deras matematikkunskaper- och matematikundervisning är på en avancerad nivå.

Informant 9 - ... den matte vi läser är väldigt avancerad, det kanske inte används i programmering, det är olika nivåer, jag läser grund i programmering och det andra är matte 4, det finns ingen koppling, men kanske matte 1c.

Informant 7 - Den matte vi gör nu är mer imaginära tal och sånt, så det är inte så mycket man kan koppla till programmering. Kanske inte på det här stadiet <programmeringsstadiet>.

De menar att de inom programmeringen inte använder några avancerade matematiska formler, samt att svår matte inte kan implementeras i den grundläggande programmeringen. Med andra ord upplever eleverna att det finns ett glapp i deras kunskaper i de två ämnena, samt att detta förhindrar att deras matematikkunskaper kan utvecklas genom programmeringsundervisningen. De ser också större möjligheter att utveckla sina matematiska kunskaper om deras programmeringskunskaper och programmeringsundervisning varit på en högre nivå.

Informant 7 - Nja, inget jag kommer på direkt, hade vi legat lika i hur långt fram vi hade kommit i programmeringskunskaper som i mattekunskaper kanske det hade varit en helt annan fråga, men nu ligger dom så långt ifrån varandra så då är det svårt att säga nånting.

Dock finns en tanke om att möjligheterna kommer med en balans i nivå mellan kunskaperna i ämnena.

Informant 9 - Kanske om man läser vidare (inom programmering) så blir det svårare (matematik).

Med andra ord anser eleverna att de genom grundläggande programmering och den matematiken de stöter på där, inte kan utveckla redan befintliga kunskaper.

Informant 1 - eftersom att jag har redan lärt mig det i förväg, det jag har lärt mig i programmering har jag redan lärt mig innan (Syftar på matematiken).

Dock ser de möjligheter att repetera de befintliga kunskaper, och på så sätt stärka dem.

Informant 8 - Alltså jag får ju öva, och övning ger ju färdighet, så jag glömmer inte bort det i alla fall.

En stor skillnad är att man inom programmeringen snabbt får feedback med hjälp av den utvecklingsmiljö man arbetar i. Skriver man något fel får man veta det direkt då delar av koden man skriver blir felmarkerade. I jämförelse med matematiken, menar eleven att sådan info inte finns och att man istället får vänta ända till dess man fått tillbaka ett prov för att upptäcka felet.

Informant 8 - I programmeringen, du måste skriva liksom rätt, för annars blir det rött, det blir fel, i matten kan jag skriva hur mycket fel som helst utan att någon säger att det är fel, förrän jag ser det efteråt på provet. Så att, det är ju lite, på programmeringen vet man när man gjort fel, i matten är det inte samma sak, ifall man inte får rätt svar då har man gjort fel, men på proven vet man ju liksom inte vad som är rätt svar, så att, det är ju lite skillnad.

Skillnader upplevs också mellan problemen och uppgifterna i ämnena där matematikens uppgifter anses vara skilda från omvärlden och snarare inriktade på specifika situationer. I motsats upplevs programmeringens problem vara sådana som ger riktiga resultat och som svarar på riktiga problem. Även utanför skolan, kommenterar en elev att matematiken ger få exempel på tillämpningar medan eleven i många fall kan se hur programmering skulle kunna implementeras för att lösa problem i vardagen.

Informant 10 - Än så länge har jag ju inte använt matematiken så mycket praktiskt själv förutom att räkna ut några korta saker hemma, så jag kan inte säga att jag vet så mycket om det området (matematikens nytta i att lösa praktiska problem), för i programmering gör vi mycket mer program praktiskt, då får jag själv komma på lösningarna till "det här och det här", men i matte just nu, övar vi bara på hur man löser det i specifika situationer.

Informant 10 - Jag får ju utnyttja de matematiska kunskaperna praktiskt här, det märks ju, och det är ju för att lösa mina egna problem som jag då har skapat med programmeringsprogrammet och sånt, för programmering handlar ju nämligen om att vi försöker få fram någonting, men man måste ju skriva det på något sätt, och då måste man lösa problemet också, för hur man skriver det, och då kommer ju alltid matematiken fram, det blir ju så man måste tänka när man, speciellt med såna konditioner som upprepar sig, man tänker ut hur många gånger den ska upprepa sig, och då kommer det handla om matematiken också

Gällande problemlösning finns uppfattningar om att programmeringen är mer öppen och tillåtande. Inom matematiken upplevs lösningsgångar för problem vara stängda och begränsade till ett enda rätt sätt. Med andra ord tar problemlösningen inom programmering större hänsyn till variation och synliggör fler vägar och sätt att lösa problem på. Sammanfattat menar eleverna att programmeringen bidrar till en bredare problemlösningsförmåga och insikt i flera lösningsätt.

Informant 5 - Programmering hjälper mer med problemlösning, ren problemlösning, den kan inte lära dig hur du ska använda siffror, den hjälper dig hur du ska, liksom, behandla siffrorna rätt, hur du ska lösa ett visst problem, hur kan du tänka dig att lösa, hur du ska ta vägen, där mer att konstruera själva vägen åt dig, och att ge dig mer, flera alternativ, mer än att bara "så där löser du problemet". Som ett matematiskt problem, ingen, jag är 100% säker att det är någon matematiker som tänker att "det här problemet har bara en lösning", att det finns ett sätt att lösa, det finns tusentals sätt att lösa ett problem och vi vet som student att, som elev, att det finns bara det här sättet som jag lärt mig i boken, men när det kommer till programmering kan jag se 3, 4 sätt som du kan lösa, så om du inte förstår det här sättet, kan du använda det andra som är lättare för dig, det har programmering lärt oss. På så sätt kan jag använda det här (programmeringskunskaper) i matten.

Informant 9 - Man kan göra vad som helst, man kan bestämma mycket (inom programmering), men i matte är man begränsad, man är styrd, man måste lösa på ett visst sätt. Men när du programmerar ett program, du kan, det finns flera olika sätt att programmera just det programmet, men matte, för att lösa ett problem finns bara ett alternativ.

4.2 Matematikens roll för programmering och programmeringens roll för matematiken

Eleverna upplever flera samband mellan programmeringen och matematiken där de ger exempel på hur matematiken används inom programmeringen samt hur programmeringen används inom matematiken. Först presenteras elevernas uppfattningar om matematikens roll för programmering och därefter omvänt - programmeringens roll för matematiken.

4.2.1 Matematikens roll för programmering

Många av eleverna ser matematiken som en grundförutsättning för programmeringen och vissa att programmeringen är baserad på matematik och inte tvärt om.

Informant 10 - Matematiken föreslår ju lösningar för olika problem som kan komma sig när man skriver programmeringsprogram, å andra sidan så vet jag inte om så mycket man läser i programmeringen kan användas i matematiken vilket är intressant, det känns mer som att programmeringen är baserad på mycket på det man läser i matematiken, och det märks i alla fall, det är intressant.

Dock menar vissa att programmering i vissa fall inte är lika beroende av matematik, exempelvis om problemet saknar anknytning till matematik.

Informant 5 - Vi kommer prata mer om vilken typ av problem det är, om det är rent matematiskt eller det är någonting helt annat, som inte har någonting att göra med matte. (Då har det inte någon märkbar koppling till matematik)

Eleverna upplever att matematiken syns extra tydligt i vissa situationer, medan den i andra situationer mest kan ses i tankesätt.

Informant 4 - Ja absolut, för det första så är det, om man ska kunna programmera så måste man kunna matte, speciellt i, alltså, om man snackar webbutveckling så är det HTML-kod och då är det inte så mycket matte utan mer design, hur man ska tänka och layout, men c-sharp, JavaScript och sånt, man måste kunna matte, speciellt om, vi hade typ, vi programmerade Pythagoras sats, vi hade Yatzy, alltså allt handlade om matteuppgifter eller för det mesta, derivata till exempel och då måste man kunna grunden, hur man ska räkna, hur man ska göra och så sätter man in det, eller man använder något språk för att göra det <Programmeringen lösningen alltså>

Genom att beskriva hur en dator fungerar kan eleverna redogöra för matematikens roll i exempelvis omvandlandet från kommandon till maskinkod som kräver exempelvis siffror och binärkod. Med andra ord ser eleverna matematiken som ett språk att översätta verkliga problem till datorn via siffror, variabler och steg som datorn kan följa.

Informant 3 - Man skriver ju kommandon och inte siffror, kommandona är ju siffrorna egentligen som är omvandlade på något sätt för att datorn ska förstå, kommandona gör ju olika grejer och säger till datorn vad den ska göra.

Informant 1 - En dator kan ju inte prata förutom matten, så här nummer och siffror, logiska tecken, så den kan bara veta 1 2 3, den kan ordningen, vi kan inte till säga till den att typ "gör si eller gör så" den måste ha en ordning, det är då den förstår...

Matematiken bidrar även med formler och beräkningsmetoder som behövs för att få datorer och program att fungera, och den som kan matematiken väl har stora

möjligheter att implementera dessa kunskaper i programmeringen. Dock anser flera elever att matematikkunskaper inte stärker förmågan att skriva kod, men istället menar att ens tankesätt förbättras vilket kan underlätta inom programmering. Matematiken kan bidra med lösningar på delproblem, där programmeraren kan fundera kring vilken matematik som behöver vara involverad för att lösa delproblem.

Informant 10 - Alltså, jag tänker ibland i vardagen, så tänker jag på olika saker, man kanske kan lösa det här utifrån ett programmeringsperspektiv eller matteperspektiv, och till exempel om jag kunde programmera fram någonting som bara poffade fram framför mig för att lösa det här problemet, kanske hur skulle det programmet se ut, och sen vilken matematik som skulle vara involverad...

Informant 10 - Jag får ju utnyttja de matematiska kunskaperna praktiskt här (inom programmering), det märks ju, och det är ju för att lösa mina egna problem som jag då har skapat med programmeringsprogrammet och sånt, för programmering handlar ju nämligen om att vi försöker få fram någonting, men man måste ju skriva det på något sätt, och då måste man lösa problemet också för hur man skriver det och då kommer ju alltid matematiken fram, det blir ju så man måste tänka när man, speciellt med såna konditioner som upprepar sig, man tänker ut hur många gånger den ska upprepa sig, och då kommer det handla om matematiken också.

4.2.2 Programmeringens roll för matematiken

Programmeringen har enligt eleverna flera roller inom matematiken. Främst används programmeringen som ett verktyg för att skapa program- och produkter som effektiviserar uträkningar. Som exempel på detta är miniräknare eller egenskapade program som räknar ut exempelvis andragradsekvationer.

Utöver de tillämpningar programmeringen ger i form av produkter, ser eleverna flertalet andra vinster i och med programmeringen. Som första exempel ser eleverna programmeringen som en utforskande arbetsmetod som utvecklade förmågor att lösa problem, testa olika lösningsmetoder och resultat samt att vara kreativ vilket enligt eleverna anses vara positivt inom matematiken.

Informant 5 - Så det är samma sak med programmering, ehm, så ja, så kan liksom det här utvecklas i programmering, så du sitter i matte, läser du matte, så ser du att så kan man testa, det finns olika sätt att testa ditt resultat med, så kan du se att så kan jag göra i programmering. Du behöver inte använda samma siffror och samma sätt men det är typ likt.

Utöver dessa förmågor ser eleverna att man genom programmeringsarbete som involverar matematik, får möjlighet att både repetera- och utveckla matematikkunskaper samt testa sina kunskaper praktiskt.

Informant 8 - Alltså jag får ju öva, och övning ger ju färdighet, så jag glömmer inte bort det i alla fall.

Informant 3 - Fast jag kanske kan använda det, liksom hur jag ställer upp den formeln i matematiken...

Informant 10 - Jag får ju utnyttja de matematiska kunskaperna praktiskt här, det märks ju, och det är ju för att lösa mina egna problem som jag då har skapat med programmeringsprogrammet och sånt, för programmering handlar ju nämligen om att vi försöker få fram någonting, men man måste ju skriva det på något sätt, och då måste man lösa problemet också för hur man skriver det och då kommer ju alltid matematiken fram, det blir ju så man måste tänka när man, speciellt med såna konditioner som upprepar sig, man tänker ut hur många gånger den ska upprepa sig, och då kommer det handla om matematiken också.

4.3 Uppkomsten- och utvecklingen av att se samband mellan programmering och matematik

Nedan presenteras när-, var- och hur samband mellan programmering och matematik upptäcks och utvecklas enligt eleverna. Sambanden är de som tidigare beskrivits i tidigare två kategorier.

Eleverna får på olika sätt kunskaper om programmering och matematik och gör därefter jämförelser mellan ämnena och upptäcker samband, likheter och skillnader. För många av eleverna är det skolan som huvudsakligen utvecklat deras sätt att reflektera kring ämnena men eleverna upplever samtidigt att de på egen hand genom att fundera och tänka logiskt kring saker samt genom upplevelser, kopplar samman programmering med matematik på olika sätt. Elever har även uppfattningar om att programmering och matematik hör ihop som de inte riktigt kan förklara.

Informant 3 - Det känns som att programmering med datorer och sånt, jag har bara tänkt på att det ska vara matte det ska handla om.

Med utgångspunkt i undervisning i programmering och matematik får eleverna ta del av uppgifter och information som läraren presenterar. Eleverna upplever att detta material synliggör likheter mellan ämnena vilka för dem samman både direkt- och indirekt. Som exempel kan läraren prata om- eller visa exempel på när programmeringen och matematiken båda är involverade i problem vilket synliggör samband mellan ämnena. Samtidigt behöver inte läraren tydligt framföra dessa samband utan de upptäcks i efterhand när eleven studerar antingen matematik eller programmering och känner igen sig i den ena eller andra kontexten.

Informant 5 - Alltså, du kan säga framför allt med hjälp av läraren, han eller hon säger inte exakt att "det här är programmering" "det här är matte", "sådär är dom lika" men när dom pratar om själva problemet eller hur du löser problemet så då du kan se liksom hintar för att typ det där typ är lika, dom pratar om lösningen på problemet på samma sätt. Sen när det kommer till att själv, du sitter ner och försöker att lösa problem, du kommer att märka själv, att du löser den steg för steg på samma sätt när du har ett problem, ett matematiskt problem så följer du steg, det är samma steg som du följer för att lösa ett programmeringsproblem, så kommer du märka det på alla sätt (att ämnena liknar varandra).

Utöver lärarens undervisning upplever flera elever att deras eget intresse och utforskande får dem att se samband mellan programmering och matematik. Då lärarens tid är begränsad och programmeringsämnet stort, krävs det ibland att man som elev kompletterar sina kunskaper genom att använda internet eller andra källor för att söka information.

Informant 4 - Alltså, Youtube och internet är en stor del av hur jag pluggar, pluggar till programmering, för man kan inte hitta allt man behöver med hjälp av läraren för, det är ganska stort och läraren hinner inte, och så finns det speciella saker som man måste söka upp...

I fråga om när eleverna börjat se samband mellan programmering och matematik, menar samtliga elever att detta skett i samband med den första programmeringskursen de läst. Som förklaring menar de att de innan inte vetat mycket om programmering och därmed inte kunnat göra se några samband. Eleverna upplever också att upptäckten av samband görs efter en tid in i kursen.

Informant 5 - Det var typ, låt oss säga efter första året jag började i programmering, när var det typ 7an eller 8an, det var där jag började programmering, där började jag märka att det är matte.

Informant 9 - Både och, när jag var på skolan, när jag pluggade själv som man känner igen vissa saker, omedvetet tänker man

Som förklaring till detta menar en av eleverna kan vara att man i början av kursen är helt upptagen med programmering vilket gör att man missar att se sambanden.

Informant 5 - Nej, det, man kan inte märka det direkt, eftersom du är insatt på att göra det där och insatt på göra det där, du vet inte att dom är lika. Men sen när du kollar på resultatet, när dom kommer fram, sen när du bara tar ett steg tillbaka och kollar på båda så kan du se det tydligt att det där följer 1, 2, 3 och det där följer 1, 2, 3, exakt likadant. 1, 2, 3. Och du kan se att här har vi algoritmer och här har vi också algoritmer, båda algoritmer, när du pratar om algoritmer, det har liksom ingen betydelse matematiskt i programmering och

motsatt men de har exakt samma mening, det är samma mening och samma koncept, liksom utförandet är olika.

4.4 Resultatsammanfattning

Sammanfattat upplever eleverna flera samband mellan programmering och matematik vilket exemplifieras i likheter, skillnader och de olika roller ämnena spelar för varandra. Uppkomsten- och utvecklingen av de samband elever upplever beskrivs också med hjälp av att besvara frågor om när-, var- och hur de uppstått och utvecklats.

Bland likheterna finns konkreta likheter som begrepp och matematiska modeller men också djupare likheter i problemlösningstrategier och arbetsmetoder. Samtidigt skildras skillnader i sådant som också utgör likheter, exempelvis inom problemlösning. Eleverna upplever att programmering ger större möjligheter för dem att finna flera vägar inom problemlösning medan matematiken oftast har en lösningsväg som anses vara den rätta. Genom likheter och skillnader som dessa upplever eleverna att ämnena hör samman och spelar olika roll för varandra. Exempelvis ser eleverna att programmeringen i och med den mer fria problemlösningen, kan ge exempel på hur man skulle kunna arbeta i matematiken för att utveckla problemlösning.

Eleverna upplever främst att deras kännedom och förståelse för samband mellan ämnena utvecklas genom undervisningen och att de inte kunnat reflektera över samband mellan ämnena innan de genomgått undervisning i programmering. Eleverna upplever också att utvecklingen av att se samband sker på flera håll, exempelvis genom egna funderingar eller informationssökning. Dock upplever samtliga elever att läraren har en stor roll i att presentera programmering och matematik samt samband som kan upptäckas mellan dem. Vidare upplever eleverna att det inte alltid är säkert att man vid undervisningstillfället ser sambanden utan att man över tid ser likheter och skillnader vid eget arbete och egna reflektioner.

5 Diskussion

I detta kapitel diskuteras först resultaten i förhållande till tidigare forskning och den verksamhet där fenomenet studerats. Vidare diskuteras resultatens rimlighet och saknade aspekter samt till vilken grad syftet och frågeställningarna besvarats genom resultaten. Som avslut på resultatdiskussionen presenteras de slutsatser som kan dras utifrån resultatdiskussionen.

Resultatdiskussionen följs av metoddiskussionen där metodval samt alternativ till den valda metoden diskuteras utifrån för- och nackdelar. På samma sätt diskuteras urval och databearbetningsmetod. Därefter diskuteras analysarbetets genomförande som följs av en sammanfattning av metoddiskussionen.

5.1 Resultatdiskussion

Resultatet diskuteras till att börja med utifrån dess betydelse för den verksamhet det berör, med andra ord diskuteras resultatens innebörd för elever, lärare, undervisning i matematik och programmering samt dess betydelse för elevers utveckling av digital kompetens både i Sverige och i EU. Därefter granskas resultatet kritiskt och dess rimlighet diskuteras. Här kopplas resultatet till tidigare forskning för att jämföras mot resultat från andra studier inom området. Vidare diskuteras sådana aspekter som fattas i studiens resultat, exempelvis skillnader mellan elever, elevgrupper och lärarens kompetens i programmering och andra aspekter intressanta för resultatet. Slutligen diskuteras om syftet är uppnått samt om forskningsfrågorna är besvarade.

5.1.1 Resultatens betydelse

Med elevernas uppfattningar i centrum har denna studie gett exempel på samband mellan ämnena matematik och programmering, vilka i första hand kommer att nås av intressenter inom skolan och speciellt inom matematiken- och programmeringens didaktik. Detta skulle kunna innebära att elever främst gynnas av studien via den kunskapsbildning som sker hos exempelvis pedagoger. Dock säger inget emot att elever med hjälp av denna studie och dess resultat, kan synliggöra de samband andra elever ser mellan ämnena och på så sätt ge eleverna nya perspektiv gällande samband mellan programmering och matematik.

För läraren kan studiens resultat vara extra intressanta då undervisningens innehåll till stor del väljs av läraren. Dessa resultat ger en grund att utgå ifrån för att forma en undervisning om man önskar att snappa upp- eller lägga extra fokus på att synliggöra samband mellan ämnena. Denna studie visar mest sannolikt inte alla dimensioner av de samband eleverna upplever, men kan väcka tankar om den potentialer som finns i undervisning man driver. Med andra ord bör denna studie kunna väcka lärares tankar om hur deras undervisning påverkar eleverna genom de moment och uppgifter som äger rum i undervisningen.

Vidare kan resultaten av studien diskuteras i förhållande till det gemensamma inom ämnena, det vill säga de likheter och skillnader eleverna upplever och hur dessa kan utnyttjas inom undervisningen i vardera ämne. Att ämnena delar begrepp och arbetsprocedurer kanske inte kommer som en överraskning, men det bör ändå vara intressant att se till de likheter och skillnader elever beskriver kring både begrepp och arbetsprocedurer som problemlösning. Resultaten tyder på att ämnena har liknande karaktär och syfte, samt att man inom programmeringen har stora möjligheter att utveckla

sin problemlösningsförmåga utan att styras in på det ena- eller andra spåret. Med andra ord bör programmeringen i undervisningen kunna ge flera exempel på hur vi kan arbeta i matematiken för att gynna elevernas problemlösningsförmåga. Detta visar också att syftet uppfylls med införandet av att i programmering och matematiken utveckla elevernas förmåga att lösa- och formulera problem genom kreativa-, logiska- och strukturerade arbetsätt och då den digitala kompetensen hos elever (Fridolin, Hadzialic & Kaplan, 2015). Med andra ord ger resultatet en hint om de fördelar undervisning kan ha där två ämnen av liknande karaktär och syfte används för att utveckla gemensamma begrepp och förmågor på flera plan. Dels ger undervisningen i respektive ämne utrymme för eleverna att utveckla sina problemlösningsförmågor, samtidigt som variationen i problemlösningsarbetet kan ge förslag på nya sätt att arbeta genom för att stärka problemlösningen i det ena- eller båda ämnena. Även förståelse för begrepp, samband och procedurer stärks genom upprepat arbete i det ena eller andra ämnet.

I förhållande till Regeringens beslut (2015) samt Skolverkets direktiv (2016) om implementering av programmering i matematiken, kan resultaten ge ett exempel på effekterna av läroplansändringen både i korthet men sannolikt också på sikt. Rent konkret visar resultaten vilka samband eleverna kan se mellan ämnena samt att dessa leder till utveckling av förmågor. Under studien förutsätts dock att eleverna läst både programmering och matematik, samt att läraren har goda kunskaper i både ämnena, vilket sannolikt påverkar resultatet med tanke på tidigare forskning som utgått från lärarnas perspektiv (Sandell, 2017). Därför fokuserar denna del av diskussionen på de möjligheter som troligtvis finns om lärarnas kompetens i programmering stärks, samtidigt som eleverna får möjlighet att antingen läsa programmering i en enskild kurs eller genom matematikundervisningen i enighet med Skolverkets förslag (2016). Möjligt är, likt tidigare diskuterat, att elever inom Sverige som får undervisning i båda ämnena kan se samband som de resultatet ger exempel på.

Med kännedom om liknande satsningar inom EU (European schoolnet, 2015; Computational thinking, 2016) samt de övergripande målet att stärka elevers digitala kompetens (Europaparlamentet, 2006) kan studiens resultat troligtvis ha samma betydelse som för svenska elever och lärare samt den undervisning de verkar i. Resultatet kan också tillsammans med Regeringens beslut (2015) och Skolverkets iscensättande (2018) ge exempel på både vägen mot utveckling av digital kompetens samt resultatet av just denna väg. Troligtvis finns flera sätt att stärka digital kompetens. Genom matematiken och programmeringen är ett sätt som visat sig vara både utmanande och givande för både lärare och elever, och denna studie ger några resultat som visar på hur implementeringen resulterar i utveckling av kunskaper och förmågor hos eleverna.

5.1.2 Resultatens rimlighet

Resultatens rimlighet kommer att diskuteras utifrån de tre kategorierna som presenterats med hjälp av tidigare forskning och egna tankar.

Är det rimligt att eleverna upplever de samband som beskrivs i kategorierna?

I en undersökning (Faag, 2018) undersöktes hur några elever i grundskolans år 1-4 uppfattade matematiska aspekter av programmering. Resultatet visar att elevernas uppfattningar kan förklaras i kategorierna *styrning*, *matematik*, *arbetsätt* och *stegvis lösningsprocess*. I denna studie är inte undersökningens syfte detsamma, men man kan tydligt se att resultatet i Faags studie (2018) visar att programmeringen

kan kopplas till matematik samt att programmeringen ses som ett arbetssätt och en lösningsprocess i steg. Att resultaten i denna studie liknar tidigare nämnda resultat, tyder på att elevernas aspekter av elevers uppfattningar både i grundskolan och gymnasieskolan liknar varandra. Samtidigt stärks trovärdigheten av resultaten då båda studier utifrån elevernas perspektiv ser likheter mellan programmering och matematik.

I en annan studie (Wärnå, 2019) med syfte att undersöka hur lärare och elever på gymnasiet upplever implementeringen av arbetet med programmeringsspråk- och bibliotek gällande svårighetsgrad, programmeringens nytta för matematiken samt undervisningens påverkan på intresse för programmering, är det många elever som både uppfattar programmeringen som svår, men också anser att den inte underlättar förståelsen för matematik. Resultaten i denna studie ger två kopplingar till de tidigare studierna av Wärnå (2019) och Faag (2018). Dels ser elever samband mellan matematiken och programmeringen Faag (2018) samtidigt som elever inte anser att programmering underlättar förståelsen för matematik (Wärnå, 2019). Samtidigt visar resultatet i denna studie hur eleverna i vissa delar av programmeringen inte gör några kopplingar till matematiken, exempelvis då det i webbutveckling handlar om design. Men att de i många andra fall kan se tydliga kopplingar till matematiken, exempelvis i begrepp och arbetssätt. Dessa resultat kan ge en förklaring till varför vissa elever uppfattar att ämnena hänger samma och andra inte, genom att se till elevernas sätt att motivera det ena eller andra påståendet. Lyfter man ytterligare ett av resultaten i denna studie, att kunskaper om samband växer fram med tiden, kan det vara så att eleverna i de andra studierna inte fått tillräckligt med tid att fundera kring deras upplevelser. Med andra ord finns en viss rimlighet i att eleverna både upplever att ämnena hör samman, samt att de i andra fall inte gör det beroende på vilka delar av ämnet man talar om.

Enligt en undersökning (Sjölander, 2018) ser grundskolelärare tydliga samband, dels mellan alla fem matematiska förmågor och programmering men också till samtliga matematiska områden vilket kan jämföras mot resultatet i denna studie. Till skillnad undersöker Sjölander (2018) grundskolelärares uppfattningar, medan denna studie utgår från gymnasieelevers upplevelser. Dock finns likheter i delar av resultaten. Gällande utvecklandet av förmågor finns en överensstämmelse resultaten emellan. Både lärarna i studien av Sjölander (2018) och eleverna i denna studie kan se samband mellan förmågorna i programmering och matematik. Lärarna kan också koppla samtliga matematiska områden till programmering, vilket skiljer sig mot elevernas uppfattningar i denna studie. Det framgår rätt tydligt att eleverna i denna studie ser en skillnad i nivå på programmeringen och den matematik de läser, vilket de anser gör den aktuella matematiken irrelevant för deras programmeringsundervisning. Min tanke är att eleverna i början av programmeringskurser inte stöter på matematik som utmanar dem just på grund av att inte störa deras utveckling av programmeringskunskaper. Detta tror jag är ett didaktiskt val av läraren som grundas i att matematiken inte ska vara hindret i programmeringen, utan att programmeringen och dess utmaningar ska få utrymme. Samtidigt ser eleverna en möjlighet i att programmeringsundervisningen i framtiden kommer att innehålla mer avancerad matematik, vilket skulle betyda att det matematiska innehåll som är kopplat till den kurs de läser, skulle få plats inom programmeringen. Med andra ord kan lärarna i grundskolan se flera samband mellan matematikens förmågor och innehåll och programmering, och säkerligen också lärarna på gymnasiet. Eleverna ser vissa samband på en lägre matematisk nivå, men

ser att de i framtiden också kan se dessa samband. Sammanfattat upplever jag att en rimlighet finns i att eleverna både kan se likheter, men också inte kan koppla all matematik till programmeringen i det stadiet de befinner sig i just nu.

Sammanfattningsvis känns de likheter och skillnader som beskrivits av eleverna och som resulterat i den första kategorin som rimlig utifrån jämförelser mot tidigare studiers resultat.

Är det rimligt att ämnena spelar en viss roll för varandra utifrån de beskrivna uppfattningarna?

Tidigare har rimligheten av skillnader och likheter diskuterats, vilket hänger samman med de roller ämnena spelar för varandra. Faags studie (2018) visar att ett samband mellan programmering och matematik kan uppfattas av eleverna, samtidigt som eleverna visar svårigheter i att definiera syftet med programmeringen (Faag, 2018). I denna studie har likheterna mellan ämnena dels förklarats i ämnens liknande karaktär och huvudsakliga syfte att lösa problem. Matematikens roll i programmeringen beskrivs bland annat genom att bistå med de matematiska nödvändigheter i form av siffror, beräkningar och begrepp som krävs för att skapa program inom programmeringen som löser uppgifter och problem eleverna stöter på i undervisningen och i vardagslivet. Omvänt är programmerings roll att bistå matematiken med program och produkter som underlättar bland annat uträkningar i matematiken. I jämförelse kan syftet definieras av gymnasieeleverna, medan grundskoleeleverna har svårare att göra det. Med andra ord kan eleverna i båda studier se sambanden mellan programmering och matematik, dock kan grundskoleeleverna inte lika tydligt som gymnasieeleverna se syftet med programmering. Att poängtera i detta fall är att uppfattningarna i denna studie utgörs av några av de intervjuade eleverna, vilket inte betyder att alla elever som intervjuats kan ge just dessa förklaringar till programmeringens syfte. Troligt är att en variation finns mellan grundskoleelevernas uppfattningar precis som den finns mellan gymnasieelevernas uppfattningar i denna studie. Som rimlig följd av detta är en viss variation mellan studiernas resultat, men som kan förklaras i elevernas olika uppfattningar om ämnens roller.

Studiens resultat visar att eleverna uppfattar att deras programmeringsundervisning bidrar till att stärka deras matematikkunskaper och förmågor. Detta liknar resultaten från Sandells studie där lärarna i undersökning anser att programmeringen bidrar på flera olika sätt, exempelvis genom att fungera som ett verktyg för att förstå matematik, utveckla elevernas matematiska förmågor eller för att skapa mer intresse för matematiken (Sandell, 2017). Det lärarna sett som potentiella utvecklingsmöjligheter i programmeringen i den tidigare studien har eleverna i denna studie också upplevt. Detta tyder på att det av lärarna beskrivna är samma verklighet som eleverna upplever, vilket gör resultaten trovärdiga.

En intressant fråga att ställa sig är varför elever i denna studie kan se syftet med programmering medan andra studier visar att eleverna har svårt att se detta (Faag, 2018; Sjölander, 2018). Sjölander (2018) lyfter att problemen ser liknande ut i andra länder enligt Sentance & Csizmadia (2015) där likheten finns gällande lärares bristande kunskap i ämnet och även elevernas svårigheter i att förstå programmeringsämnet i sig. Kanske är det lärarnas bristande kunskap som påverkar elevernas förståelse. Urvalet i denna studie, som fokuserat på elever med som läst programmering och matematik, bidrar säkerligen också till att eleverna kan se fler

kopplingar än då eleverna i studien på grundskolan enbart arbetat med en programmeringsuppgift. Troligtvis finns flera faktorer som bidrar till förståelse för programmeringsämnet, och det vore intressant att undersöka vad som bidrar till elevers förståelse för programmeringsämnet.

Sammanfattat känns resultaten trovärdiga i jämförelse med resultat från liknande studier. Dock bör hänsyn tas till de omständigheter som gett resultaten i denna studie, som utgår från elever som både läst programmering samt haft matematik- och programmeringsundervisning med lärare som är väl insatta i både programmering och matematik, mer om detta i metoddiskussionen.

Är det rimligt att upptäckten av sambanden går till på det sätt som är beskrivet?

Götling och Löfwenhamn (2018) presenterar likt Faag (2018) och Wärnå (2019) programmeringsimplementationens styrkor och utmaningar i matematiken ur elevernas perspektiv. Resultaten i dessa studier tyder dels på att eleverna genom programmeringen får möjlighet att upptäcka matematikens tillämpningsområden, att programmeringsarbetet i matematiken bidrar till omväxling i undervisningen samt att de genom programmering får en grund för framtidens digitaliserade samhälle. Det denna studie resulterat i visar på samma sak, att programmeringsundervisningen stärker eleverna i matematiken främst inom förmågor att ta sig an- och lösa problem. Detta gör att resultaten i denna studie ligger nära delar av de resultat tidigare studier genererat, vilket stärker dess rimlighet.

Även lärarna ser möjligheter i att utveckla just problemlösningsförmågan genom programmering (Arkå-Nilsson, 2018). I denna studies resultat ser vi att eleverna själva upplever att de utvecklar just problemlösningsförmågan i undervisningen. Likheten mellan möjligheter om- och de aktuella upplevelserna i undervisningen, pekar på att fenomenet bör vara rimligt då det upplevs av både lärare och elever.

En av de mest intressanta aspekterna i resultatet är huruvida programmeringen verkligen kan utveckla matematikkunskaper och tvärt om. Resultaten i studien visar att elever ser samband mellan arbetsätten i båda ämnen samt det gemensamma syftet vilket är att lösa problem. De nämner också att programmering i vissa fall inte liknar programmering, exempelvis om man tittar till betygskriterier så finns vissa skillnader, samt exempelvis att webbutveckling med fokus på design inte till större del uppfattas beröra matematik enligt eleverna. Samtidigt som det finns möjligheter finns begränsningar, vilket Sjölander (2018) påpekar i sin studie. Där presenteras flertalet andra studier (Lewish & Shah, 2012; Sung et al., 2017; Gülbahar & Kaleliouglu, 2014; Calao, Moreno-León, Correa & Robles, 2015; Benton, Saunders, Kalas, Hoyles, & Noss, 2017) vars resultat är motsägelsefulla till om utveckling inom programmering bidrar till utveckling i matematiken. Sannolikt är att eleverna i olika situationer inte upplever att de utvecklar matematiska kunskaper och förmågor för att de helt enkelt inte gör det, eller att de inte upplever det. Resultaten i denna studie visar att elever kan ha insikt i att förståelse för samband kommer senare, att man i stunden är upptagen med programmeringen och därför inte ser sambanden med matematiken förrän senare. Med andra ord verkar det vara rimligt att programmeringen har begränsningar när det gäller att utveckla matematikkunskaper och förmågor, men också att eleverna har begränsningar i vad de kan uppleva och beskriva i närtid till det i undervisningen upplevda.

Sammanfattningsvis finns flertalet tecken på att resultaten i denna studie är rimliga. Samtidigt behöver man se till programmeringens begränsningar i att utveckla matematik-kunskaper och förmågor samt till elevernas begränsningar i att utveckla och beskriva samband ämnena emellan tätt inpå undervisning som är tänkt att synliggöra dessa.

5.1.3 Resultatens begränsningar

Nedan diskuteras resultatets begränsningar för att tydliggöra dess betydelse och värde. Det resultatet inte tar hänsyn till är den variation som finns mellan både elever och lärare, främst mellan dem i studien men också i jämförelse med andra lärare och elever inom Sverige.

Som ovan nämnt finns en variation i svaren mellan eleverna i studien, men med den valda metoden analyseras data för att ge en samlad bild av samtliga upplevelser eleverna har samt likheter och skillnader däremellan. Detta betyder att resultatet inte beskriver en elev- utan flera, vilket kan ge en missvisande bild om man inte tar hänsyn till detta. På samma sätt kan lärarnas insats diskuteras. Tidigare forskning visar på en variation i lärarnas kunskaper i programmering, samt troligt är att varje lärare har både gemensamma och egna sätt att forma sin undervisning på. Detta gör att vi även i detta fall måste tänka till resultatens betydelse för just detta fall och inte det allmänna.

En annan begränsning ligger i det fokus som uppstått kring en av förmågorna, problemlösningsförmågan, istället för att lyfta samtliga förmågor. Dock kan man se tydliga inslag av andra förmågor, men som inte lyfts på samma sätt av eleverna och därför inte heller i studien.

Den största bristen i resultaten är till vilken grad dessa kan generaliseras. Informantantalet är litet och förutsättningarna unika i fråga om lärarkompetens, med andra ord är informantgruppen inte representativ för någon genomsnittlig elevgrupp med hänsyn till både kunskaper om- och undervisning i ämnena. Dock ger resultatet indikationer på att elever kan göra kopplingar mellan programmering och matematik, vilket gör studiens resultat intressanta samt motiverar fortsatta studier i liknande frågor.

5.1.4 Resultaten i förhållande till syftet

I detta delavsnitt diskuteras huruvida resultatet svarar på studiens syfte och dess forskningsfrågor. Detta diskuteras först utifrån varje forskningsfråga följt av en sammanfattning svarar på om syftet med hela studien är uppnått.

Studiens resultat ger flera exempel på samband eleverna upplever finnas mellan ämnena. De samband eleverna ser utgörs av både tydliga samband som de mellan ord- och begrepp, men också av djupare jämförelser av exempelvis arbetsgång och ämnenas syfte. Med andra ord har kunskap om vilka samband eleverna ser mellan ämnena synliggjorts i resultatet, vilket svarar mot forskningsfrågan.

Även den tredje frågan är besvarad genom elevernas beskrivning av när-, hur- och var eleverna utvecklat förmågan att se sambanden mellan ämnena.

Eleverna har också genom sina upplevelser beskrivit hur de ser att deras kunskaper och förmågor inom matematik utvecklas inom programmeringen och tvärt om – hur deras

programmeringskunskaper och förmågor utvecklas genom matematikundervisningen, vilket besvarar den andra forskningsfrågan.

Sammanfattningsvis har samtliga forskningsfrågor besvarats vilka tillsammans utgör preciseringen av syftet och därmed har syftet med studien uppnåtts.

5.2 Slutsats

Slutsatsen som kan dras är att resultaten i studien ger rimliga exempel på möjligheter och begränsningar i arbetet- och undervisning där programmering knyts samman med matematiken i syfte att utveckla elevernas digitala kompetens under förhållanden där läraren har god kännedom om båda ämnen. Genom sådan undervisning kan eleverna se samband mellan begrepp och arbetssätt samt genom det ena ämnet tillägna sig kunskaper och utveckla förmågor som gynnar dem i det andra ämnet.

5.3 Metoddiskussion

Nedan diskuteras metodval samt alternativ till den valda metoden med för- och nackdelar. På samma sätt diskuteras urval och databearbetningsmetod. Därefter diskuteras analysarbetets genomförande som följs av en sammanfattning av metoddiskussionen.

Den fenomenografiska metodansatsen valdes efter syftet vilket var att undersöka elevers olika uppfattningar. Bakom valet fanns tankar om att eleverna skulle ha olika uppfattningar om det studerade fenomenet. Denna variation skulle ligga till grund för de senare resultaten som skulle beskriva hur elever genom undervisningen kan uppfatta samband mellan ämnena. Det som också var känt sedan tidigare var att lärares kompetens i ämnena varierar samt påverkar den undervisning eleverna deltar i. Detta innebar att det troligtvis skulle finnas en stor varians mellan elevers möjligheter att se samband mellan programmering och matematik inom svenska gymnasieskolor. Därför valdes exempelvis inte en fenomenologisk ansats, då det varken är intressant eller troligt att alla elever har liknande upplevelser som kan exemplifiera kopplingar mellan programmering och matematik baserat på likheter. Med skillnader i fokus samt fokus på en detaljerad beskrivning av elevers upplevelser valdes därför den fenomenografiska ansatsen.

Eftersom fenomenet också är svårt att beskriva fullt ut, lämpar sig till exempel grounded theory inte, då resultatet av en sådan studie sannolikt hade givit en beskrivning som varit snäv och därför inte kunnat generaliserats som en övergripande teori. Ett bra alternativ hade varit att analysera redan befintliga texter om fenomenet för att synliggöra olika sätt för eleverna att se samband mellan ämnena. Det som försvårar en sådan studie är i detta fall att tidigare forskning inte behandlat just gymnasieelevernas perspektiv i fråga om programmering och matematik och sambanden mellan ämnena, vilket gör att just deras upplevelser hade blivit svåra att komma åt.

Intressant hade varit att få en bredare beskrivning genom att fler perspektiv lyfts från elevernas håll. Att ändra metodansatsen hade sannolikt inte kunnat göra någon större skillnad i detta fall då, vilket tar oss in på alternativa metodval i fråga om datainsamlingsmetoder.

Studiens resultat ger några exempel på hur elever uppfattar fenomenet. Samtidigt finns troligtvis flertalet andra uppfattningar och aspekter av fenomenet som är intressanta. Andra metodval, exempelvis enkätstudier, hade gjort att flera elevers uppfattningar hade kunnat tas hänsyn till. Dock medför detta, att elevernas upplevelser inte kan beskrivas på samma sätt som de gjorts genom intervjuer. Fördelen med intervjuerna har varit att de gett eleverna möjligheter att fritt beskriva sina upplevelser. I en enkätundersökning hade ett begränsat antal möjligheter funnits att utgå ifrån vilket hade kunnat styra eleverna att svara på ett eller annat sätt.

I efterhand med studiens resultat i hand, vore det bästa att blanda båda metoderna genom att först samla in så många olika aspekter som möjligt av elevers uppfattningar, för att sedan undersöka hur en större mängd elever förhåller sig till dessa upplevelser. På så sätt hade studien kunnat ge rikare resultat samt kunnat visa en bättre bild av hur Sveriges skolor och undervisning i matematik och programmering påverkar eleverna. Som tidigare nämnt, fanns dock inte kunskap om gymnasieelevernas uppfattning i den utsträckning att det kunnat ligga till underlag för en sådan enkätundersökning. Därför valdes intervjuer i detta fall för att fånga upp några elevers uppfattningar, där urvalet också kunde styras för att sannolikt finna så många samband som möjligt mellan programmering och matematik.

Det ovan nämnda tar oss in på urvalet, där fokus legat på att finna de elever som enligt tidigare forskning fått undervisning som sannolikt bör präglats av synliggörandet av samband mellan ämnena. Tanken var lärare med programmerings- och matematikbakgrund bör kunna ge de bästa förutsättningarna för elever att se samband mellan ämnena, då de sannolikt själva upplever att det finns samband mellan dem. Två alternativ till urval kommer att diskuteras; (1) att välja elever som läser matematikundervisning där ämnesplaner ändrats att innehålla programmering eller (2) det valda sättet att välja elever. Det första alternativet hade givit intressanta inblickar i hur programmeringsimplementeringen lyckats genom matematikundervisningen då denna utgår från den vanliga matematikundervisningen som sedan hösten 2018 ska ha inslag av programmering. Dock visar tidigare forskning på att denna implementering varit svår för gymnasielärare, vilket sannolikt kan ha påverkat undervisning och därmed eleverna. Detta skapade en osäkerhet om elever hade kunnat beskriva sina upplevelser av båda ämnen, om de endast haft en begränsad del programmering eller till och med ingen alls. Därför valdes elever som både haft programmering och matematik, samt lärare som undervisar i både programmering och matematik. På detta sätt säkerställs att eleverna har en uppfattning om vardera ämnet, samt att sannolikheten är större att dessa elever kan se samband mellan ämnena än de elever som enbart läst matematik utifrån de nya läroplanerna. Nackdelen med urvalet är att elever som uppfyller dessa kriterier finns i ett mindre antal än den andra beskrivna gruppen, vilket försvårar möjligheterna i ett mindre arbete som detta att få tillgång till informanter under kort tid.

Samtidigt som lärare valts utefter kunskaper i matematik och programmering, kvarstår frågan huruvida lärarna i sin undervisning integrerar eller separerar ämnena samt vilka didaktiska val lärarna gör. I och med att den tidigare undervisningen inte följts i denna studie, finns inget som säkerställer att undervisningen utformats med fokus på att synliggöra samband mellan ämnena. Önskvärt vore att ha flera lärare att tillgå med samma kompetens, där man också kunnat följa undervisningen för att se hur undervisningens upplägg påverkar elevernas uppfattningar om samband mellan ämnena.

När valen stod klara kring metod och urval valdes med hjälp av litteratur (Fejes & Thornberg, 2016) en passande analys, det vill säga den databearbetningsmetod som användes (Dahlgren & Fallsberg, 1991). Denna metod syftade till att synliggöra de olika delar som eleverna genom intervjuer bidragit med genom att analysera deras uttalanden. Genom att följa den givna modellen i sju steg kunde arbetet genomföras systematiskt. Denna analysmetod ger också data möjlighet att tala för sig själv genom att synliggöra dess innehåll i kategorier som framhäver både likheter och skillnader i uppfattningar. Detta går hand i hand med syftet i studien, att det är elevernas upplevelser som står i fokus, samt att dessa sannolikt skulle ha både likheter och skillnader vilka alla är relevanta för studien. Alternativt hade en interpretativ fenomenologisk analys kunnat ge liknande resultat genom att den tar hänsyn till informanternas reflektioner i en viss kontext (Fejes & Thornberg, 2016). Denna analysmetod hade likväl kunnat ge intressanta resultat, men de uppfattningar denna studie syftat till att undersöka är inte riktigt de speciella situationer som IPA-metoden syftar till att undersöka. Med andra ord lämpar sig modellen enligt Dahlgren och Fallsberg (2016) samtidigt som andra analysmetoder gett liknande resultat med mindre variationer.

Gällande analysarbetet finns två huvudsakliga frågor att diskutera. Först diskuteras noggrannheten och hänsynen som tas till detaljer i kondensationssteget samt jämförelser mellan passager, grupper och slutligen kategorier. Sedan svårigheter i att artikulera kategorier med utgångspunkt i den första diskuterade frågan.

Till en början syntes transkripten i sin ursprungliga form som ett massivt datamaterial. Vidare tillades kommentarer och passager togs ut och gavs mening, där passagera i majoriteten av fallen hade flera viktiga betydelser. Datamaterialet var nu ännu större vilket till en början till synes verkade ohanterligt. Dock gav stegen i modellen en grund i hur data skulle sorteras, vilket gjordes genom att alla passager klipptes ut och sorterades i grupper. I detta steg fick man en inblick i hur ens noggrannhet i det tidigare steget skulle spela roll för resten av analysarbetet. Eftersom materialet i början kommenterats noggrant och varje passages betydelse plockats ut, kunde nu dessa betydelser finna sin plats i de grupper som bildats. Så den första svårigheten låg i att bedöma vilka detaljer som var viktiga, och eftersom man i början inte vet vad som är viktigt, måste allt vara viktigt.

När grupper tagit form var det på samma sätt svårt att veta hur noga man skulle skilja på olika grupper. Det framkom dock i litteraturen att detta var upp till forskaren att avgöra, vilket ledde till att jag utifrån mitt syfte valde att slå samman- eller att inte slå samman grupper samtidigt som viktig information inte fick försvinna ur analysens fokus.

Detta mynnar ut i två frågor som är svåra att besvara men ändå måste diskuteras; (1) var det inledande steget tillräckligt detaljerat? Fick allt relevant för studien en plats och betydelse i analysen? (2) Var grupperingar och kategorier skapta för att på bästa sätt beskriva elevernas uppfattningar och de likheter och skillnader som fanns mellan dem?

Gällande den första frågan togs beslutet att ha med allt. Varje del av elevernas uttalanden analyserades för att försöka finna flera förklaringar och betydelser snarare än att acceptera en viss förklaring. Dock togs inga förklaringar eller betydelser ur tomma intet, snare fick transkriptet som genererats noggrant av de ljudinspelade intervjuer fungera som jämförelsematerial. På detta sätt säkerställdes så noga det går

att tolkningarna motsvarar betydelsen av elevernas utsagor. Genom detta har det yttersta gjorts för att fånga alla detaljer av ursprunglig data vilket stärker trovärdigheten i analysen.

Grupperingarna var beroende av den tidigare delen av analysen vilken diskuteras ovan. I och med det tidigare steget fanns all relevant data med, vilken återstod att gruppera. Likadant här kan kategorierna ta form på olika sätt, i detta fall utgick jag både från syftet samt från de likheter och skillnader som uppstått mellan grupperna. Det finns säkert flera sätt för de resulterande kategorierna att växa fram, och det som synes i detta arbete är att kategorierna i huvudsak liknar forskningsfrågorna. Dock finns flera betydelsefulla kopplingar mellan kategorierna, vilka pekar på att kategorierna sannolikt kunnat formats olika men att de i slutändan sammanslaget skulle innehålla samma data. Med andra ord har analysen i detta arbete fått en viss form, utifrån det syfte och de forskningsfrågor som legat till grund för hela arbetet inklusive intervjuer, databearbetning och analys. Som svar på frågan blir att kategoriseringen sannolikt kunnat presenteras i en annan form, men att betydelsen varit densamma.

Sammanfattat är metod, urval och analys anpassat efter studiens syfte. Samtidigt finns möjligheter att andra varianter av utföranden hade kunnat ge ett annat resultat som inte lika väl stämt överens med studiens syfte. Därför valdes tidigare beskrivna metod (kapitel 3) för detta arbete istället för ovan nämnda alternativ.

5.4 Framtida forskningsfrågor

Som förslag på sådant som kan utöka denna undersökning är studier som beskriver vilka förmågor och begrepp som är lämpliga att utveckla genom den ena- eller andra undervisningen. I denna studie lyfts huvudsakligen problemlösningsförmågan, men sannolikt är att andra förmågor också kan utvecklas.

Intressant är också att hitta sätt att arbeta på som förenklar- och synliggör uppkomsten och utvecklingen av att se mellan programmering och matematik, men också kopplingar till andra ämnen. Framtida studier kan exempelvis undersöka vilka arbetsmetoder som bidrar till djupa jämförelser mellan ämnena som ger konkreta idéer om hur det ena ämnet kan stödja det andra.

I denna studie är urvalet troligtvis inte representativt för den genomsnittliga elevgruppen som läser matematik med inslag av programmering. Därför motiveras studier som undersöker effekten av att implementera programmering i matematiken i den genomsnittliga elevgruppen.

Referenser

- Andersson, A. (2018). *Programmering i skolan: Vad är det enligt lärarna?* (Kandidatuppsats/Bachelor's Essay). Gävle: Akademin för Teknik och Miljö, Högskolan i Gävle. Tillgänglig: <http://www.diva-portal.org/smash/get/diva2:1211847/FULLTEXT02.pdf>
- Arkå-Nilsson, J. (2018). *Programmering som moment i gymnasie matematiken : En intervjustudie med berörda lärare.* (Kandidatuppsats/Bachelor's Essay). Halmstad: Akademin för lärande, humaniora och samhälle, Högskolan i Halmstad. Tillgänglig: <http://www.diva-portal.org/smash/get/diva2:1248488/FULLTEXT02.pdf>
- Benton, L., Saunders, P., Kalas, I., Hoyles, C., & Noss, R. (2018). Designing for learning mathematics through programming: A case study of pupils engaging with place value. *International Journal of Child-Computer Interaction*, 16(1), 68-76. doi: 10.1016/j.ijcci.2017.12.004
- Blom, K. (2019). *Programmering med MATLAB*. Stockholm: Liber.
- Calao, L. A., Moreno-León, J., Correa, H. E., & Robles, G. (2015). Developing Mathematical Thinking with Scratch An Experiment with 6th Grade Students. *Lecture Notes in Computer Science*, 9307, 17-27. doi: 10.1007/978-3-319-24258-3_2
- Collins dictionary. (2020). *Programming*. Hämtad 2020-02-12, från <https://www.collinsdictionary.com/dictionary/english/programming>
- Computational Thinking (2016). *Thinking as a computer scientist*. Hämtad 2020-02-12, från <https://ec.europa.eu/jrc/en/computational-thinking>
- Dahlgren, L.-O., & Fallsberg, M. (1991). Phenomenography as a qualitative approach in social pharmacy research. *Journal of Social and Administrative Pharmacy : JSAP*, 8(4), 150–156. Hämtad från: <http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-32776>
- Denscombe, M. (2017). *Forskningshandboken: för småskaliga forskningsprojekt inom samhällsvetenskaperna.* (Fjärde upplagan). Lund: Studentlitteratur.
- Europaparlamentet. (2006). *Nyckelkompetenser för livslångt lärande*. Hämtad 2019-11-19, från <https://eur-lex.europa.eu/legal-content/SV/TXT/?uri=celex:32006H0962>
- European Schoolnet. (2015). *Computing our future – Computer programming and coding Priorities, school curricula and initiatives across Europe*. Hämtad 2019-11-29, från http://www.eun.org/documents/411753/817341/Computing+our+future_final_2015.pdf/d3780a64-1081-4488-8549-6033200e3c03
- Faag, L. (2018). *Programmering genom de lärandes ögon: Hur uppfattar elever matematiska aspekter av programmering?.* (Kandidatuppsats/Bachelor's Essay). Jönköping: Högskolan för lärande och kommunikation, Jönköpings Universitet. Tillgänglig: <http://www.diva-portal.se/smash/get/diva2:1214736/FULLTEXT01.pdf>
- Fejes, A., Thornberg, R. (2016). *Handbok i kvalitativ analys.* (Andra upplagan). Stockholm: Liber.

- Regeringskansliet. (2016). *Vårt löfte till barnen – mer teknik i skolan*. Hämtad 2019-11-27, från <https://www.regeringen.se/debattartiklar/2016/08/vart-lofte-till-barnen--mer-teknik-i-skolan/>
- Regeringskansliet. (2015). *Programmering in på schemat i ny skolstrategi*. Hämtad 2019-11-27, från <https://www.regeringen.se/debattartiklar/2015/09/programmering-in-pa-schemat-i-ny-skolstrategi/>
- Götling, S., Löfwenhamn, O. (2018). *Programmering i matematikundervisningen: En fallstudie om utmaningar och styrkor med att programmering ska integreras i matematikundervisningen på gymnasiet*. (Kandidatuppsats/Bachelor's Essay). Stockholm: Skolan för Teknikvetenskap, Kungliga Tekniska Högskolan. Tillgänglig: <http://www.diva-portal.org/smash/get/diva2:1231335/FULLTEXT01.pdf>
- Kalelioglu, F., & Gülbahar, Y. (2014). The Effects of Teaching Programming via Scratch on Problem Solving Skills: A Discussion from Learners' Perspective. *Informatics in Education*, 13(1), 33–50.
- Lewis, C. M., & Shah, N. (2012). Building upon and enriching grade four mathematics standards with programming curriculum. I SIGCSE'12 - *Proceedings of the 43rd ACM Technical Symposium on Computer Science Education* (s. 57–62). Graduate School of Education, University of California, Berkeley. Doi: 10.1145/2157136.2157156
- Regeringen. (2015). *Uppdrag att föreslå nationella it-strategier för skolväsendet*. Hämtad 2019-11-27, från <https://www.regeringen.se/regeringsuppdrag/2015/09/uppdrag-att-foresla-nationella-it-strategier-for-skolvasendet/>
- Regeringskansliet. (2017). *Stärkt digital kompetens i skolans styrdokument*. Hämtad 2019-11-29, från <https://www.regeringen.se/pressmeddelanden/2017/03/starkt-digital-kompetens-i-laroplaner-och-kursplaner/>
- Sandell, M. (2017). *Programmering i matematikundervisningen: En studie om några gymnasielärares syn på potentiella fördelar och nackdelar för elevers lärande i matematik*. (Kandidatuppsats/Bachelor's Essay). Stockholm: Skolan för Teknikvetenskaplig Kommunikation och Lärande, Kungliga Tekniska Högskolan. Tillgänglig: <http://www.diva-portal.org/smash/get/diva2:1173064/FULLTEXT01.pdf>
- Svenska Akademiens ordlista 14 [SAOL 14]. (2015a). *Programmera*. Hämtad 2020-01-07, från <https://svenska.se/tre/?sok=programmera&pz=1>
- Svenska Akademiens ordlista 14 [SAOL 14]. (2015b). *Samband*. Hämtad 2020-01-07, från <https://svenska.se/tre/?sok=samband&pz=1>
- Sentance, S., & Csizmadia, A. (2015). Teachers' perspectives on successful strategies for teaching Computing in school. I IFIP TC3 Working Conference 2015: A New Culture of Learning: Computing and Next Generations Vilnius, Lithuania.
- Sjöberg, L. (2019). *Lärares uppfattningar om införandet av programmering i gymnasieskolans matematikämne*. (Kandidatuppsats/Bachelor's Essay). Stockholm: Skolan för Industriell Teknik och Management, Kungliga Tekniska Högskolan. Tillgänglig: <http://www.diva-portal.org/smash/get/diva2:1316923/FULLTEXT01.pdf>

Sjölander, C. (2018). *Programmering i matematikundervisningen : En kvalitativ studie kring lärares syn på hur programmering bör implementeras i matematikundervisningen i grundskolan*. (Kandidatuppsats/Bachelor's Essay). Gävle: Akademin för Teknik och Miljö, Högskolan i Gävle. Tillgänglig: <http://www.diva-portal.org/smash/get/diva2:1213396/FULLTEXT01.pdf>

Skolverket. (2016). *Redovisning av uppdraget om att föreslå nationella IT-strategier för skolväsendet – förändringar i läroplaner, kursplaner, ämnesplaner och examensmål*. Hämtad 2019-11-29, från <https://www.skolverket.se/getFile?file=3668>

Skolverket. (2017a). *Läroplan för gymnasieskolan*. Hämtad 2020-02-12, från <https://www.skolverket.se/undervisning/gymnasieskolan/laroplan-program-och-amnen-i-gymnasieskolan/laroplan-gy11-for-gymnasieskolan>

Skolverket. (2017b). *Ämne – Matematik*. Hämtad 2019-11-29, från https://www.skolverket.se/undervisning/gymnasieskolan/laroplan-program-och-amnen-i-gymnasieskolan/gymnasieprogrammen/amne?url=1530314731%2Fsyllabuscw%2Fjsp%2Fsubject.htm%3FsubjectCode%3DMAT%26courseCode%3DMATMAT03c%26tos%3Dgy&sv.url=12.5dfce44715d35a5cdfa92a3#anchor_MATMAT03c

SOU 2015:28. *Gör Sverige i framtiden – digital kompetens*. ISBN: 978-91-38-24261-2

Sung, W., Ahn, J., & Black, J. B. (2017). Introducing Computational Thinking to Young Learners: Practicing Computational Perspectives Through Embodiment in Mathematics Education. *Technology, Knowledge and Learning*, 22(3), 443-463. <https://doi.org/10.1007/s10758-017-9328-x>

Tivenius, Olle (2015). *Uppsatsens inre liv*. Lund: Studentlitteratur

Vetenskapsrådet (2017). God forskningsred. Hämtad 2019-11-12, från https://www.vr.se/download/18.2412c5311624176023d25b05/1555332112063/God-forskningsred_VR_2017.pdf

Wärnå, K. (2019). *Programmering i skolmatematiken: problemlösning eller problemskapande*. (Kandidatuppsats/Bachelor's Essay). Karlskrona: Fakulteten för Datavetenskap, Blekinge Tekniska Högskola. Tillgänglig: <http://www.diva-portal.org/smash/get/diva2:1331592/FULLTEXT02.pdf>

Informationsbrev om deltagande i forskningsstudie

Mitt namn är Daniel Östberg och jag studerar till ämneslärare i matematik och teknik. En del av min utbildning består av genomförandet av en forskningsstudie inriktad mot gymnasieskolan med inriktning på teknikämnen.

Denna undersökning syftar till att ge kunskap om hur gymnasielever på teknikprogrammet ser till sina matematik- och programmeringskunskaper, den upplevda undervisningen i ämnena och dess betydelse samt hur elever ser samband mellan programmering och matematik.

Undersökningen kommer att genomföras genom intervjuer. Uppskattat antal intervjuer beräknas bli mellan 5 och 10 stycken. Tidsåtgången för en intervju uppskattas ligga mellan 5 och 15 minuter.

Du tillfrågas härmed om deltagande i denna undersökning då studien vänder sig till elever i på teknikprogrammet som läst någon programmeringskurs samt någon matematikkurs som haft inslag av programmering. Detta för att man som elev ska kunna svara på frågor om hur man uppfattar undervisning i matematik och programmering samt hur man uppfattar ämnena i sig, samt samband mellan ämnena.

Förväntningarna på informanter är att de efter förmåga svarar på intervjufrågor. Tidsåtgången för intervjuerna beräknas maximalt ta 15 minuter och ljud kommer att spelas in. Efter att intervjun transkriberats raderas ljudinspelningen.

Undersökningen kommer att presenteras i form av en uppsats vid Mälardalens högskola som i sin slutversion läggs ut på databasen DiVA. Önskar informanten ta del av resultatet av studien kontaktar ni mig (se kontaktinformation nederst i informationsbrevet) alternativt besöker Divas portal diva-portal.org och söker efter mitt namn.

Ditt deltagande i undersökningen är helt frivilligt. Du har rätten att när som helst avbryta ditt deltagande utan närmare motivering och utan några negativa konsekvenser för dig. All data från intervjuerna kommer hanteras konfidentiellt och otillgängligt för obehöriga. Varken skola, lärare eller enskilda personer kommer att kunna spåras upp i arbetets rapport.

Ytterligare upplysningar lämnas av mig studerande eller handledare, se kontaktuppgifter nedan.

[ort], 2019-11-29

Signatur: _____

Studerande:

Daniel Östberg

[tel.]

dog15001@student.mdh.se

Handledare:

Universitetsadjunkt

Roger Andersson

[tel.]

roger.andersson@mdh.se

Intervjuguide

- **Informantnummer:**
 - **Årskurs:**
 - **Inriktning:**
 - **Tidigare kurser med inslag av programmering:**
 - **Matematikkurser:**
1. Upplever du att det finns några samband mellan dina programmeringskunskaper och dina matematikkunskaper?
 - a. Om inte, skulle du kunna nämna några av dina kunskaper som **typiska för ämnet**?
 - b. Kan du se några **likheter** mellan det du lärt det du kan i programmeringen och matematiken?
 - c. Kan du se några **skillnader** mellan det du lärt det du kan i programmeringen och matematiken?
 2. Kan du nämna något programmeringsbegrepp som du kan koppla till matematiken?
 - a.
 3. Kan du nämna något matematikbegrepp som du kan koppla till programmeringen?
 - a.
 4. Hur har du gjort dessa kopplingar/fått insikt i dessa samband? /Vad har bidragit till dem? (**När/var/hur**)
 - a. Är dessa något du **själv** kommit fram till under tid?
 - b. Är dessa något du kommit fram till **med hjälp av lärare** eller genom **undervisning**?
 - c. Är dessa något du kommit fram till genom **andra källor** så som litteratur, YouTube eller annat?
 5. Hur upplever du att dina kunskaper i programmering utvecklas i matematikundervisningen?
 - a. Hur då?
 - b. Är det några **speciella delar** av programmeringen **som utvecklas speciellt** i matematikundervisningen?
 - **Stöd:** Förmågor (Begrepp-, procedur-, problemlösning-, modellering-, resonemang-, kommunikation-, relevansförmåga)
 - **Stöd:** Centralt innehåll
 6. Hur upplever du att dina kunskaper i matematik utvecklas i programmeringsundervisningen?
 - a. Hur då?
 - b. Är det några speciella delar av matematiken som utvecklas speciellt i programmeringsundervisningen?