



Hur upplever erfarna lärare kopplingen mellan matematik och programmering?

- En kvalitativ studie om lärares synpunkter på programmering inom matematikundervisningen på lågstadiet.

How do experienced teachers perceive the connection between mathematics and programming?

- A qualitative study on teachers' views on programming in mathematics teaching at primary school.

Emil Toikkanen och Pontus Vilén

Akademien för utbildning, kultur
och kommunikation
Självständigt arbete 2 i lärarutbildningen
Avancerad nivå
15 hp

Handledare: Roger Andersson

Examinator: Pernilla Sundqvist

Termin VT23

År 2023

SAMMANFATTNING

Emil Toikkanen och Pontus Vilén

”Hur upplever erfarna lärare kopplingen mellan matematik och programmering?”

En kvalitativ studie om lärares synpunkter på programmering inom matematikundervisningen på lågstadiet.

“How do experienced teachers perceive the connection between mathematics and programming?”

Årtal 2023

Antal sidor: 28

Syftet med denna kvalitativa studie är att synliggöra förhållandet mellan matematik och programmering samt att undersöka vilka möjligheter erfarna lärare upplever med programmering inom ämnet matematik. Vi har samlat in empiriskt material för att uppnå syftet genom att intervjua fyra erfarna och behöriga lärare inom matematik och programmering. Den empiriska data analyserades med hjälp av ramverket TPACK. Resultatet visade att lärarna ser flera kopplingar mellan programmering och matematik, där algebra inkluderat mönster och stegvisa instruktioner är den mest utmärkande. En slutsats är att en multimodal undervisning ger läraren möjligheter att låta eleverna befästa både programmerings- och matematikkunskaper på ett grundläggande sätt.

Nyckelord: Programmering, matematik, algebra, multimodal, lågstadiet, TPACK.

ABSTRACT

The purpose of this qualitative study is to show the relationships between mathematics and programming and to investigate what opportunities experienced teachers perceive with programming within the subject of mathematics. We have collected empirical material to achieve the aim by interviewing four experienced and qualified teachers in mathematics and programming. The empirical data was analyzed using the TPACK framework. The result showed that the teachers see several connections between programming and mathematics, where algebra including patterns and step-by-step instructions is the most distinctive. One conclusion is that a multimodal teaching gives the teacher opportunities to let the students consolidate both programming and mathematics skills in a fundamental way.

Keywords: Programming, mathematics, algebra, multimodal, elementary, TPACK.

Innehållsförteckning

1	Inledning.....	1
1.1	Syfte och frågeställningar	2
1.2	Begreppsdefinitioner	2
1.3	Uppsatsens disposition	3
2	Bakgrund	4
2.1	Styrdokument	4
2.2	Tidigare forskning.....	5
2.2.1	Samband mellan programmering och matematik	5
2.2.2	Lärarens möjligheter med programmering i matematikundervisningen...6	
2.2.3	Faktorer som utmanar förståelsen för relationen mellan programmering och matematik	7
3	Teoretiskt ramverk.....	9
3.1	TPACK.....	9
4	Metod	11
4.1	Metodologisk ansats	11
4.2	Urval.....	11
4.3	Genomförande.....	12
4.4	Analysmetod	12
4.5	Forskningsetik.....	13
4.6	Studiens tillförlitlighet	14
4.7	Metoddiskussion	15
5	Resultat	17
5.1	Olika uttrycksformer i användandet av programmering i matematikundervisningen	17
5.1.1	Programmera varandra	17
5.1.2	Programmera med digitala verktyg.....	18
5.1.3	Konstruerande programmering	18
5.2	Programmering och dess koppling till matematikens kunskapsområden	19
5.2.1	Koppling till kunskapsområdet algebra.....	19
5.2.2	Problemlösning och stegvisa instruktioner.....	19
5.3	Möjligheter med programmering och faktorer för att lyckas med undervisningen	20
5.3.1	Väcka ett intresse hos eleverna för att möta framtiden.....	20
5.3.2	Förståelse och erfarenhet hos lärare	21
6	Analys och diskussion.....	22
6.1	Analys utifrån TPACK.....	22
6.2	Resultatdiskussion	23
6.2.1	En multimodal programmeringsundervisning.....	23
6.2.2	Programmering som en algebraisk problemlösning.....	24
6.2.3	Didaktiska val och kunskaper i programmeringsundervisningen	25
6.3	Slutsats.....	26
6.4	Relevans och framtida forskning	27
	Referenslista	28
	Bilaga 1 – Missivbrev till lärare.....	33
	Bilaga 2 - Intervjuguide.....	35

1 Inledning

De senaste decenniernas tekniska utveckling har bidragit till ett samhälle där behovet av teknisk kunskap och digital kompetens är betydande faktorer för att människor ska fungera och utvecklas (SOU 2016:85). Forskning visar att tekniska kunskaper numera tillhör allmänbildningen och i ett alltmer digitaliserat samhälle har digitala kompetenser även inkorporerats i yrkesutövningen (Manilla, 2017; Eugene, 2016; Romero, m.fl., 2019). Det ställer i sin tur krav på att skolsystemet anpassar kunskapsinnehållet till den nya arbetsmarknaden (Manilla, 2017).

I en EU rapport (Europeiska kommissionen, 2017) framgår det att mer än 90% av olika yrkeskategorier kräver någon grad av digital kompetens. Med den snabbt utvecklande digitaliseringen bestämde Regeringskansliet år 2018 att återinföra programmering i den svenska läroplanen, där det ska vara ett tydligt inslag i matematikämnet. Eleverna ska undervisas om digitala verktyg och programmering för att utveckla deras matematiska tänkande, med hjälp av metoder, resonemang och modellering fastställer Skolverket (2021). Under vår utbildning till grundskollärare i årskurs F-3 har vi undervisats och praktiserat digitala verktyg med koppling till programmering och matematik. Vi har fått resonera om programmering och dess betydelse tillsammans med ämnet matematik. Vår upplevelse är att det hade varit intressant att få en utökad förståelse om sambandet mellan programmering och matematik, samt synliggöra fördelarna som programmering kan bidra med.

Samtidigt visar forskning att det finns olika hinder för lärarna när det kommer till programmering i undervisningen. Lärarna får ingen fortbildning inom ämnet, där ett tydligt syfte mellan programmering och matematik presenteras (Kafai, 2016; Misfeldt & Ejsing-Duun, 2015; Rolandsson, 2015; Sjöberg, 2019). Lärarnas Riksförbund uppmärksammade i två enkätundersökningar att matematiklärare känner stor osäkerhet kring hur kunskapsinnehållet i matematik, som kan kopplas till hur programmering ska presenteras och undervisas (Lärarnas Riksförbund, 2016; Lärarnas Riksförbund, 2020). Baserat på detta ska den här studien belysa möjligheter och tankar som erfarna matematiklärare på lågstadiet besitter, samt hur de ser på arbetet med programmering i matematikundervisningen.

1.1 Syfte och frågeställningar

Syftet med denna studie är att synliggöra förhållandet mellan undervisning i matematik och programmering samt att undersöka vilka möjligheter erfarna lärare upplever med programmering inom ämnet matematik. Studiens syfte ska besvaras med stöd av följande frågeställningar:

- Vilka kopplingar ser erfarna lärare på lågstadiet mellan matematikämnet och programmering?
- Vilka möjligheter upplever erfarna lärare att det finns med användandet av programmering i matematikundervisningen på lågstadiet?

1.2 Begreppsdefinitioner

I detta avsnitt förklaras begrepp som är väsentliga för studien.

Algebra är ett delområde av matematiken där aritmetiska uträkningar och formella manipulationer används med hjälp av abstrakta symboler snarare än vanliga tal (Corry, 2023). I läroplanen under delområdet algebra i det centrala innehållet beskrivs vad eleverna inom ämnet matematik ska ges förutsättningar att utveckla:

Matematiska likheter och likhetstecknets betydelse. Obekanta tal och hur de kan betecknas med en symbol. Enkla mönster i talföljder och enkla geometriska mönster samt hur de konstrueras, beskrivs och uttrycks. Entydiga stegvisa instruktioner och hur de konstrueras, beskrivs och följs som grund för programmering. Hur symboler används vid stegvisa instruktioner. (Läroplan för grundskolan, förskoleklassen och fritidshemmet [Lgr22], 2022, s. 55–56).

Algoritm kan förklaras som en steg för steg instruktion av hur en uppgift kan åtgärdas (Mannila, 2017).

Kod är en konstruktion av en teknisk lösning som designas, analyseras och ändras beroende på vad resultatet blir. Genom kombinationer av programspråket tolkar och utför det digitala verktyget instruktionerna (Louden & Lambert, 2012).

Loop även kallat slinga innebär i datasammanhang en funktion som återupprepas (Nationalencyklopedin, u.å.).

Multimodalitet innebär i studien att öppna upp för att lära sig på olika vis. Det innefattar ett meningsskapande i sociala sammanhang som tar hänsyn till flera olika uttryckssätt (Edvardsson, m.fl., 2018).

Programmering definieras enligt Nationalencyklopedin (u.å.) som ”formaliserad planläggning, t.ex. av en programstyrd verktygsmaskin eller tvättmaskin. I databehandlingssammanhang innebär programmering skrivande av instruktioner för en dators arbete.”. Språket som matas in i datorn kallas för kodning (Helenius, m.fl., 2018). Programmering är en process som innebär att konstruera instruktioner för en dator på ett språk som den förstår. Den här processen kan ges i uttryck på olika sätt (Skolverket, 2018).

Scratch är ett kostnadsfritt programspråk för barn. Med stöd av verktyget kan användaren utforma digitala berättelser, spel och övriga animationer (Scratch, 2023).

1.3 Uppsatsens disposition

Studien består av 7 kapitel vilka introduceras med ett första kapitel som presenterar studiens inledning. I kapitel 2 framförs studiens bakgrund, tidigare forskning och en presentation av styrdokumentet utifrån ett nutida och historiskt perspektiv. Kapitel 3 beskriver TPACK, vilket är studiens teoretiska ramverk. I kapitlet sammanfattas de innehållande kunskaperna ramverket består av. Studiens metodavsnitt presenteras i kapitel 4, där motivering av genomförande, urval, tillförlitlighet, forskningsetik, analysmetod beskrivs och diskuteras. Resultatet presenteras i kapitel 5 utifrån den tematiska analys som använts vid framställning av de semistrukturerade intervjuerna. I kapitel 6 presenteras analysmetoden där utsagorna i resultatet analyseras utifrån ramverket TPACK. I kapitlet framförs en resultatdiskussion där studiens resultat sätts i förhållande till tidigare forskning. Avslutningsvis presenteras en sammanfattning av studiens resultat och förslag till framtida forskning.

2 Bakgrund

I kapitlet redogörs för hur programmering har tillämpats och utvecklats i skolan de senaste 40 åren. Avslutningsvis belyses hur studiens litteratursökning gått till och till det presenteras tidigare forskning inom ämnet.

2.1 Styrdokument

År 1980 tillkommer datalära i läroplanen som ett nytt område för ämnena samhällskunskap och matematik. Kunskapsutvecklingen låg huvudsakligen på ämnena snarare än programmeringsdelen, med anledning att undvika programmeringsproblem. Under tidiga 80-talet hade majoriteten av skolorna inte tillgång till datorer. I och med bristen av datorer försattes skolor istället med ett statligt stimulanspaket för att tillgodogöra det nya området. Datalära fick inget vidare genomslag då det tog lång tid för skolorna att få resurserna, vilket medförde att undervisningen istället genomfördes med papper och penna (Kilhamn, m.fl., 2021).

År 1994 avskaffade Skolverket datalära från läroplanen (Kilhamn, m.fl., 2021). Vid mitten av 90-talet minskade flera skolors undervisningstimmar för programmering. Detta till följd av lärares bristande kunskaper kring ämnet, men även för avsaknaden av integrering med övriga skolämnen. Användandet av programmering i skolan blev omotiverat då datorerna redan hade väl fungerade program och verktyg som möjliggjorde det som tidigare behövde skapas på egen hand. Under 90-talet blev det betydande för skolorna att undervisa om hur eleverna använde och sökte efter källor med hjälp av internet. Övergången till Lpo94 medförde en konkretisering i att eleverna skulle lära med datorn snarare än det som framgick i tidigare läroplaner, det vill säga att lära om datorn. Fokuset hamnade på användandet av datorer snarare än programmeringen (Mannila, 2017).

I samband med revideringen för Lgr11 som skedde 2018 kom programmering tillbaka till läroplanen, den här gången som en del av det som kallas för digital kompetens (Kilhamn, m.fl., 2021). Digital kompetens innebär att eleverna ska utveckla användandet av digitala medel, utveckla ett kritiskt förhållningssätt till information och förståelsen för digitaliseringens påverkan på samhället (Specialpedagogiska skolmyndigheten, 2023). Skolverket (2017) hänvisar till fyra aspekter av digital kompetens. De förklarar att kunskaper från programmering ska fokusera vid att utveckla elevers problemlösning i form av kodning, att låta eleverna välja olika lösningar, dokumentera och omsätta idéer till handling.

Vidare lyfter Kilhamn m.fl. (2021) att programmering nu blivit en del av kursplanerna för matematik och teknik. Undervisningen i matematik ska bland annat bygga på ett förhållningsätt där eleverna utvecklar förmågor för att kunna använda sig av digitala verktyg och programmering. De ska ligga till grund för problemlösning, matematiska begrepp och beräkningar för att tolka och presentera olika resultat (Skolverket, 2022). I det algebraiska matematikområdet ska eleverna utveckla förmågan: ”genom entydiga stegvisa instruktioner och hur de konstrueras, beskrivs och följs som grund för programmering” (Skolverket, 2022, s. 56).

2.2 Tidigare forskning

Den tidigare forskningen i den aktuella studien presenteras i tre huvudområden; Programmering och olika matematiska samband, Lärarens möjligheter med programmering i matematikundervisningen och faktorer som utmanar förståelsen mellan programmering och matematik. Den tidigare forskningen utgår ifrån vetenskapliga artiklar som har inhämtats ifrån ERIC (ProQuest) och PRIMO. De vetenskapliga artiklarna ingår i någon av följande kategorier; ”mathematics”, ”programming” och ”coding”. Sökorden som kompletterade kategorierna var; ”mathematical fields”, ”mathematical knowledges”, ”connection”, ”opportunities”, ”challenges” och ”working methods”.

2.2.1 Samband mellan programmering och matematik

Tidigare forskning visar att det finns flera samband mellan programmering och matematik (Miller, 2019; Cui & Ng, 2021; Rich, m.fl., 2020; Erümit, 2020; Wing, 2008; Calao, m.fl., 2015). Bland annat påvisar flera forskare sambandet mellan programmering och mönster (Miller, 2019; Cui & Ng, 2021; Rich, m.fl., 2020; Erümit, 2020; Calao m.fl., 2015). En forskare som fokuserat på detta är Miller (2019). Miller (2019) visar i sin studie att det lärande som sker genom kodningsundervisning leder till högre nivåer av elevers matematiska tänkande i förhållande till att urskilja matematiska mönster och strukturer. Resultatet visar även att elever som undervisas inom kodning har lättare än elever som inte undervisats i kodning att göra generaliseringar i arbetet med kodning och i den matematiska undervisningen som innefattar mönster och strukturella former (Miller, 2019).

I en annan studie av Cui och Ng (2021) framkommer det att de utmaningar elever stöter på inom programmering främst handlar om skillnaderna mellan matematiskt och datalogiskt tänkande, som innefattar resonemangsförmågan och kunskapen inom matematik och data. Forskarna visar att en del av relationen mellan dessa olika sätt att tänka, innefattar aktiviteter som inkluderar mönster inom programmeringen. Aktiviteter med mönster ger eleverna möjlighet att utforska upprepningar i matematiken samtidigt som det tränar loopar, det vill säga repetitioner inom programmeringen (Cui & Ng, 2021). Det här resultatet visar även Rich, m.fl. (2020) studie om datalogiskt tänkande inom kurs-och läroplaner för de yngre åldrarna. Resultatet visar att mönster inom datalogiskt tänkande är sambandet som är lättast att integrera i matematikundervisningen. Det påvisas i form av loopar som innefattar bestämda mönster. Resultatet av studien visar emellertid att det behöver finnas ett tydligt sammanhang mellan de matematiska aktiviteterna läraren integrerar med programmeringen. Det krävs en analys mellan matematik och programmering för att ämnesinnehållet ska kunna erbjuda en kunskap som främjar elevernas inläring av båda ämnena (Rich, m.fl., 2020).

Andra samband mellan programmering och matematik som tidigare forskning visar är problemlösningsförmåga och algoritmiskt tänkande (Erümit, 2020; Lovric 2018; Wing, 2008; Calao, m.fl., 2015; Tritrakan, m.fl., 2016). I en studie om elevernas problemlösningsförmåga och algoritmiska tänkande ämnade Erümit (2020) ta reda på vilka aktiviteter i programmeringsapplikationen Scratch som tillåter eleverna att utveckla problemlösningsförmågan och det algoritmiska tänkandet. Eleverna arbetade med Scratch kontinuerligt, det som skilde sig åt var lärarnas planering av aktiviteter i undervisningen. Resultatet visar att undervisning där eleverna får arbeta

med spel kopplat till programmering har en positiv effekt på elevernas problemlösningsförmåga och algoritmiska tänkande, vilket innebär att bryta ned problem i delar och att söka upprepade mönster i problemens utformning. Vidare visar resultatet att diskussioner om matematiska lösningar kan leda till en möjlighet där eleverna får använda sin kunskap i oprövade situationer, där förmågan att lösa problem är centralt (Erümit, 2020). Även Lovric (2018) redogör för att programmeringsverktyg är lämpliga hjälpmedel för att kunna utföra matematisk problemlösning. Forskaren anser att tid är en förutsättning för att elever tillsammans ska förstå hur algoritmerna i mjukvaror fungerar. Likt Erümit (2020) lyfter Lovric (2018) Scratch som programmeringsverktyg, där Lovric (2018) betonar det visuellt fördelaktiga med programmeringsverktyget. På liknande sätt belyser Wing (2008) hur problemlösningsprocessen har stöd av datorer och datalogiskt tänkande för att lösa, redogöra och utvärdera olika problem. Resultatet visar att datalogiskt tänkande kan utveckla problemlösningsförmågan inom matematik och forskarna anser det vara gynnsamt i och med det digitalt utvecklande samhället.

Att det finns ett positivt samband mellan kodning och såväl problemlösningsförmåga som algoritmiskt tänkande stärks även av Calaos, m.fl. (2015) studie som dessutom visar på positiva samband mellan kodning, resonemang och modelleringen. I sin studie undersökte Calao, m.fl. (2015) om kodning har en positiv inverkan på elevers matematiklärande utifrån fyra matematiska huvudområden; algoritmer, problemlösning, resonemang och modellering. Deltagarna i studien blev uppdelade i en experimentgrupp och en kontrollgrupp. Eleverna i experimentgruppen fick under tre månader arbeta med olika övningar inom programmering och algoritmer som ett komplement till den grundläggande matematikundervisningen, medan eleverna i kontrollgruppen enbart fick förhålla sig till den sedvanliga matematikundervisningen. Resultatet visar att eleverna som ingick i experimentgruppen hade utvecklat sina kunskaper och färdigheter inom alla fyra områden efter tre månaders tid. Det noterades samtidigt att eleverna som ingick i kontrollgruppen inte hade förbättrat sina resultat utifrån den vanliga matematikundervisningen. Slutsatsen som forskarna drar är att programmering i matematikundervisningen kan hjälpa eleverna att utveckla de matematiska förmågorna och att det därav måste anses ha en stor betydelse och vinst för matematikundervisningen (Calao, m.fl., 2015).

2.2.2 Lärarens möjligheter med programmering i matematikundervisningen

Tidigare forskning visar att lektionens utformning och lärares didaktiska val samt engagemang inom programmering i matematikundervisningen påverkar elevers kognitiva utveckling och problemlösningsförmåga (Yildiz Durak, 2018; Tritrakan, m.fl., 2016; Pelizzari, m.fl., 2023). I en turkisk studie ämnade Yildiz Durak (2018) ta reda på elevers strategier, innehavande motivation, reflektion och problemlösning kopplat till programmering. Resultatet av studien visar att ett arbetssätt som följer en progressionsnivå och som ligger på en passande nivå i relation till elevernas kunskaper kommer påverka elevernas engagemang och lärande positivt. Resultatet visar även att programmeringsaktiviteter som börjar i tidig ålder har positiv inverkan på barns kognitiva utveckling (Yildiz Durak, 2018). Även Tritrakan, m.fl. (2016) anser i sin studie att undervisningens utformning är en betydande faktor för elevernas lärande. Forskarna påpekar att en kreativ och genomtänkt undervisning inom programmering bidrar till att eleverna bland annat utvecklar sin

problemlösningsförmåga. Lärares olika kunskaper och betydelsen av engagemang i programmeringsundervisningen presenteras även i en studie av Pelizzari, m.fl. (2023). Där ämnade forskarna ta reda på betydelsen av vad kodning innebär för barn på förskolan. Studiens mål var att undersöka hur introduktionen av kodning och dess områden förhåller sig till barn med ett lekfullt och engagerat beteende. Resultatet visar att en kombination av lärarens tekniska kunskaper och engagemang utvecklar förmågan att identifiera problem och ger eleverna insikt om hur de löser problem på ett lekfullt och konstruktivt sätt.

Programmering förknippas ofta med digitala verktyg, men det behöver inte vara synonymt med varandra. Vilken potential programmeringsaktiviteter med skärmfria kodningsleksaker har presenteras i en fallstudie av Welch, m.fl. (2022). Forskarna avsåg följa en liten grupp barn på fem till sex år som skulle engagera sig och få inledande programmeringslektioner av sina lärare. Forskarna undersökte vilka färdigheter små barn använder sig av för att konstruera grundläggande programmering och matematik. Resultatet visar att sociala sammanhang, där eleverna lär tillsammans inom programmering påverkar barns kunskapsutveckling positivt. Detta stärks av Kjällander, m.fl. (2016) som presenterar samma upptäckt i sin studie. Vidare anser Welch m.fl. (2022) att elevernas engagemang och samarbete på programmeringslektionerna visade sig stödja deras utveckling av grundläggande mätkoncept. Forskarna menar att olika former av multimodalitet är positivt för att konstruera och kommunicera kunskap emellan eleverna och lärarna. Det här påvisar även Messer, m.fl. (2018) i sin studie och menar att elever som arbetar inom programmering med papper och penna har lika stor chans att utveckla sina matematiska förmågor som de elever som får använda digitala verktyg i programmeringen.

2.2.3 Faktorer som utmanar förståelsen för relationen mellan programmering och matematik

I en studie av Misfeldt, m.fl. (2020) framgår det att en del lärare inte kan se sambandet mellan programmering och matematik. Det uppenbarar sig däremot att lärarna, oavsett nivå på förkunskaper om programmering, kan se fördelar med det. Det här belyser likaså Mozelius, m.fl. (2019) i sin studie om lärares attityder gentemot programmering. De fann i studien att sju av åtta lärare har en positiv inställning till programmering, förutsatt att de får en fortbildning som täcker grundläggande kunskaper kring hur programmering ska undervisas. I Mozelius m.fl. (2019) studie framgår det även att lärare uppfattade programmering som ett lämpligt sätt att utveckla elevernas matematiska förmågor, mer specifikt elevers problemlösningsförmåga och logiska tänkande. Detta stärks av Kilhamn m.fl. (2021) som i sin studie klarlägger fördelarna med programmering utifrån lärarnas perspektiv. De menar att lärare ska använda programmering i syfte att hjälpa eleverna att förstå olika matematiska områden. Författarna hävdar däremot att detta kan falla i praktiken, då lärare inte förstår programmering som en del av matematiken, utan något som hamnar utanför matematikens kunskapsområde. Samtidigt påvisar Ahmed, m.fl. (2020) i sin studie att det finns en tydlighet i relationen mellan programmering och matematikämnet. Resultatet visar att programmering fungerar som ett stöd i elevernas matematikinläring om det används på rätt sätt och med en tydlig didaktisk insikt, innefattande val och metoder. Vidare argumenterar forskarna att programmeringsinslag utan insikt om elevernas

kunskaper, lärande och didaktiska val, kan resultera i att hindra eleverna i deras matematikutveckling (Ahmed, m.fl., 2020).

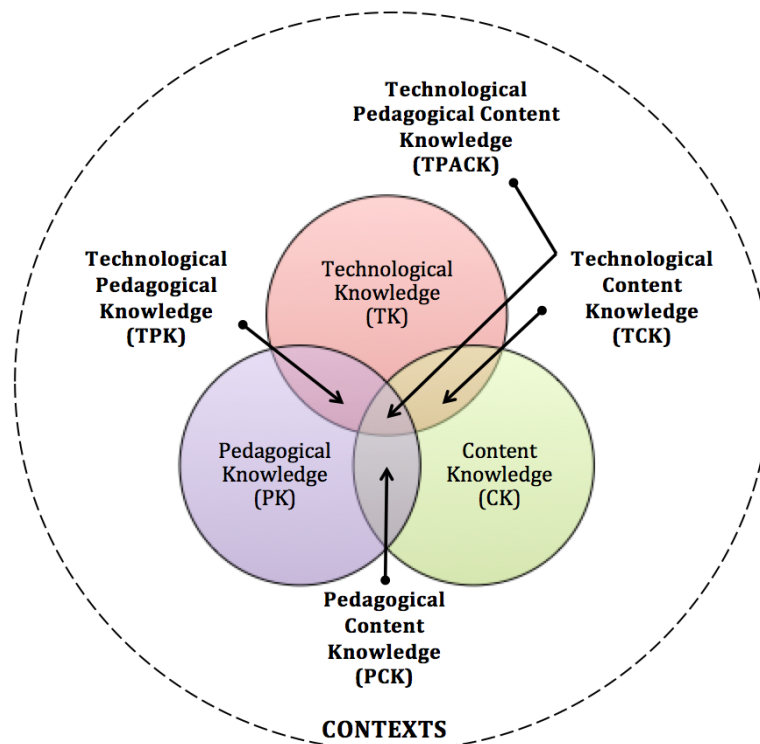
Tidigare forskning visar således att lärarens roll är viktig för att relationen mellan programmering och matematik ska bli gynnsam för eleverna (Kilhamn, m.fl., 2021; Ahmed, m.fl., 2020). Samtidigt visar tidigare forskning att lärare kan uppleva en oro kring programmering i matematikundervisningen, till följd av upplevelsen av bristande kunskaper (Sentance & Csizmadias, 2018; Rolandsson, 2015; Cederqvist, 2019). Sentance och Csizmadias (2018) undersöker i sin studie lärares relation och attityd till programmering i matematikundervisningen. Studiens resultat visar på en oro hos lärarna, som uppgav att deras egen kunskap om programmering var bristfällig, trots att de på egen hand ägnat tid åt att utveckla dessa kunskaper. Även Rolandsson (2015) påvisar i sin doktorsavhandling en oro hos lärare gällande olika inlärningsupplägg. Studien visar att bristande didaktiska kunskaper hos lärare kan vara en negativ faktor för hur eleverna lär sig om programmering (Rolandsson, 2015). En annan forskare anser att tolkningsutrymmet i styrdokumentet är en faktor till att flera lärare känner en osäkerhet kring hur programmeringsinnehållet ska presenteras utifrån förändringar och tillkommande delar i skolans styrdokument (Cederqvist, 2019).

3 Teoretiskt ramverk

Den aktuella studien utgår ifrån ramverket *Technological Pedagogical Content Knowledge* (TPACK) av Mishra och Koehler (2009). Studien innefattar även Willermarks (2018) tolkningar och analys av ramverket. Ramverket TPACK bygger på relationerna mellan de tekniska, pedagogiska och ämnesmässiga kunskaperna hos lärare (se figur A & tabell 1). Syftet med ramverket för den här studien är att synliggöra hur lärare reflekterar över de olika kunskaperna inom ramverket då kunskaperna kan innebära antingen möjligheter eller hinder beroende på om de har dem eller inte.

3.1 TPACK

Ramverket är framtaget av Mishra och Koehler (2006) och är en vidareutveckling från Shulmans (1986) ramverk *Pedagogical Content Knowledge* (PCK). Shulman (1986) ansåg att det inte räckte för lärare att vara pedagogiskt skickliga eller ha goda ämneskunskaper. Avgörande var istället om läraren kunde kombinera dessa två kunskapsområden och visa på kunnande om hur, vad och varför de lär ut ett visst ämnesområde (Shulman, 1986). Utöver de pedagogiska och ämnesbaserade kunskaperna ansåg Mishra och Koehler (2006) att ett tredje kunskapsområde som innefattade tekniska kunskaper behövde tillämpas för en mer framgångsrik och effektiv undervisning. Dessa tre kunskapsbaser och hur de samverkar och kombineras gör det möjligt att förstå lärares undervisning på ett djupare plan (Willemark, 2018).



Figur A. En beskrivning av TPACK och dess kunskaper (Mishra & Koehler, 2006).

Tabell 1. *Sammanfattning av TPACK (Willermark, 2018).*

Kunskaper	Sammanfattning
Content Knowledge	Ämneskunskap inbegriper de kunskaper läraren behöver för att möjliggöra en undervisning. Inom det specifika ämnet kan det röra sig om begrepp, fakta eller centrala teorier (Willermark, 2018).
Pedagogical Knowledge	Pedagogisk kunskap handlar om lärande och undervisning. Det inbegriper didaktiska val som belyser frågor om elevers inläring. Det innefattar kunskap om metoder och strategier angående lektionsupplägg, klassrumshantering samt strategier för att kartlägga elevers kunskaper (Willermark, 2018).
Technological Knowledge	Teknisk kunskap kan beskrivas som en förståelse för digitala verktyg och dess användningsområden. Inom ramverket fungerar kunskapen främst för att urskilja och bekräfta lärares kunskaper och progression inom den tekniska utvecklingen (Willermark, 2018).
Technological Content Knowledge	Involverar kunskap om samarbetet mellan teknisk kunskap och ämneskunskap. Det utvecklar kunskapen om hur teknik kan modelleras för bearbetning och förståelse för ett bestämt ämnesinnehåll (Willermark, 2018).
Pedagogical Content Knowledge	Innebär att den pedagogiska kunskapen och ämneskunskapen är bundna till varandra. Kunskapen handlar om hur en undervisningssituation kan knytas an till ett bestämt ämnesinnehåll och hur undervisningen anpassas efter den särskilda individens lärande, innefattande kunskaper, erfarenheter och intressen (Willermark, 2018).
Technological Pedagogical Knowledge	Innefattar kunskapen om hur didaktiska val i lärandet kan förändras när en särskild teknik används. Det inkluderar pedagogisk kunskap om möjligheter med tekniken och hur tekniska aspekter påverkar lärarens pedagogiska strategi (Willermark, 2018).
Technological Pedagogical Content Knowledge	Sammanfattningsvis utgör TPACK kunskaper om hur teknik, pedagogik ämnesinnehåll samarbetar. Det ger förståelse för hur teknik kan komplettera undervisningen som en pedagogisk strategi (Willermark, 2018).

TPACK är således ett ramverk som klarlägger relationerna mellan tekniska, pedagogiska och ämnesmässiga kunskaper. Det innefattar både individuella delar och samverkande enheter för att uppnå en effektiv och meningsfull undervisning, där teknik sätts i förhållande till undervisningssituationen som ett pedagogiskt tillvägagångssätt (Willermark, 2018). För att programmeringen i matematikundervisningen ska bli givande och meningsfull är den beroende av att lärarna både besitter en ämneskunskap av matematik och programmering, samt en pedagogisk kunskap där metoder och kunskap ur ett nyanserat och produktivt sätt ligger till grund för elevernas utveckling. Den tekniska aspekten krävs för att lärarna ska kunna modellera och visa användningsområden för verktyg som utvecklar och befäster kunskapen hos eleverna (Kim & Lee, 2018). TPACK (Mishra & Koehler, 2006) och Willermarks (2018) tolkning av ramverket används i den aktuella studien för att förstå och tolka den insamlade empirin. Hur empirin har samlats in och vilka metodologiska val har gjorts redogörs för i nästa kapitel.

4 Metod

Det aktuella kapitlet tar upp metoderna som legat till grund för studien. Kapitlet inleds med att åskådliggöra den metodologiska ansatsen, urvalet, genomförandet och analysmetoden. Därefter diskuteras forskningsetiken, studiens tillförlitlighet samt studiens metodologiska val.

4.1 Metodologisk ansats

Data samlades in genom en kvalitativ forskningsmetod i form av semistrukturerade intervjuer. I den kvalitativa undersökningen har åsikter, intryck och synpunkter varit centrala. Totalt genomfördes fyra semistrukturerade intervjuer med lärare som undervisar på lågstadiet. Vi använde oss av semistrukturerade intervjuer i form av förutbestämda frågor som tillät respondenten att uttrycka sig fritt och komma med egna tankar och reflektioner. Kompletterande följdfrågor ställdes för att få en tydligare uppfattning av respondenternas svar. Metoden valdes för att få fördjupade motiveringar av respondenterna. Semistrukturerade intervjuer är en lämplig metod eftersom frågorna kan anpassas utefter situationen, samt förtydligas utan att intervjun avviker från det aktuella ämnet (Håkansson, 2021). Ytterligare fördelar med semistrukturerade intervjuer är att respondenten känner sig trygg i sammanhanget, då det upplevs mer som ett samtal än ett förhör (Denscombe, 2018).

4.2 Urval

De respondenter som har medverkat i de semistrukturerade intervjuerna har valts ut i enlighet med det bekvämlighetsurval som David och Sutton (2016) lyfter. David och Sutton (2016) redogör för att ett bekvämlighetsurval innebär att forskaren väljer ut respondenter som uppfattas lämpliga till den specifika studien utifrån egen förtroenhet. Studien baseras även på målstyrda urval, mer specifikt ett icke-sekventiellt tillvägagångssätt som innebär att respondenterna är valda utefter olika kriterier för att kunna besvara studiens forskningsfrågor (Bryman, 2016). Respondenterna arbetar på olika skolor men är verksamma inom samma kommun. Urvalet limiterades till legitimerade grundskollärare som tjänstgjort i minst 7 år (se tabell 2). Lärarna ska under dessa år ha undervisat i matematik och programmering. Dessa krav var av relevans eftersom vi ville att informanterna skulle ha kunskap om ämnet matematik och dess olika kunskapsområden, samt en arbetslivserfarenhet av yrket. Respondenterna i studien har vidtagit någon form av utbildning inom programmering och har via eget intresse eller fortbildning eftersträvat nya kunskaper inom området. Till följd av kraven på kunskap och erfarenhet är antalet respondenter i studien begränsande.

Tabell 2. En beskrivning av respondenterna i studien. Hur många år de har varit verksamma inom läraryrket efter sin examen, vilken klass de undervisar i och deras beteckning i studien.

Respondent	Verksam (år)	Nuvarande klass (åk)	Beteckning
1	25 år	Åk 1	Lärare A
2	23 år	Åk 1,2 & 3	Lärare B
3	13 år	Åk 1	Lärare C
4	16 år	Åk 2	Lärare D

4.3 Genomförande

Ett missivbrev (se bilaga 1), där information om studien framkom, skickades ut till fyra lärare som skulle delta i intervjuerna. Tillsammans med missivbrevet fick respondenterna även ta del av en utformad intervjuguide (se bilaga 2), vilket i enlighet med Bryman (2018) bidrar till ett mer utvecklat samtal under intervjun. Författaren anser att respondenternas trygghet ökar till följd av att de får ta del av information om intervjun i förväg. Den semistrukturerade intervjuguiden bestod av totalt sjutton frågor med utrymme för eventuella följdfrågor. Av de sjutton frågorna var fyra stycken bakgrundsfrågor om respondenten. David och Sutton (2018) påpekar att en semistrukturerad intervjuguide som har utrymme för följdfrågor, möjliggör en mer kreativ uppfattning av forskarna.

Videokonferenstjänsten Microsoft Teams användes för att genomföra intervjuerna. Intervjuerna varade mellan 30–40 minuter. Båda studiens författare närvarade vid samtliga intervjuer, vilket ökar studiens tillförlitlighet och minskar risken för feltolkningar. Intervjuerna inleddes med att fråga om respondenten godkände att intervjun spelades in, med syftet att inte utelämna viktig information. Med hjälp av Teams inspelningsfunktion sparades samtalet ned för att användas till transkriberingen. David och Sutton (2016) påpekar att inspelade intervjuer är fördelaktigt då det fångar dialogen fullt ut.

4.4 Analysmetod

För att analysera de fyra semistrukturerade intervjuerna tillämpades en tematisk analys. I enlighet med Bryman (2018) används den tematiska analysmetoden med syftet att söka efter gemensamma kategorier i data. Efter intervjuerna var genomförda transkriberades materialet. Studiens skribenter transkriberade två intervjuer vardera. Efter transkriberingen av de fyra intervjuerna och en diskussion kring det insamlade materialet färgkodades kategorier utefter respondenternas svar, utifrån liknande och avvikande svar. Kategorierna identifierades utifrån studiens empiri och sammanställdes därefter i form av tre teman med avsikt att besvara studiens syfte och forskningsfrågor (se tabell 3). De teman som sammanställdes presenteras i följande underrubriker; Olika uttrycksformer i användandet av programmering i matematikundervisningen, Möjligheter med programmering och faktorer för att lyckas med undervisningen samt Programmering och dess koppling till matematikens kunskapsområden.

Tabell 3. Exempel på den tematiska analysmetodens tillvägagångssätt. Hur utsagorna blev till kategorierna och hur kategorierna slutligen blev till ett tema.

Utsagor	Kategorier	Tema
<p>”Vi gör att de får programmera mig, alltså vad ska jag göra? Gå framåt, hur långt? Jag brukar gå in i väggen och in i bord så det skrattar så de dör liksom, lite sånt [...] (Lärare A).”</p> <p>”Oftast gör jag det ute att de får programmera varandra. Blir mer som en lek, en robot och en programmerare [...] (Lärare C).”</p>	<p>Eleverna programmerar varandra i programmeringsundervisningen</p>	
<p>”Jag har använt till exempel färger och former bara där eleverna ska lägga ett mönster som de skulle utföra själva. Till exempel. Vad händer när du hamnar på en röd lapp? Och vad händer när du hamnar på en gul lapp? [...] (Lärare D)”</p> <p>”[...] att se sekvenser i mönster, vart börjar det om? Alltså vilken bit är grunden och vart det börjar om. Mönster i f-1 och uppåt så är det ju stegvisa instruktioner och också att man ritat instruktioner med symboler, pilar och paustecken (Lärare A).”</p>	<p>Praktisk programmeringsundervisning med händerna</p>	<p>Olika uttrycksformer i användandet av programmering i matematikundervisningen</p>
<p>Vi jobbar med Scratch och Bluebots och Beebots. Det finns en app till Bluebots så man kan styra den med en iPad, så det är väl det som är variationen och progressionen egentligen, att göra längre kodningar för att få den att skriva typ en mening (Lärare A).”</p> <p>Scratch junior har de jobbat med en del vet jag och ibland får vi låna sådana där Bluebots (Lärare B).</p>	<p>Digitala verktyg i programmeringsundervisningen</p>	

4.5 Forskningsetik

För att uppnå forskning av god etisk karaktär bör forskare ta hänsyn till de fyra forskningsetiska principerna (Tivenius, 2015). Den aktuella studien tar hänsyn till dessa vilka är; informationskravet, samtyckeskravet, konfidentialitetskravet och nyttjandekravet.

Informationskravet handlar om att forskaren har en skyldighet att informera villkoren och rättigheterna till informanterna (Tivenius, 2015; Vetenskapsrådet, 2017). Informationskravet tillgodosågs genom att missivbrevet i förväg skickades ut till informanterna. Informationen om deltagandet upplystes i missivbrevet och

repeterades vid intervjun. De informerades om att deltagandet var frivilligt och att de utan motsättningar eller konsekvenser kunde välja att avstå sitt deltagande.

Samtyckeskravet innebär att informanterna ger tillåtelse för sitt deltagande i studien (Tivenius, 2015; Vetenskapsrådet, 2017). Samtliga respondenter tillkännagav sitt samtycke innan intervjun genomfördes.

Studien har även tagit hänsyn till konfidentialitetskravet vilket innebär att namn på informanten eller arbetsplatsen inte kommer tas med utan tillstånd, samt att materialet som samlats in kommer att hanteras konfidentiellt (Tivenius, 2015; Vetenskapsrådet, 2017). Det här framkom vid delgivandet av missivbrev till respondenterna.

Slutligen innebär nyttjandekravet att de uppgifter som informanterna uppger enbart kommer användas till den aktuella studien och inget annat (Tivenius, 2015; Vetenskapsrådet, 2017). Det insamlade materialet kommer efter godkänd uppsats att raderas. Respondenterna informerades om att studien kommer publiceras offentligt på DIVA-portalen där andra kan ta del av den, innan intervjuerna genomfördes.

4.6 Studiens tillförlitlighet

Under denna del presenteras studiens tillförlitlighet i förhållande till den kvalitativa forskningsansatsen. I enlighet med Bryman (2018) benämns de grundläggande kriterierna vid värdering och bedömning av kvalitativa studier som tillförlitlighet och äkthet. Tillförlitlighet delas in i fyra delkriterier, *trovärdighet*, *överförbarhet*, *pålitlighet* och *möjlighet att styrka och konfirmera*, vilka motsvarar de kriterier som används vid kvalitativ forskning.

Det första delkriteriet för bedömning av tillförlitlighet är *trovärdighet*, vilket i enlighet med Bryman (2018) beskrivs som att studiens resultat ska spegla den givna verkligheten. Detta innefattar att studiens respondenter ska känna vid den information och de resultat som studien presenterar. För att uppfylla detta kriterium fick respondenterna under intervjuernas slut möjlighet att komplettera med eventuella förtydliganden för att undvika missförstånd.

Det andra delkriteriet är *överförbarhet*, vilket enligt Bryman (2018) innefattar huruvida en studies forskning är överförbar till en annan verklighet eller miljö. Bryman (2018) menar att en läsare kan söka sig till information och beskrivningar i resultatet, som stämmer överens med dennes egen miljö. Överförbarheten i den här studien stärks genom en konkret beskrivning av tillhandahållandet av den insamlade data. Resultatet presenteras genom tydliga redogörelser av respondenternas svar vilka läsaren kan söka sig till för att jämföra med dennes egen miljö.

Pålitlighet är det tredje delkriteriet för tillförlitlighet och förklaras i enlighet med Bryman (2018) som motsvarigheten till reliabilitet. Delkriteriet innefattar granskningar och redogörelser av forskningens samtliga delar, vilket kan genomföras av kollegor. För att uppfylla delkriteriet granskas arbetet av kurskamrater och handledare, för att slutligen låta en examinator godkänna arbetet. För att möjliggöra granskning av studiens samtliga delar, och därigenom öka studiens pålitlighet, har det varit centralt att tydligt och transparent presentera studiens metodval och

genomförande. Detta har gjorts under kapitlet metod, med tillhörande underrubriker.

Det fjärde och sista tillförlitlighetskriteriet är *möjlighet att styrka och konfirmera*. Enligt Bryman (2018) ska forskaren utom rimligt tvivel ha handlat med god tro, och att inga teoretiska inriktningar eller personliga värderingar ska ha påverkat studiens utförande och resultat. I enlighet med detta delkriterium har studien utom rimliga tvivel genomförts utan ställningstaganden som påverkat studiens färdigställda verk. Studiens författare har genom hela processen strävat efter att vara objektiva och opartiska. På grund av den kvalitativa forskningens natur är det emellertid omöjligt att helt utesluta att viss tolkning sker.

4.7 Metoddiskussion

Studiens syfte är att synliggöra förhållandet mellan programmering och matematik på lågstadiet samt att undersöka vilka möjligheter erfarna lärare upplever med programmering inom ämnet matematik. Forskningen grundades i en kvalitativ forskningsansats, med en kvalitativ analysmetod, vilket vi ansåg gav oss rätt förutsättningar till en djupare förståelse för studiens forskningsfrågor. Semistrukturerade intervjuer innehållande både öppna och direkta frågor gav oss möjligheten att samla in lärarnas beskrivningar och tankar om kopplingen mellan programmering och matematik samt hur lärarna arbetar med programmering i matematikundervisningen. Denscombe (2018) anser att semistrukturerade intervjuer ger en möjlighet för respondenten att utveckla sina idéer utförligare om det aktuella ämnet.

Valet att använda två olika urvalsstrategier, det vill säga bekvämlighetsurval och målstyrda urval, motiveras genom att studien hade som mål att få in flertalet respondenter. Detta för att i sin tur kunna samla in ett rikt empiriskt material. Det målstyrda urvalet utgick ifrån en kravbild som respondenterna skulle uppfylla för att väljas ut att delta i studien. Kravbilden bestod av olika kriterium inom kunskap och erfarenhet som ansågs väsentligt för studiens syfte och forskningsfrågor. Valet av erfarna lärare motiveras genom att studien fokuserar på möjligheter med programmering och dess koppling till matematikämnet. Lärare med kort yrkeserfarenhet har eventuellt inte reflekterat över dessa samband på liknande sätt. Under den begränsade tid som studien genomfördes inom hittades fyra respondenter som uppfyllde kravbilden och som var villiga att delta i studien. Ett mer omfattat urval, där fler lärare och skolor från olika kommuner var representerade, hade kunnat antas öka studiens tillförlitlighet. Detta då det empiriska material som studiens analys grundas på hade varit rikare.

Skribenterna genomförde alla fyra intervjuer tillsammans. Det gav möjlighet för ett bredare dataunderlag, då två personer istället för en kunde formulera och ställa följdfrågor till lärarnas redogörelser. Att vara två vid intervjutillfället minskar även risken för feltolkningar, vilket i förlängningen ökar studiens tillförlitlighet. Samtidigt finns det en svårighet att skapa fullt liknande intervjuer i form av semistrukturerade intervjuer med följdfrågor (Denscombe, 2018), vilket vi upplevde under intervjuerna med respondenterna. Det här kan ha påverkat och begränsat studiens tillförlitlighet. Efter intervjuerna transkriberade vi två intervjuer vardera på grund av tidsåtgången som följer. Om båda studiens författare hade transkriberat alla intervjuer var för sig och därefter jämfört transkriberingarna med varandra, så hade studiens

tillförlitlighet ökat. I efterhand har diskussionen handlat om hur kompletterande observationer hade kunnat tillföra en tydligare bild utifrån studiens syfte och frågeställningar, vilket i sin tur hade stärkt studiens tillförlitlighet. Vi anser att observationer främst hade kunnat ge oss en tydligare bild av hur respondenterna använder sig av programmering i matematikundervisningen och att det därav hade gett ett tydligare underlag till att belysa och diskutera både möjligheter och kopplingar mellan programmering och matematik.

Inom kvalitativ forskning finns en kontrovers kring huruvida resultaten går att generalisera (Denscombe, 2018). Det anses problematiskt att generalisera resultaten till andra miljöer då enstaka intervjupersoner inte utgör en representativ urvalsgrupp (Denscombe, 2018; Bryman, 2018). Med tanke på att det här är en mindre studie med fyra respondenter kan inte urvalet anses som representativt för en hel yrkesgrupp. Generaliserbarhet inom kvalitativ forskning menar Denscombe (2018) kräver ett annat angreppssätt då det vanligtvis innefattar en mindre population. Det angreppssättet kallas för överförbarhet. Överförbarhet handlar om att läsaren själv får göra en bedömning om huruvida studiens resultat kan tillämpas på liknande fall. För att läsaren ska kunna göra den bedömningen har kriterierna för den aktuella studiens urval presenterats ingående.

5 Resultat

I detta kapitel presenteras resultatet från de intervjuer som genomfördes med de fyra lärarna som deltog i studien. Resultatet redovisas i de teman som framkom vid den tematiska analysen av empirin; Olika uttrycksformer i användandet av programmering i matematikundervisningen, Möjligheter med programmering och faktorer för att lyckas med undervisningen och Programmering och dess koppling till matematikens kunskapsområden. Ett antal underteman har sedan formulerats till de olika teman för att organisera lärarnas svar. Dessa har inspirerats utifrån de identifierade kategorierna i den tematiska analysen.

5.1 Olika uttrycksformer i användandet av programmering i matematikundervisningen

I intervjuerna framgick det att programmeringen i matematikundervisningen varierade mellan de olika lärarna. Resultatet visade att lärarna använde olika uttrycksformer av programmering för att utveckla det matematiska kunskapsinnehållet för eleverna, samt för att bidra med en differentierad undervisning.

5.1.1 Programmera varandra

Lärare A förklarade att hen använde sig av en övning där eleverna får ”styra” läraren eller varandra via röstkommandon, så att personen kunde utföra olika moment. Lärare A förklarar hur stegvisa kommandon är delar för att programmering ska fungera, eftersom instruktionerna måste följa en korrekt ordning:

Vi gör att de får programmera mig, alltså vad ska jag göra? Gå framåt, hur långt? Jag brukar gå in i väggen och in i bord så de skrattar så de dör liksom, lite sånt. Den är jätterolig, att de får programmera mig eller varandra, för det blir ju fel, då kommer man in på det här med buggar och det är sånt som händer när saker och ting inte fungerar, att det blir bugg i programmeringen, inte att den går sönder, utan att det är något som har blivit fel (Lärare A).

Lärare B belyser att eleverna behöver reflektera över sina instruktioner, utifrån en vardaglig situation. Tillsammans behöver eleverna komma överens om korrekta steg i situationerna:

Jag gjorde för ett tag sedan, att de skulle programmera mig. Jag skulle bre en smörgås och jag menar, säger de åt mig att ta en klick smör och jag tar ner hela handen i smörpaketet, då blir det protester. Så där får man inte göra, jag menar ni måste tala om för mig vad jag ska göra då, ni ska ju programmera mig och berätta vad jag ska göra (Lärare B).

Lärare C beskrev vikten av att plocka ner programmeringsundervisningen på en grundläggande nivå. Eleverna får testa på att utföra programmering samt att bli programmerade av varandra. Läraren beskriver aktiviteten som en lek, där eleverna innehar olika roller för att möjliggöra en lektion utifrån instruktioner:

Oftast gör jag det ute att de får programmera varandra. Blir mer som en lek, en robot och en programmerare. Och så ska de knacka liksom tre gånger på ryggen en gång på

axeln, sedan kanske två gånger till och sen startar de genom att trycka på huvudet och så ska personen gå i raka linjer liksom. Det blir som en liten lek också och stöter de på ett hinder (Lärare C).

5.1.2 Programmera med digitala verktyg

Resultatet av intervjuerna visade att samtliga lärare integrerade digitala verktyg i arbetet med programmering i matematikundervisningen. De digitala verktygen som framhölls var robotarna Bluebots och Beebots samt olika applikationer till surfplattor. Syftet var att erbjuda eleverna ytterligare arbetsätt för att differentiera undervisningen. Samtliga lärare berättar:

Vi jobbar med Scratch och Bluebots och Beebots. Det finns en app till Bluebots så man kan styra den med en iPad, så det är väl det som är variationen och progressionen egentligen, att göra längre kodningar för att få den att skriva typ en mening (Lärare A).

Scratch junior har de jobbat med en del vet jag och ibland får vi låna sådana där Bluebots (Lärare B).

De har sådana här Bluebots (Lärare C).

Vi har använt en del appar där man har fått skriva sin egen bana och hur en gubbe ska gå. Även konstruera egna loopar och så vidare (Lärare D).

5.1.3 Konstruerande programmering

Utöver att programmera varandra och med hjälp av digitala verktyg, uttryckte även lärarna hur de arbetade med att konstruera och tillverka olika former av mönster, samt att följa stegvisa instruktioner. Gemensamt ansåg lärarna att undervisningens utformning spelar roll i utvecklandet av olika matematiska förmågor. Lärare D beskrev hur eleverna får arbeta praktiskt i form av att förstå olika instruktioner med hjälp av färger. Hen beskriver vidare hur färgkoderna representerar olika rörelser som eleverna ska genomföra:

Jag har använt till exempel färger och former bara där eleverna ska lägga ett mönster som de skulle utföra själva. Till exempel. Vad händer när du hamnar på en röd lapp? Och vad händer när du hamnar på en gul lapp? Och vad händer när du hamnar på en grön lapp och så vidare. De ska rotera eller hoppa eller vad de nu ska göra bara för att förstå själva grunden i det (Lärare D).

Lärare B beskrev kopplingen mellan programmering och matematik som en större del än enbart den digitala aspekten. Hen lyfter även att programmering är praktiskt arbete utan digitala verktyg, där eleverna bland annat kan konstruera pärlhalsband utifrån ett visst mönster:

[...] ja, då blir det ännu svårare med programmering. Ja, så är det ju. För att många tror ju att programmering är mycket Bluebots eller Scratch eller sånt där, att bara trycka på knappar, framåt och bakåt och så vidare. Jag tycker att programmering är så mycket annat, som med mönster. Man ser ju hela matematiken är ju mönster, ta multiplikationstabellen, det är ju också mönster. Det är mycket mönster, men inte bara att man ska måla ett pärlhalsband efter ett visst mönster, utan det är även talmönster och talrader på olika sätt (Lärare B).

Även lärare A beskriver hur hen arbetar praktiskt i form av att rita olika instruktioner, utan digitala verktyg, innefattande mönster som en del av programmeringen i matematiken:

Jag jobbar jättemycket med mönster, att bygga mönster, att fortsätta mönstret, att se sekvenser i mönster, vart börjar det om? Alltså vilken bit är grunden och vart det börjar om. Mönster i f-1 och uppåt så är det ju stegvisa instruktioner och också att man ritat instruktioner med symboler, pilar och paustecken (Lärare A).

5.2 Programmering och dess koppling till matematikens kunskapsområden

Samtliga lärare reflekterade över kopplingen mellan programmering och matematikämnet. Resultatet visade att de tydligaste kopplingarna lärarna gör mellan programmering och matematik är inom kunskapsområdena problemlösning och algebra som innefattar mönster, talföljder och stegvisa instruktioner.

5.2.1 Koppling till kunskapsområdet algebra

Lärare A redogör för vad en loop innebär i ett programmeringssammanhang och lyfter upp vikten av en begreppsförståelse. Läraren berättar även om sina uppfattningar om kopplingen mellan programmering och det matematiska området algebra, där bland annat mönster och talföljder ingår. Mönster är enligt lärare A ett område som berör flera delar av matematikens kunskapsområden:

Naturligast för mig är just när man jobbar med mönster och slingor. Att det återkommer, att det kommer loopar, att det repeteras, att det startar från början. Alltså såna saker och då kommer ju också språket in på en nivå som mina elever förstår. Ja, men igenom mönster. Mönster finns ju i allt egentligen, det finns ju när man jobbar med naturliga tal och alltså i talföljder och när man jobbar med hundrurutan. Att man hittar mönstren nedåt och åt sidan, det finns ju på många sätt. Man har ju jobbat med mönster jämt. Det är lite så vårt positionssystem är uppbyggt också, att se mönster, vad som händer i ental, tiotal, hundratal (Lärare A).

Lärare D uppger betydelsen mellan programmeringens utförande och skapandet av mönster i matematikundervisningen. Hen drar paralleller till stegvisa instruktioner som krävs för att eleverna ska kunna genomföra en uppgift:

Ja, då tänker jag mera det här. Hur en uppgift ska utföras, alltså, vad ska jag göra för att utföra en uppgift? Och det är samma sak i programmeringen, vad är det jag behöver göra för att jag skulle kunna utföra uppgiften. Eller för att göra det här mönstret? Det hänger ju ihop lite grann (Lärare D).

5.2.2 Problemlösning och stegvisa instruktioner

Problemlösning visades vara det andra kunskapsområdet där lärarna hade enklast att se sambandet till programmering. Flera lärare ansåg att det lätt blir fel i momenten där eleverna ska programmera varandra eller programmera ett tekniskt verktyg. Lärarna beskriver hur eleverna lär sig av sina misstag vid programmering. Lärare C beskriver att programmering är en rad olika instruktioner där alla delmoment behöver överensstämna i instruktionen för att det ska fungera:

De felen och det som du gör rätt, det kommer ju liksom hända att du måste backa tillbaka och titta och det är det här med problemlösning. Var gjorde jag felet, för jag hör ju många som säger den går inte dit eller de här pärlorna räcker inte till mönstret. Något har blivit fel och du måste undersöka just vad för något. Man kan inte bara kasta in handduken och säga att det inte går. Backa tillbaka och gör det stegvis och var noggrann för annars blir det ju såna grejer. Just att dina fel gör ju att det blir fel på slutet (Lärare C).

Lärare A tycker problemlösning är en central del i programmering. Hen anser att programmeringsspråket blir ett språk som både behandlar algebra och som tydligt går att koppla till stegvisa instruktioner.

Det här med stegvisa instruktioner, för att jag menar att får du en ekvation så behöver du lösa den steg för steg. Så blir det även med problemlösning, läs texten, vad får du veta, vad kan du använda dig av. Där blir det en sorts, en sorts koppling till programmering, att du just använder dig av stegvisa instruktioner. Sen är det ju det här med att använda och uttrycka symboler istället för siffror. Att språk är mer än bara bokstäver och ljud. Ett språk kan vara massor och det gör vi med siffror, men även att vi i årskurs 1 och förskoleklass, så kan vi skriva blomma + blomma = 4 (Lärare A).

5.3 Möjligheter med programmering och faktorer för att lyckas med undervisningen

Majoriteten av lärarna såg programmering som ett viktigt inslag i matematikämnet. De beskrev olika möjligheter med programmering i matematikundervisningen samtidigt som de reflekterade över vad som kan vara problematiskt, och vad lärare behöver tänka på, för att lyckas med programmering i matematikundervisningen.

5.3.1 Väcka ett intresse hos eleverna för att möta framtiden

Lärare A påvisade att en differentierad undervisning är en faktor för att väcka elevernas intresse och skapa motivation till arbetet med matematik och programmering:

[...] ja men absolut, för varierad undervisning ska det ju vara. Att variera undervisningen får våra elever, det vet vi ju skapar motivation (Lärare A).

Även lärare C betonade fördelarna av att väcka ett intresse för eleverna tidigt inom digitala verktyg och programmering:

En fördel är att de får bekanta sig med det redan på lågstadiet. Man kan ju väcka ett intresse också för de som inte stött på det och låta dem få prova på det. De kanske kan bli intresserade av det (Lärare C).

Ett utvecklande av den digitala kompetensen och förståelsen för hur samhället digitaliseras är avgörande för att lärarna ska lyckas med sitt uppdrag. Lärare B beskriver:

[...] ja, alltså det här när man tänker programmering. Det är något som det bara blir mer av i framtiden. Jag menar de här barnen, när de blir vuxna och ska ut i sitt arbetsliv, då kommer programmering vara en ren och klar självklarhet i nästan vad och än de ska jobba med (Lärare B).

Lärare A beskriver vikten av att erbjuda eleverna grunderna av en digital kompetens för att eleverna ska ha rätt förutsättningar till att möta framtiden:

Den generationen som jag har nu kommer möta en helt annan framtid än den vi försöker lära dem. Så då måste de på något vis ha basen med sig, det som man tror kommer att fortsätta i det här med programmering, teknik och datalogiskt tänkande (Lärare A).

Även lärare D belyser att det är av betydelse att eleverna ges förståelse om grundprinciperna inom programmering och matematik samt att de utgör ett tydligt syfte:

[...] de här grundprinciperna tänker jag. Även varför ska vi programmera och att de får med sig syftet med det (Lärare D).

5.3.2 Förståelse och erfarenhet hos lärare

Vidare visar resultatet att lärarna ansåg det vara viktigt att implementera programmeringen i matematikundervisningen. Lärarna resonerade om olika faktorer som kan vara hinder för lärare att koppla programmering med matematiken. Lärarna påpekade att de upplever att en del lärare har en bristande förståelse för kopplingen mellan programmering och matematik. Lärare A förklarar vad som kan utveckla en negativ självkänsla hos lärare i arbetet med programmering:

Bristande kunskap i programmering och en oro att göra fel, för känner du inte dig trygg i dig själv, att du vet vad du håller på med, då skapar det ju en jätteoro. De här som är 60+ idag kanske inte riktigt har motivationen att läsa på och lära sig, fast det står i läroplanen. Då kanske man nöjer sig med att använda Bluebots och leka lite, sen känner man att man har arbetat med programmering (Lärare A).

Lärare C funderade över om att vissa lärare inte ser programmering som en självklar del i matematiken:

[...] man får tänka till ordentligt att vad har programmering med matematiken att göra egentligen. Programmeringen är liksom inte självklar att den ska ingå i matten utan den kanske skulle vara som ett eget ämne eller ingå i teknikämnet (Lärare C).

Lärare A reflekterade även om att lärares intresse för programmering och digitala verktyg är en faktor för att lyckas med en meningsfull undervisning inom programmering i matematikämnet. Lärare A beskriver hur hen har utvecklat sitt intresse med digitala verktyg:

Mitt intresse har ju alltid funnits. Jag var en av de första lärarna som började jobba med iPads när dem kom 2011. Då kom ju allt det här med alla appar och jag kommer ihåg att jag gjorde ett jätteprojekt om första läs och skrivinläringen med hjälp av iPads. Min kompetensutveckling är av eget intresse, alltså att jag har fått reda på saker, och läst och tagit hem appar och provat själv och även köpt till mina egna barn (Lärare A).

6 Analys och diskussion

Kapitlet inleds med att studiens analys presenteras. Analysen utgår ifrån ramverket TPACK som hjälper till att förstå relationerna mellan de tekniska, pedagogiska och ämnesmässiga kunskaperna hos respondenterna. Utifrån de identifierade teman i resultatet stödjer ramverket förståelsen av studiens empiri. Kapitlet fortsätter med resultatet som studien visat i koppling med tidigare forskning. Avslutningsvis belyser kapitlet *slutsats, relevans* och *framtida forskning* för studien.

6.1 Analys utifrån TPACK

Utifrån Koehler och Mishra (2006) ramverk kan vi genom tekniska, pedagogiska -och ämnesmässiga kunskaper utveckla uppfattningen om hur programmering inom matematikämnet förstås och hur dessa tre kunskapsbaser kan tillämpas för en mer framgångsrik och effektiv undervisning. Det pedagogiska området (*Pedagogical Knowledge*) inbegriper didaktiska val och innefattar kunskap om metoder och strategier om bland annat lektionsupplägg (Willermark, 2018). Det kan sättas i relation till hur respondenterna arbetade med en varierande undervisning i klassrummet angående programmering genom teoretiska och praktiska upplägg. Genom en multimodal undervisning, vilket i studien innefattar en variation mellan traditionella och digitala uttrycksformer, fick eleverna gestalta sina kunskaper på en grundläggande nivå. Eleverna fick programmera varandra och tillverka mönster för att utveckla kunskaper till arbetet med digitala verktyg och även för att utveckla problemlösningsförmågan. Det här anser respondenterna kan väcka ett intresse hos eleverna genom en differentierad undervisning och interaktion tillsammans, men även som en ögonöppnare för den digitaliserande världen. I sammanhanget med kunskap om metoder och strategier påvisade respondenterna genomgående en tanke av att upplysa eleverna angående sina beslut och misstag i programmeringsundervisningen.

Vidare visade resultatet att kopplingen mellan matematik och programmering är en faktor för att möjliggöra en undervisning med genomtänkt kunskapsinnehåll i matematikämnet. Kopplingen mellan dessa områden kan förklaras och konkretiseras genom lärarens ämneskunskaper (*Content Knowledge*). Lärarens ämneskunskaper innefattar de kunskaper som läraren behöver utveckla för att tillgodose en undervisning inom de särskilda ämnena. Det kan röra sig om fakta, teorier eller aktuella begrepp (Willermark, 2018). Det här går att sätta i sammanhanget där respondenterna upplever att det krävs ett intresse och en kunskap inom matematik och programmering för att utveckla en förståelse för dess relation och samband. Resultatet påvisar olika matematiska områden som har en tydlig koppling till programmering, vilket möjliggör en kunskapsbefästande undervisning som inkluderar enstaka begrepp och teorier inom programmering och matematik. Det är tydligt att främst algebra, inkluderande mönster och stegvisa instruktioner, men även problemlösning utgör de huvudsakliga områdena där lärarna har enklast att se sambandet mellan programmering och matematik. Respondenterna anser att processen vid stegvisa instruktioner hjälper eleverna att utveckla sin problemlösningsförmåga då eleverna tillsammans får reflektera och åtgärda sina beslut.

Lärarens tekniska kunskaper (*Technological Knowledge*) är den tredje faktorn inom ramverket. Den beskrivs som förståelsen att uppfatta och bekräfta lärarens kunskaper inom den tekniska utvecklingen samt ge insikt för digitala verktyg och dess användningsområden (Willermark, 2018). Det här kan sättas i relation till hur respondenterna arbetar med olika applikationer på surfplattor eller arbetar med robotar i programmeringen inom matematikundervisningen. Resultatet visar att tekniken och de digitala verktygen är väsentliga för att erbjuda flera uttrycksformer som får eleverna att uppleva och tänka matematiskt och datalogiskt. Respondenterna upplever att lärare kan ha svårt att följa den tekniska utvecklingen och användningsområdena för digitala verktyg om det inte finns ett eget engagemang inom området.

Sammanfattningsvis framhåller TPACK tre kunskaper som tillsammans samarbetar för att uppnå en effektiv och meningsfull undervisning (Willermark, 2018). Det synliggörs i resultatet genom att respondenterna uppfattar undervisningen, dess olika kunskaper och en differentierad undervisning som central. Tekniken sätts i förhållande till lärarnas pedagogiska kunskaper och utvecklandet av relationen mellan programmering och matematik. Syftet med ramverket i den här studien är inte att bedöma respondenternas nivå av uppfyllelse av TPACK som kunskapsområde, utan för att synliggöra hur lärare reflekterar över kunskaperna inom programmering i matematikundervisningen, då de kan bidra med möjligheter eller hinder beroende på om lärarna har dem eller inte. För att uppnå en framgångsrik undervisning är kunskaper från de pedagogiska (*Pedagogical Knowledge*), ämnesmässiga (*Content Knowledge*) och tekniska områdena (*Technological Knowledge*) relevanta.

6.2 Resultatdiskussion

I det här kapitlet diskuteras studiens resultat utifrån de identifierade kunskaperna av TPACK och i koppling till tidigare forskning. De presenteras under följande rubriker; en multimodal programmeringsundervisning, didaktiska val och kunskaper i programmeringsundervisningen och programmering som en algebraisk problemlösning.

6.2.1 En multimodal programmeringsundervisning

Studiens resultat visar att samtliga lärare arbetade och reflekterade över integrerandet av programmering inom matematikundervisningen. Vidare framgick det i resultatet att flera lärare arbetade med fokus på en varierad undervisning där eleverna genom interaktion får gestalta sina kunskaper både teoretiskt och praktiskt. Welch, m.fl. (2022) förklarar att sociala sammanhang inom programmering påverkar barns kunskapsutveckling på ett positivt sätt. Forskarna påpekar också att en multimodal undervisning är utvecklande för att konstruera och kommunicera kunskap emellan lärare och elever (Welch, m.fl., 2022). Lärarna i undersökningen beskrev olika aktiviteter där eleverna skulle samarbeta genom grundläggande problemlösningssuppgifter, som innehåller stegvisa instruktioner. Lovric (2018) understryker betydelsen av att eleverna tillsammans behöver reflektera över problemlösning. Forskaren anser att eleverna behöver tid på sig för att tillsammans förstå hur algoritmer i programmering fungerar. Lärarna i studien utformade arbetsuppgifter inom programmering som låg på en grundläggande nivå för eleverna. Yildiz Durak (2018) framhäver att ett arbetssätt som ligger på en rimlig nivå i

förhållande till elevernas kunskaper kommer påverka elevernas engagemang och kunskapsutveckling positivt. Hon anser att programmeringsaktiviteter som börjar i en tidig ålder har en positiv påverkan på barns kognitiva utveckling.

Det framkom i resultatet att respondenternas undervisning till stor del innefattade mönster och stegvisa instruktioner som en del i att utveckla elevernas matematiska förmågor. Cui och Ng (2021) påpekar att aktiviteter innehavande mönster ger eleverna möjlighet att utforska upprepning i matematik, parallellt som det förbereder eleverna i förståelsen av loopar inom programmering. Respondenterna beskrev att de bland annat arbetade praktiskt med mönster i form av att skapa och rita med händerna. Messer m.fl. (2018) förklarar att elever som arbetar praktiskt med papper och penna har samma möjlighet att utveckla sina matematiska förmågor som elever som fått arbeta med digitala verktyg.

Samtliga respondenter i studien berättade även att de arbetade med olika digitala verktyg inom programmeringen i matematikundervisningen. De använde sig av BlueBots och BeeBots, samt olika applikationer till surfplattor. Erümit (2020) påvisar att undervisning där elever får arbeta med olika digitala aktiviteter kopplat till spel och programmering har en positiv effekt på deras utvecklande av olika matematiska förmågor. Samtidigt visar Welch m.fl. (2022) resultat att skärmfria kodningsleksaker ger utlopp för kommunikation och integration mellan elever på ett mer utvecklat sätt. Elever lär sig om programmering genom ett socialt sammanhang och utvecklar därav sina matematiska förmågor tillsammans med varandra (Welch, m.fl., 2022).

6.2.2 Programmering som en algebraisk problemlösning

Resultatet visar att samtliga lärare såg området algebra, vilket innefattar mönster och stegvisa instruktioner i kopplingen mellan matematik och programmering. De ansåg främst att mönster har en tydlig koppling till programmering, eftersom det bland annat används vid kodning av digitala verktyg. Miller (2019) påvisar att relationen mellan programmering och mönster har ett samband. Forskaren anser att elever som arbetar med kodning inom programmering har lättare att förstå strukturella former och mönster. Lärarna i studien ser algebra och mönster som ett sätt att beskriva förändring och något som kan repeteras eller startas om, vilket respondenterna redogör för som en loop. Rich m.fl. (2020) förklarar att mönster inom programmering är det matematiska samband som är lättast att synliggöra i koppling med matematikundervisningen, då det påvisas i form av loopar, som är bestämda mönster i en återupprepande sekvens.

Lärarna i studien beskrev genomgående stegvisa instruktioner som en del av olika aktiviteter för att förstå programmeringens gång. Respondenterna uttryckte främst förståelsen av stegvisa instruktioner i det praktiska arbetet, utan att koppla stegvisa instruktioner till en algoritm, som till exempel berättar för en BlueBot hur den ska manövrera eller vid användningen av Scratch och dess olika koder. Erümit (2020) anser att aktiviteter där elever får arbeta med Scratch utvecklar deras algoritmiska tänkande i form av att eleverna får bryta ned problem i mindre delar. Det är grundläggande för programmering och matematik, då elever får utveckla sina förmågor med stöd av stegvisa instruktioner och hur de konstrueras (Skolverket, 2022).

Utifrån kunskapsområdet algebra uttrycker lärarna en koppling mellan programmering och det andra kunskapsområdet problemlösning. Lärarna lyfter buggar vid introduceringen av felsökning, både inom arbetet med digitala verktyg och utan. Respondenterna beskriver buggar som ett fel och ett sätt att lösa felet är att dela upp instruktionerna i mindre delar, för att undersöka var felet uppstod. Erümit (2020) lyfter den digitala aspekten och vikten av att bryta ned problemen som en del av en problemlösningsprocess. Även Wing (2008) anser att datalogiskt tänkande kan utveckla problemlösningsförmågan hos eleverna, då det används för att redogöra, utvärdera och lösa olika problem.

6.2.3 Didaktiska val och kunskaper i programmeringsundervisningen

Resultatet visar att det finns olika faktorer som kan skapa en oro hos lärare i programmeringsundervisningen inom matematik. Faktorerna som oron grundas i är enligt respondenterna en bristande förståelse av programmering och matematik, samt en svag självkänsla i användandet av digitala verktyg. Sentance och Csizmadias (2018) påvisar en oro hos lärare gällande programmering inom matematikundervisningen. Trots att lärarna ägnat en större tid på egen kompetensutbildning, uppgav de sin kompetens inom området som bristfällig (Sentance & Csizmadias, 2018). Respondenterna i vår studie förklarar att lärare kan ha problem med att förstå kopplingen och behöver se sambandet mellan programmering och matematik, då det inte upplevs som en självklarhet. Kilhamn m.fl. (2020) belyser att flera lärare inte ser kopplingen mellan programmering och matematik. Lärare misslyckas i undervisningen av olika matematiska områden, då de inte har förståelsen av programmering och dess samband med matematik (Kilhamn, m.fl., 2020). Forskarna anser att programmeringen hamnar utanför matematikens ramar, vilket även Misfeldt m.fl. (2020) påvisar. Utöver sambanden mellan programmering och matematik, har skolans styrdokument en stor roll i hur lärare implementerar programmering inom matematikundervisningen. Cederqvist (2019) påpekar att styrdokumentet har ett stort tolkningsutrymme. Forskaren anser att förändringar och delar som tillkommer i styrdokumentet utvecklar en osäkerhet kring hur programmering ska undervisas och presenteras.

Lärarna i studien menar att den generation som går på lågstadiet idag kommer möta en annan framtid, innefattande yrken och förväntningar. Lärarna uttryckte vikten av att förse elever med grundläggande kunskaper för att möta behovet av ett allt mer digitaliserat samhälle. Det här anspelar på hur Manilla (2017) ser på tekniska kunskaper i dagens samhälle. Arbetsmarknadens nya behov ställer krav på den digitala kompetensen, vilket har kommit att avspegla skolsystemets kunskapsinnehåll (Manilla, 2017). Även Statens offentliga utredning (SOU 2016:85) påvisade att digital kompetens är av stor vikt för att människor ska fungera och verka i dagens samhälle. Den digitala kompetensen kan enligt Skolverket (2017) delas in i fyra olika aspekter. Flera av dessa aspekter exemplifierar lärarna i studien i sin undervisning med programmering inom matematikämnet, nämligen att eleverna ska utveckla en problemlösningsförmåga, utforma olika lösningar samt omsätta idéer till handlingar (Skolverket, 2017).

Resultatet av studien visar att lärarnas didaktiska val har en inverkan på undervisningen och elevers intresse i arbetet med programmering och digitala verktyg. Resultatet påvisar också att lärare upplever en differentierad undervisning som viktig för att väcka ett intresse, öka motivationen och befästa kunskaper inom

matematik och programmering. Rolandsson (2015) anser att lärares didaktiska kunskaper är en viktig faktor för hur elever tar in kunskapsinnehållet inom programmering. Ahmed m.fl. (2020) påpekar att programmering kan fungera som stöd i inläring av matematik förutsatt att det finns ett tydligt syfte och genomtänkta didaktiska val. Insikter som inte beaktar elevers kunskaper och lärande kan resultera i hinder och därav begränsa elevers i deras matematikutveckling. Även Tritrakan m.fl. (2016) anser att en kreativ och planerad undervisning inom programmering är viktigt för elevers lärande, vilket även Pelizzari, m.fl. (2023) lyfter.

Delar av resultatet visar även att det krävs ett engagemang och god inställning för att lyckas fullt ut med programmering i matematikundervisningen för lärare. Det framkom att lärarnas eget intresse för digitala verktyg är väsentligt för att utveckla elevernas intresse och kunskaper. Calao m.fl. (2015) anser att matematikundervisningen gynnas av olika övningar inom programmering, då de utvecklar elevers matematiska förmågor innefattande algoritmer, problemlösning, resonemang och modellering. Mozelius m.fl. (2019) påpekar att flertalet lärare har en positiv inställning till programmering. Däremot upplever lärare att de behöver fortbildning för att vidareutveckla förståelsen kring programmeringsundervisningen (Mozelius, m.fl., 2019).

Sammanfattningsvis visar studien att respondenterna förfogar över kunskaperna inom ramverket TPACK. Genom analys och resultatdiskussion synliggörs den pedagogiska (*Pedagogical Knowledge*), ämnes- (*Content Knowledge*) och tekniska kunskapen (*Technological Knowledge*) som centrala för ett arbete med programmering inom matematikundervisningen. Respondenterna arbetar med en varierad undervisning där interaktionen tillsammans är grunden för ett arbete med både traditionella och digitala uttrycksformer. Studien visar också att respondenterna ser flera kopplingar mellan matematik och programmering. Det framgick att algebra var den koppling alla respondenter såg, varpå samtliga uppgav att stegvisa instruktioner och mönster var två centrala delar i deras undervisning, vilka är delområden för algebra. De såg även problemlösning som en del av arbetet med programmering, där eleverna genom samarbete fick lösa olika problem. Avslutningsvis framgick det av respondenterna att en varierad undervisning som väcker elevernas intresse och bidrar till motivation, tillsammans med en grundläggande undervisning där eleverna får med sig basen med programmering och dess användningsområden, är viktigt för att eleverna ska befästa kunskapen inom matematik och programmering. Eleverna förses med kunskaper som kan ligga till grund för en digital kompetens för att de bättre ska möta framtiden.

6.3 Slutsats

Syftet med studien var att synliggöra förhållandet mellan programmering och matematik på lågstadiet, samt att undersöka vilka möjligheter erfarna lärare upplever med programmering inom matematikämnet. Utifrån analys och diskussion av studiens resultat kan vi dra slutsatsen att lärarna ser flera kopplingar mellan programmering och matematik, där algebra är den mest utmärkande. Kopplingen synliggörs främst via mönster och stegvisa instruktioner i undervisningen. Vi kan vidare dra slutsatsen att programmering ger lärarna möjlighet att erbjuda eleverna en multimodal undervisning där de tillsammans får arbeta praktiskt, teoretiskt och med hjälp av digitala verktyg befästa både programmerings- och matematikkunskaper. En multimodal undervisning kopplad till programmering bidrar till att väcka elevernas

intresse och förse dem med grundläggande kunskaper, som kan gynna deras digitala framtid.

6.4 Relevans och framtida forskning

Med tanke på att forskningen kring programmering och dess relation till matematiken i grundskolans tidiga år är något bristfällig, vill vi med hjälp av den här studien tillföra en förståelse av sambandet mellan programmering och matematik i undervisningen. Studien belyser även didaktiska val och faktorer som kan bidra till en förbättrad undervisning för lärare.

Denna studie visar hur erfarna lärare arbetar med programmering i matematikundervisningen, samt vilka kopplingar de kan dra mellan områdena. Det hade varit intressant att genomföra observationer hos lärarna för att synliggöra deras arbetssätt och aktiva val ytterligare. En annan dimension i forskningen hade varit att få ett elevperspektiv, där eleverna hade kunnat delge sina erfarenheter och tankar angående programmering i matematikundervisningen på lågstadiet.

Referenslista

- Ahmed, G., Nouri, J., Zhang, L. & Norén, E. (2020). Didactic Methods of Integrating Programming in Mathematics in Primary School - Findings From a Swedish National Project. DOI:[10.1145/3328778.3366839](https://doi.org/10.1145/3328778.3366839)
- Bryman, A. (2018). *Samhällsvetenskapliga metoder*. (3. uppl.). Liber.
- Calao, L. A., Moreno-León, J., Esther Correa, H., & Robles, G. (2015). Developing Mathematical Thinking with Scratch An Experiment with 6th Grade Students. *Using Educational Analytics to Improve Test Performance*, 17-27. https://doi.org/10.1007/978-3-319-24258-3_2
- Cederqvist, A. (2019). Pupils' ways of understanding programmed technological solutions when analysing structure and function. *Education and Information Technologies*, 2019, <https://doi.org/10.1007/s10639-019-10006-4>
- Corry, L. (2023). Algebra. *Encyclopedia Britannica*. Hämtad 2023, 9 maj från <https://www.britannica.com/science/algebra>
- Cui, Z., & NG, O.L. (2021). The interplay between mathematical and computational thinking in primary school students' mathematical problem-solving within a programming environment. *Journal of Educational Computing Research*, 59(5), 988-z 1012. <https://doi.org/10.1177/0735633120979930>
- David, M., & Sutton, C. (2016). *Samhällsvetenskaplig metod*. (1. uppl.). Studentlitteratur.
- Denscombe, M. (2018). *Forskningshandboken: för småskaliga forskningsprojekt inom samhällsvetenskaperna*. (2. uppl.). Studentlitteratur.
- Edvardsson, J., Godhe, A. & Magnusson, P. (2018). *Digitalisering, literacy och Multimodalitet*. Studentlitteratur.
- Erümit, A. K. (2020). Effects of different teaching approaches on programming skills. *Education and Information Technologies*, 25(2), 1013-1037. <https://doi.org/10.1007/s10639-019-10010-8>
- Eugene, G. (2016). Robots, Programming and Coding, Oh My!. *Childhood Education*, 92:4, 298-304, DOI: [10.1080/00094056.2016.1208008](https://doi.org/10.1080/00094056.2016.1208008)
- Europeiska kommissionen, Generaldirektoratet för informationsområdet och medier, Gualtieri, V., Curtarelli, M., Donlevy, V. (2017). *ICT for work : digital skills in the workplace : final report*, Publications Office. <https://data.europa.eu/doi/10.2759/498467>
- Helenius, O., Misfeldt, M., Rolandsson, L., & Ryan, U. (2018). *Om programmering i matematikundervisning*. Hämtad från <https://larportalen.skolverket.se>

- Kafai, Y. (2016). From computational thinking to computational participation in K-12 education. *Communications of the ACM*, 59(8), 26-27. DOI: 10.1145/2955114.
- Kilhamn, C., Rolandsson, L., & Bråting, K. (2021). Programmering i svensk skolmatematik: Programming in Swedish school mathematics. *LUMAT: International Journal on Math, Science and Technology Education*, 9(1), 283–312. <https://doi.org/10.31129/LUMAT.9.2.1457>
- Kim, S-W & Lee, Y. (2018). The Effects of the TPACK-P Educational Program on Teachers' TPACK: Programming as a Technological Tool. *International Journal of Engineering & Technology*, 7 (3.34) (2018) 636-643. doi:[10.14419/ijet.v7i3.34.19405](https://doi.org/10.14419/ijet.v7i3.34.19405)
- Kjällander, S., Åkerfeldt, A., & Petersen, P. (2016). *Översikt avseende forskning och erfarenheter kring programmering i förskola och grundskola*. Skolverket. Hämtad från <http://urn.kb.se/resolve?urn=urn:nbn:se:su:diva-181018>
- Louden, K.C & Lambert, K.A. (2012). *Programming languages: Principles and Practice*. 3.uppl., USA: Course Technology.
- Lovric, M. (2018). Programming and Mathematics in an Upper-Level University Problem-Solving Course. *PRIMUS : Problems, Resources, and Issues in Mathematics Undergraduate Studies*, 28(7), 683–698. <https://doi.org/10.1080/10511970.2017.1403524>
- Lärarnas Riksförbund. (2016). *Digital framtid utan fallgropar*. Hämtad från https://www.lr.se/download/18.766408f2157e419429fc1598/1477390255020/digital_framtid_utan_fallgropar_201610.pdf
- Lärarnas Riksförbund. (2020). *Programmering – en skolreform utan program*. Hämtad från <https://www.lr.se/opinion--debatt/undersokningar/2020/2020-04-30-programmering---en-skolreform-utan-program>
- Mannila, L. (2017). *Att undervisa i programmering i skolan: varför, vad och hur?. (Upplaga 1)*. Studentlitteratur.
- Messer, D., Lucy, L., Holiman, A. & Natalia, N. (2018). Evaluating the Effectiveness of an Educational Programming Intervention on Children's Mathematics Skills, *Spatial Awareness and Working Memory*. <http://dx.doi.org/10.1007/s10639-018-9747-x>
- Misfeldt, M., & Ejsing-Duun, S. (2015). Learning Mathematics through Programming: An Instrumental Approach to Potentials and Pitfalls. In K. Krainer, & N. Vondrová (Eds.), *CERME9: Proceedings of the Ninth Congress of the European Society for Research in Mathematics Education (524-2530)*. Prague, Czech Republic: Charles University in Prague, Faculty of Education and ERME.

- Misfeldt, M., Jankvist, U. T., Geraniou, E., & Bråting, K. (2020). Relations between mathematics and programming in school : juxtaposing three different cases. *Proceedings of the 10th ERME Topic Conference on Mathematics Education in the Digital Era, MEDA 2020*, 225-262. Hämtad från <http://urn.kb.se/resolve?urn=urn:nbn:se:uu:diva-420996>
- Miller, J. (2019). STEM education in the primary years to support mathematical thinking: using coding to identify mathematical structures and patterns. *ZDM Mathematics Education* 51, 915–927. <https://doi.org/10.1007/s11858-019-01096-y>
- Mishra, P., & Koehler, M. J. (2006). Technological pedagogical content knowledge: A framework for teacher knowledge. *Teachers College Record*, 108(6), 1017-1054. <https://doi.org/10.1111/j.1467-9620.2006.00684.x>
- Mozelius, P., Ulfenborg, M., & Persson, N. (2019). Teacher Attitudes Towards The Integration Of Programming In Middle School Mathematics. *INTED 2019 Proceedings*, 701–706. <https://doi.org/10.21125/inted.2019.0249>
- Nationalencyklopedin. (u.å.). Loop. *Nationalencyklopedin*. Hämtad 2023, 9 maj från <https://www.ne.se/uppslagsverk/encyklopedi/l%C3%A5ng/loop>
- Nationalencyklopedin. (u.å.). Programmering. *Nationalencyklopedin*. Hämtad 2023, 9 maj från <https://www.ne.se/uppslagsverk/encyklopedi/l%C3%A5ng/programmering>
- Pelizzari, F., Marangi, M., Rivoltella, P. C., Peretti, G., Massaro, D., & Villani, D. (2023). Coding and childhood between play and learning: Research on the impact of coding in the learning of 4-year-olds. *Research on Education and Media*, 15(1), 9-19. doi: <https://doi.org/10.2478/rem-2023-0003>
- Rich, K. M., Spaepen, E., Strickland, C., & Moran, C. (2020). Synergies and differences in mathematical and computational thinking: Implications for integrated instruction. *Interactive Learning Environments*, 28(3), 272-283. <https://doi.org/10.1080/10494820.2019.1612445>
- Rolandsson, L. (2015). *Programmed or Not*. (TRITA-ECE, 2015:3) [Doktorsavhandling] Kungliga Tekniska Högskolan SE-100. <https://www.diva-portal.org/smash/get/diva2:791197/FULLTEXT02.pdf>
- Romero, M., Duflot, M., & Viéville, T. (2019). Le jeu du robot : analyse d'une activité d'informatique débranchée sous la perspective de la cognition incarnée.. *Review of science, mathematics and ICT education*, 13 (1), ff10.26220/rev.3089ff. fffal-02144467
- Scratch. (2023, 9 maj). *Om Scratch*. <https://scratch.mit.edu/about>
- Sentance, S., & Csizmadia, A. (2017). Computing in the curriculum: Challenges and strategies from a teacher's perspective. *Education and Information*

- Technologies*, 22(2), 469–495. <https://doi.org/10.1007/s10639-016-9482-0>
- Shulman, L. (1986). Those Who Understand: Knowledge Growth in Teaching. *Educational Researcher*, 15(2), 4–14.
- Skolverket. (2020). Bedömning på lektionstid underlättar vid distansundervisning. <https://www.skolverket.se/skolutveckling/inspiration-och-stod-i-arbetet/inspiration-och-reportage/bedomning-pa-lektionstid-underlattarvid-distansundervisning>
- Skolverket (2018). *Del 3: Programmering med matematik (Matematikundervisning med digitala verktyg II)*. https://larportalen.skolverket.se/LarportalenAPI/api-v2/document/path/larportalen/material/inriktningar/1-matematik/Grundskola/428_matematikundervisningmeddigitalaverktygI_%C3%A5k4-6/del_03/Material/Flik/Del_03_MomentA/Artiklar/MA2_4-6_03A_01_programmeringmed.docx
- Skolverket. (2017). *Få syn på digitaliseringen på grundskolenivå. Ett kommentarmaterial till läroplanerna för förskoleklass, fritidshem och grundskoleutbildning*. Skolverket.
- Skolverket. (2022). *Läroplan för grundskolan, förskoleklassen och fritidshemmet*. Skolverket.
- Specialpedagogiska skolmyndigheten. (19 Januari 2023). *Utveckla digitala kompetens*. <https://www.spsm.se/stod/specialpedagogiskt-stod/digitalt-larande/utveckla-digital-kompetens/>
- Sjöberg, L. (2019). *Lärares uppfattningar om införandet av programmering i gymnasieskolans matematikämne*. (Examensarbete). Kungliga Tekniska Högskolan, Skolan för Industriell Teknik och Management. Tillgänglig: <http://www.diva-portal.org/smash/get/diva2:1316923/FULLTEXT01.pdf>
- SOU 2016:85. *Digitaliseringens effekter på individ och samhälle: fyra temarapporter*. https://www.regeringen.se/contentassets/bf87c5fce6fc4f9a889d57ea2e46a27d/sou-2016_85_webb-pdf-med-framsida.pdf
- Tivenius, O. (2015). *Uppsatsens inre liv*. Studentlitteratur AB.
- Tritrakan, K., Kidrakarn, P., & Asanok, M. (2016). The use of engineering design concept for computer programming course: A model of blended learning environment. *Educational Research and Reviews*, 11(18), 1757-1765.
- Vetenskapsrådet. (2017). God forskningsred. Vetenskapsrådet. Hämtad 2023 9 maj från https://www.vr.se/download/18.2412c5311624176023d25b05/1555332112063/Godforskningssed_VR_2017.pdf

- Welch, L., Shumway, J., Clarke-Midura, J. & Lee, V. (2022). Exploring Measurement through Coding: Children's Conceptions of a Dynamic Linear Unit with Robot Coding Toys. *Educ.Sci.* 2022, 12, 143. <https://doi.org/10.3390/educsci12020143>
- Willermark, S. (2018). Digital Didaktisk Design- *Att utveckla undervisningspraktiken i och för en digitaliserad skola.* [Doktorsavhandling] University West SE 46-186. <http://www.diva-portal.org/smash/get/diva2:1174749/FULLTEXT01.pdf>
- Wing, J.M. (2008). Computational thinking and thinking about computing. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, <https://doi.org/10.1098/rsta.2008.0118>
- Yildiz Durak, H. (2018). The Effects of Using Different Tools in Programming Teaching of Secondary School Students on Engagement, Computational Thinking and Reflective Thinking Skills for Problem Solving. *Tech Know Learn* 25, 179–195. <https://doi-org.proxy.mau.se/10.1007/s10758-018-9391-y>

Bilaga 1 – Missivbrev till lärare

Mälardalens Universitet

Västerås

2023-03-30



Hej!

Vi är två lärarstudenter som studerar sista terminen på Mälardalens Universitet. Vi skriver just nu vårt examensarbete.

Ämnet vi skriver om är *Programmering i matematikundervisningen*. Syftet med arbetet är att undersöka om hur behöriga lärare arbetar med programmering i matematikundervisningen, men även hur lärare upplever fördelar med programmering i undervisningen.

Vi kontaktar dig i förhoppning att du vill medverka i en intervju. Intervjun beräknas att ta cirka 40 minuter och kommer att genomföras via ett digitalt videokonferensprogram. Vi tror intervjun kommer leda till en givande diskussion där vi kan få mer insikter om hur ämnet ter sig i praktiken, vilket skulle vara hjälpsamt för att utveckla vårt arbete.

Intervjuerna kommer att vara enskilt och spelas in, detta på grund av att säkerställa att vi har fått med all information som vi vill ta med. Vi följer även vetenskapsrådets forskningsetiska principerna, där informanterna, det vill säga lärarnas identitet kommer att vara anonyma och de lovas även konfidentialitet. Det betyder att varken namn på verksamheten eller deltagaren kommer tas med utan tillstånd. All information som vi har samlat in under intervjun kommer att användas i vår studie. Det inspelade materialet kommer att raderas bort efter avslutat arbete. 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.

Undersökningen kommer att presenteras i form av en uppsats vid Mälardalens universitet som i sin slutversion läggs ut på databasen DiVA.

Har du några funderingar kring arbetet kan du kontakta oss för mer information.

Med vänliga hälsningar,

Pontus Vilén

☎ 070 898 25 95

✉ pvn21002@student.mdu.se

Emil Toikkanen

☎ 070 047 47 51

✉ etn19003@student.mdu.se

Handledare:

Roger Andersson

✉ roger.andersson@mdu.se

Ytterligare upplysningar lämnas av
pvn21002@student.mdu.se och
etn19003@student.mdu.se.

Västerås 2023-03-30

Bilaga 2 - Intervjuguide

Bakgrundsfrågor

1. Hur länge har du arbetat som lärare efter din examen?
2. Vilken klass undervisar du i?
3. Vilka ämnen undervisar du i?
4. Hur länge har du arbetat med programmering i matematikundervisningen?

Programmering

- 1 Vad innebär programmering för dig?
- 2 Skulle du kunna beskriva hur du använder programmering i din matematikundervisning?
- 3 Vilka matematiska områden används programmeringen till? Känner du att du kan dessa områden? Hur? På vilket sätt?
- 4 Anser du att den finns en tydlig koppling mellan matematik och programmering? Och i så fall vilken/vilka?
- 5 Om du använder programmeringsverktyg i din undervisning? Vad/vilka använder du? Ex: Appar, robotar, spel etc. Varför/Varför inte?
- 6 Anser du dig ha rätt material och förutsättningar till att använda dig av programmering inom matematiken?
- 7 Ser du fördelar med att använda programmering i matematikundervisningen?
- 8 Ser du några utmaningar med att använda programmering i matematikundervisningen? – Varför tror du så många lärare ser utmaningar med programmering? Kopplingen, tekniken, ämnesförståelsen
- 9 Hur ser dina förkunskaper ut gällande programmering? Vilken utbildning har du gällande programmering?
- 10 Har du mottagit någon kompetensutbildning inom ämnet? Tex fortbildning.
- 11 Har du lagt egen tid för att utveckla din kunskap om programmering innanför/utanför din förtroende tid? Tror du att andra lärare gör det?
- 12 Diskuterar ni i arbetslaget om hur ni kan arbeta för att inkludera programmering inom matematiken? Om ja, på vilket sätt?
- 13 Vad tror du är det viktigaste för eleverna att få med sig när de lär sig programmering kopplat till matematik?