

Fördelarna med att inkludera  
programmering i  
den grundläggande utbildningen  
(HALVFÄRDIG)

Nicola Sandblom 2001147

Kandidatavhandling i datateknik

# Referat

...

# Innehållsförteckning

Referat .....	1
1 Inledning .....	3
2 Bakgrund.....	4
2.1 Utveckling.....	4
2.2 Programmering i finländska skolor .....	4
3 Kognitiva effekter .....	7
3.1 Datalogiskt tänkande .....	7
3.1.1 Vad är datalogiskt tänkande? .....	7
3.1.2 Varför är datalogiskt tänkande gynnsamt?.....	8
3.2 Andra kognitiva förmågor .....	9
3.2.1 Matematiska förmågor .....	9
3.2.2 Kreativt tänkande .....	10
3.2.3 Övrigt.....	10
4 Sociala aspekter.....	11
4.1 Kommunikation och gemenskap .....	11
4.2 Intresse för branschen .....	11
5 Diskussion.....	13
Referenser.....	14

# 1 Inledning

Informationsteknologi blir en allt större del av vår vardag. Teknologi används så gott som överallt, och i många yrken har det blivit en förutsättning att ha grundläggande kunskap om informationsteknologi. Eftersom den grundläggande utbildningen har som mål att allmänbilda eleverna och ge dem goda förutsättningar för framtiden, är det viktigt att informationsteknologin kommer fram på något sätt i läroplanen. Många länder runt om i världen har under de senaste årtiondena börjat inkludera programmering i den grundläggande utbildningen. Att inkludera programmering i utbildningen handlar dock inte endast om att förbereda elever för en potentiell karriär inom informationsteknologi, på samma sätt som heller inte musikundervisningen handlar om att utbilda blivande musiker. Programmering finns med i undervisningen för att allmänbilda eleverna och få dem att bli medvetna om vad som händer i bakgrunden i tekniken som finns i vårt samhälle. [1]

Syftet med denna avhandling är att lyfta fram vilka fördelar det finns med att ha programmering som en del av den grundläggande utbildningen.

To be continued...

## 2 Bakgrund

### 2.1 Utveckling

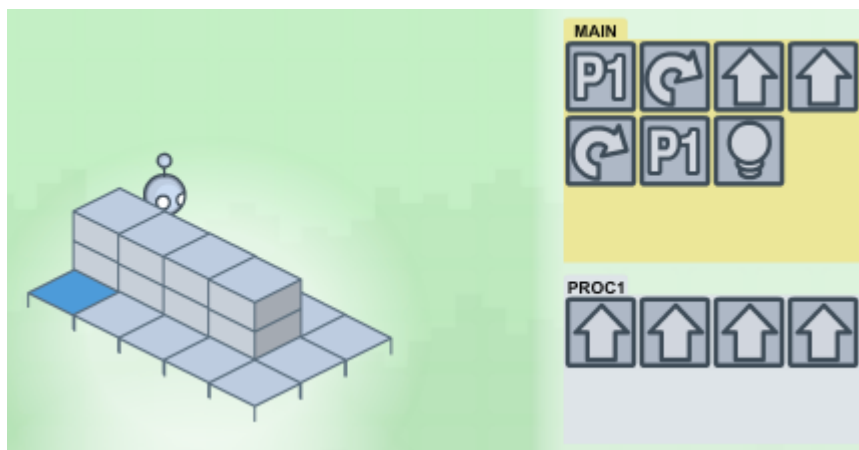
Informationsteknologi har redan länge varit aktuellt, men tidigare har programmering undervisats endast i utbildning på högre nivå. Nu har allt flera länder runt om i världen har börjat inkludera programmering som en del av den grundläggande utbildningen. Egentligen är detta inte ett helt nytt fenomen. Redan på 60-talet utvecklades programmeringsspråket Logo som var menat att användas av barn som lär sig programmering. [2] Under 80-talet var det många skolor som använde sig av programmering under lektionerna, men redan ett årtionde senare hade programmering försvunnit från den grundläggande utbildningen. Detta berodde bland annat på att det inte fanns tillräckligt med intresse för att lära sig programmering, och för att det var svårt att koppla ihop programmering med resten av utbildningen. [3]

På 2000-talet har det på nytt blivit vanligt att inkludera programmering som en del av den grundläggande utbildningen. I Finland har det i grunderna för läroplanen för den grundläggande utbildningen år 2014 bestämts att programmering ska ingå som en del av matematikundervisningen [1]. Även länder som Sverige, USA och Storbritannien har inkluderat informationsteknologi och programmering i någon form i deras grundläggande utbildning. [4] Att det i Finland först år 2014 bestämts att programmering ska ingå som en del av den grundläggande utbildningen kan anses vara ganska sent, eftersom teknologin redan i många år innan detta har varit en stor del av vardagen. Förklaringen till detta är att grunderna för läroplanen förnyas med ungefär 10 års mellanrum, så då de tidigare bestämmelserna kom ut år 2004 var ämnet inte ännu lika relevant och blev därför inte en del av dem.

### 2.2 Programmering i finländska skolor

I grunderna för läroplanen för den grundläggande utbildningen beskrivs att utbildningen ska hjälpa eleverna att utveckla digital kompetens. Till digital kompetens räknas bland annat programmeringsfärdigheter. [1]

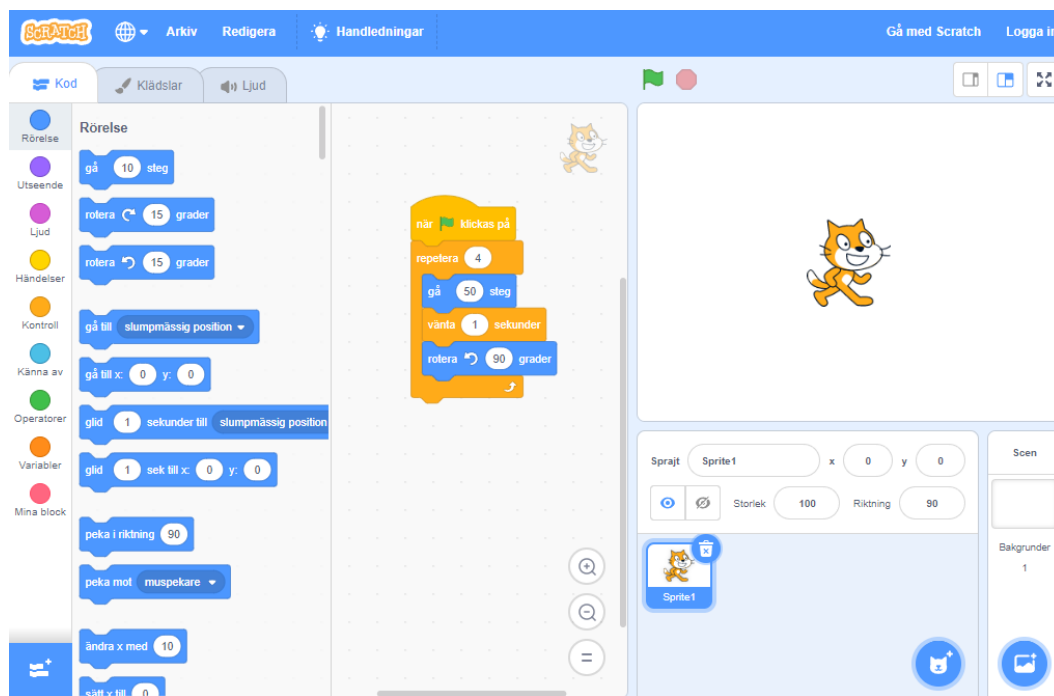
Eleverna i årskurs 1–2 ska lära sig programmering på den nivå att de kan skapa stegvisa instruktioner som även testas. Användningen av spel för att främja lärandet av dessa färdigheter uppmuntras starkt. [1] Det finns en mängd olika lekar och spel som kan användas för att lära eleverna grunderna i hur programmering fungerar. Ett exempel är spelet Lightbot, där spelaren styr en robot genom olika banor genom att skapa sekvenser av instruktioner. [5] I Figur 1 visas ett exempel på en Lightbot bana, där roboten går runt väggen till andra sidan och tänds en lampa. För att lära ut tankesättet som ligger till grund för programmering måste inte teknologi användas. Det finns en hel del fysiska lekar och brädspel som kan fungera som bra sätt att lära ut detta.



Figur 1 - Exempel på en Lightbot bana. Hämtad från [5].

Eleverna i årskurs 3–6 ska lära sig programmering för att lära sig bakgrunden till tekniska funktioner. Eleverna ska alltså lära sig att det är mänskliga lösningar som är grunden till hur tekniken fungerar. För att lära sig programmering ska visuella programmeringsmiljöer användas för att skapa datorprogram. I slutet av årskurs 6 anses goda programmeringsfärdigheter vara att eleven kan programmera ett fungerande program i en visuell programmeringsmiljö. [1] Ett exempel på en visuell programmeringsmiljö som kan användas är Scratch. [2] Scratch är en blockbaserad programmeringsmiljö. I Scratch finns olika typer av block, som till exempel variabler, if-satser och loopar, som användaren drar in i sitt program för att få önskat beteende av programmet. Genom blockbaserad programmering kan användaren lära sig grunderna till hur textbaserade programmeringsspråk fungerar, men de behöver inte ännu lära sig någon syntax. [2] I Figur 2 kan man se ett enkelt exempel av hur man kan

använda Scratch. Programmet i figuren gör att katten går i ett fyrkantsmönster då man klickar på start-flaggan. Det går att skapa program som är mycket mera invecklade än detta, vilket gör att Scratch kan användas av elever vars programmeringskunskaper är på olika nivå.



Figur 2- Skärmdump av Scratch programmet

Eleverna i årskurs 7–9 ska lära sig att tillämpa programmering för att lösa problem. [1] För programmering ska textbaserade programmeringsspråk som till exempel Python eller JavaScript användas, men även visuella programmeringsmiljöer kan fortfarande utnyttjas. [6] Förutom programmering ska eleverna även lära sig god programmeringspraxis, alltså vad som traditionellt anses vara bra kod. I slutet av årskurs 9 anses goda programmeringsfärdigheter vara att eleven kan producera enkla program med hjälp av ett textbaserat programmeringsspråk. [1]

## 3 Kognitiva effekter

Att lära sig programmering har visat sig vara fördelaktigt för utvecklingen av ett flertal kognitiva förmågor. [7] Först och främst utvecklas det datalogiska tänkandet (eng. computational thinking) när man lär sig att programmera. Detta beskrivs närmare i nästa stycke. Andra kognitiva förmågor som utvecklas genom att lära sig programmering är kreativt tänkande, matematiska förmågor samt slutledningsförmågor. Att lära sig programmering tycks även ha en viss positiv inverkan på utvecklingen några andra kognitiva förmågor än dessa, men inte till lika stor del. [7]

### 3.1 Datalogiskt tänkande

#### 3.1.1 Vad är datalogiskt tänkande?

Datalogiskt tänkande har definierats på många olika sätt, men en av de vanligaste definitionerna är att det är förmågan att skapa beräkningar som får datorer att utföra arbete för oss, samt att förklara och tolka världen som ett komplex av informationsprocesser [8]. Datalogiskt tänkande innebär dock inte att man måste programmera, utan det handlar om själva tankeprocessen. [9] Det finns egentligen många olika processer som förknippas med datalogiskt tänkande. Enligt [9] är dessa:

- Logiskt resonemang. Man kan förklara varför någonting är på ett visst sätt, och man kan förutspå hur någonting kommer att bete sig på basen av tidigare kunskap. Skolelever använder logiskt resonemang till exempel när de förklarar hur de har kommit fram till en slutsats på basen av resultatet från ett experiment de har gjort.
- Algoritmiskt tänkande. En algoritm är en sekvens av instruktioner som används för att uppnå ett visst resultat. Ett vardagligt exempel på algoritmer är recept. De har steg för steg instruktioner över hur man ska göra för att tillreda en viss rätt.



- Uppdelning av problem i mindre delar. Då man delar upp ett problem i mindre delar kan det vara lättare att lösa problemet. Det här är en metod som kan användas för att lösa större och mera komplicerade problem. Till exempel då man ska planera en presentation kommer den här metoden ofta till användning. Man delar upp innehållet i presentationen i olika delar som man sedan kan planera skilt för sig.
- Abstraktion. Denna process är den viktigaste då man talar om datalogiskt tänkande. Abstraktion innebär att man identifierar det som är viktigast i något för att kunna simplificera det. Ett exempel på en abstraktion är skolschemat. I schemat kan man se när en viss lektion ska hållas, men extra information som till exempel vad som behandlas en specifik lektion framgår inte.
- Identifiering av mönster och generalisering. Genom att identifiera mönster i problem går det att märka vilka regler som gäller i en viss situation. Nästa gång man stöter på något liknande kan man utnyttja reglerna för att lösa problemet. Den här metoden använder man sig av till exempel då man lär sig ett nytt språk och märker att liknande ord kan böjas på samma sätt.

Datalogiskt tänkande används alltså inte enbart av programmerare. En typisk människa använder sig ofta av datalogiskt tänkande utan att kanske själv inse det. Det kan till exempel handla om att lösa en textbaserad uppgift i matematiken. Innan man kan ta hjälp av kalkylatorn, måste man analysera och lägga upp uppgiften på ett sådant sätt att det faktiskt går att lägga in beräkningar i kalkylatorn och få ett svar. På basen av det här kan man dra slutsatsen av att man inte måste lära sig att programmera för att utveckla det datalogiska tänkandet, men tills vidare verkar ändå programmering vara det effektivaste sättet att utveckla det datalogiska tänkandet på. [9]

### 3.1.2 Varför är datalogiskt tänkande gynnsamt?

Eftersom datalogiskt tänkande är starkt förknippat med problemlösning, är datalogiskt tänkande en förmåga som är fördelaktig för alla individer. Genom datalogiskt tänkande kan man bättre förstå problem och hur de ska lösas,

samtidigt som man förstår vilka restriktioner det finns till att lösa problemen. I dagens läge behövs problemlösningsförmågor konstant. [10]

I en digital värld som denna är datalogiskt tänkande en förmåga som behövs i nästan alla yrken. Maskiner används som hjälpmedel i alla sektorer, och då behövs förmågan att kunna utnyttja dem. [11] På grund av de här orsakerna är det bra att elever redan i ett tidigt stadiet utvecklar förmågan till datalogiskt tänkande.

## 3.2 Andra kognitiva förmågor

Förutom datalogiskt tänkande, hjälper programmering till att utveckla även andra kognitiva förmågor. Programmering tycks ha en stor positiv inverkan på utvecklingen av kreativt tänkande och matematiska förmågor. Dessutom tycks programmering ha en viss positiv inverkan på utvecklingen av metakognitiva förmågor, slutledningsförmågor, spatiala förmågor samt resultat i skolan. [7]

### 3.2.1 Matematiska förmågor

Flera studier har visat att programmering hjälper till att utveckla matematiska förmågor. [7] Detta beror troligen på att programmering och matematiskt tänkande har många likheter. Både i programmering och matematiken behövs problemlösningsfärdigheter som även är förknippade med datalogiskt tänkande. [7]

I en studie av Sáez-López m.fl. [12] undersöktes programmeringens inverkan på lågstadielevs matematiska förmågor. Eleverna i studien använde sig av blockbaserad programmering samt robotik för att göra matematikövningar. Resultaten från studien visade statistiskt signifikanta förbättringar i elevernas matematiska förmågor. Speciellt positiva effekter hade övningarna med sekvenser och if-satser i samband med robotprogrammeringen. Resultaten stöder alltså det tidigare påståendet om att matematiska förmågor utvecklas genom programmering.

### 3.2.2 Kreativt tänkande

Kreativt tänkande främjas av programmering. [7] Detta kan förklaras av att programmering är en kreativ process. För att hitta lösningar på problem man kan stöta på när man programmerar krävs det kreativitet, därför utvecklas även det kreativa tänkandet genom programmering. Att lära sig programmering tycks vara lika effektivt för att utveckla sitt kreativa tänkande som alternativa tillvägagångssätt. [7] I skolan kan kreativitet främjas bäst av att eleverna själva får skapa något originellt, i stället för att precis följa instruktioner över hur de ska göra något eller fylla i det som saknas från en uppgift. [9]

### 3.2.3 Övrigt

...

## 4 Sociala aspekter

### 4.1 Kommunikation och gemenskap

Programmering kan ofta göras i par. Parprogrammering fungerar som ett sätt att vidare utveckla ens programmeringsförmågor, eftersom programmerarna kan ha lite olika idéer och diskutera dem tillsammans. Parprogrammering ger eleverna möjlighet till att bli bättre på att samarbeta med andra människor samtidigt som de får umgås med sina jämnåriga. [13]

...

### 4.2 Intresse för branschen

I och med att världen blir allt mera digital, ökar även antalet arbetsplatser inom den informationsteknologiska branschen. Med tanke på detta är det bra att inkludera programmering i den grundläggande utbildningen för att förbereda eleverna inför en potentiell karriär inom informationsteknologi, samt som ett försök att väcka intresse hos flera att söka sig till en sådan karriär.

Ännu i dagens läge är den stora majoriteten av personer som inom den informationsteknologiska branschen män. Branschen skulle kunna dra stor nytta av att ha större diversitet bland människor som jobbar i den. Teknologiska lösningar produceras för att användas av hela samhället. De som producerar dessa lösningar borde kunna ta i beaktande användarnas behov för att skapa de mest optimala lösningarna. Detta kan endast förverkligas genom att försöka se på problemet från så varierade perspektiv som möjligt, vilket är orsaken till att diversitet inom branschen skulle vara positivt. [14]

Konak [15] utförde en liten studie om elevers intresse för en karriär inom informationsteknologi. Elever i högstadietåldern fick delta i en programmeringsworkshop som var en dag lång. Både innan och efter workshopen fick eleverna fylla i enkäter, där det bland annat fanns frågor om deras intresse för en karriär inom informationsteknologi. Innan workshopen fanns det endast lite intresse för en sådan karriär bland flickorna, medan pojkarna uttryckte ett större

intresse. Efter workshopen hade inte pojkarnas intresse ändrat, men intresset bland flickorna hade stigit signifikant. Från detta resultat kan man dra slutsatsen att det finns en större sannolikhet att elever, specifikt flickor, utvecklar ett intresse för den informationsteknologiska branschen ifall de har lärt sig programmering.

## 5 Diskussion

## Referenser

- [1] Utbildningsstyrelsen, "Grunderna för läroplanen för den grundläggande utbildningen," 2014.
- [2] M. Jancheski, "Improving Teaching and Learning Computer Programming in Schools through Educational Software," *Olympiads in Informatics*, vol. 11, nr 1, pp. 55-75, 2017.
- [3] J. Moreno-León, M. Román-González och G. Robles, "On computational thinking as a universal skill: A review of the latest research on this ability," i *2018 IEEE Global Engineering Education Conference (EDUCON)*, 2018.
- [4] C. Szabo, J. Sheard, A. Luxton-Reilly, S. B. A. Becker och L. Ott, "Fifteen Years of Introductory Programming in Schools: A Global Overview of K-12 Initiatives," i *Koli Calling International Conference on Computing Education*, 2019.
- [5] L. Inc, "How does Lightbot teach programming?," [Online]. Available: <https://lightbot.com/hoclearn.html>. [Använd 28 mars 2022].
- [6] Utbildningsstyrelsen, "Lärandeprogression inom programmering," [Online]. Available: <https://www.opf.fi/sv/larandeprogression-inom-programmering>. [Använd 28 mars 2022].
- [7] R. Scherer, F. Siddiq och B. Sánchez Viveros, "The Cognitive Benefits of Learning Computer Programming: A Meta-Analysis of Transfer Effects," *Journal of Educational Psychology*, vol. 111, nr 5, pp. 764-792, 2019.
- [8] P. J. Denning och M. Tedre, *Computational Thinking*, Cambridge: MIT Press, 2019.
- [9] M. Berry, *Quick Start Computing: A CPD toolkit for Primary Teachers*, Swindon: BCS, 2015.

- [10] M. Mohaghegh och M. McCauley, "Computational Thinking: The Skill Set of the 21st Century," *International Journal of Computer Science and Information Technologies (IJCSIT)*, vol. 7, nr 3, pp. 1524-1530, 2016.
- [11] P. Snalune, "The Benefits of Computational Thinking," *ITNOW*, vol. 57, nr 4, pp. 58-59, 2015.
- [12] J.-M. Sáez-López, M.-L. Sevillano-García och E. Vazquez-Cano, "The effect of programming on primary school students' mathematical and scientific understanding: educational use of mBot," *Educational Technology Research and Development*, vol. 67, pp. 1405-1425, 2019.
- [13] B. Hanks, S. Fitzgerald, R. McCauley, L. Murphy och C. Zander, "Pair programming in education: a literature review," *Computer Science Education*, vol. 21, nr 2, pp. 135-173, 2011.
- [14] W. DuBow, "Attracting and Retaining Women in Computing," *Computer*, vol. 47, nr 10, pp. 90-93, 2014.
- [15] J. Konak, "Programming for girls," i *2016 IEEE Integrated STEM Education Conference (ISEC)*, 2016.