

Programmering i anpassad grundskola, åk 7–9

Den här modulen är inte längre aktuell och kommer därför att avpubliceras och arkiveras 1 juli 2025.

Målet med denna modul är att du ska fördjupa dina kunskaper i lärande och undervisning om/i programmering. Modulen fokuserar på programmering i matematikämnet.

Rubrikerna på delarna ger en kort sammanfattning av innehållet i respektive del.

1. Olika programmeringsmiljöer
2. Datorer och programmering i matematikundervisning – då och nu
3. Programmering som språk
4. Kommunicera med entydiga instruktioner

I modulen ges förslag på aktiviteter som ni kan genomföra med era elever.

Del 2. Datorer och programmering i undervisningen – då och nu

Den här modulen är inte längre aktuell och kommer därför att avpubliceras och arkiveras 1 juli 2025.

Målet med denna del är att ge en historisk tillbakablick på datorer och programmering i skolans matematikundervisning. Ni kommer även att bekanta er med begreppet tinkering.

Del 2: Moment A – individuell förberedelse

Läs

I texten "Datorer och programmering i undervisningen – då och nu" får ni en historisk tillbakablick på datorers och programmeringens framväxt i undervisningen. I samma text presenteras några forskningsresultat om programmering i matematikundervisning samt det didaktiska perspektivet tinkering. I "Aktiviteter – följa instruktioner" får ni förslag på aktiviteter, med fokus på tinkering, som ni kan använda i era klasser/grupper.

Material

Material



Datorer och programmering i undervisningen – då och nu
C. Olteanu



Aktiviteter – följa instruktioner
C. Olteanu, H. Larsson & I. Ekeback

Datorer och programmering i undervisningen – då och nu

Constanta Olteanu, Linnéuniversitetet

Programmering definieras som processen att utveckla och genomföra olika uppsättningar av instruktioner för att en dator ska kunna utföra en viss uppgift eller lösa problem. Programmering handlar om att skriva kod, men omfattar även andra aspekter däribland innehållsskapande, kommunikativa och demokratiska dimensioner (Bers, 2020).

Datorer i skolan

Införandet av datorer i skolan har pågått i drygt femtio år och har genomgått fyra faser. Den första fasen började under senare delen av 1960-talet när de första försöken gjordes för att införa datorn i skolan och i undervisningen. I Lgr 69 skrevs ”Orientering om datamaskiner” in under huvudmoment i matematik för högstadiet. Inga anvisningar gavs dock om hur detta skulle kunna gå till. I början av 1970-talet inledde Skolöverstyrelsen en försöksverksamhet med datorn i skolan och projektet Datorn I Skolan (DIS) startades. En slutrapport på projektet lades fram 1979. Skolöverstyrelsen antog sedan ett handlingsprogram som vilade på slutrapporten. I handlingsprogrammet fastslogs bland annat att grundskolans undervisning om ”datorer och datorers användning i samhället – datalära” skulle bedrivas inom So-ämnena och matematik.

Den andra fasen började läsåret 1982 då datalära infördes på högstadiet i samband med Lgr 80. I Lgr 80 lyftes datorn ur tre olika aspekter:

- demokrati – alla elever skulle känna till datorns användningsområden i samhället
- arbetsliv – undervisningen skulle beröra yrken där datorn används
- arbetsredskap – datorn skulle vara ett stöd vid inläring

Under 1980- och 1990-talet gjorde Skolöverstyrelsen flera satsningar på att integrera datorn som stöd i undervisningen bland annat en treårig satsning på datorn som pedagogiskt hjälpmedel. Resultatet av satsningens utvärdering visade bland annat att den mesta datateknikanvändningen i skolan inriktades mot färdighetsträning och ganska enkla datortillämpningar (Riis, 1991). Dessutom lyftes att datorn, som tekniskt och pedagogiskt hjälpmedel, ökade tillgängligheten för elever med olika typer av funktionsnedsättning.

Den tredje fasen kan kopplas till IT-användningen i den svenska skolan. Skolverket fick sitt första uppdrag inom IT-området 1992. I uppdraget betonades att man skulle stimulera användningen av datorn som ett redskap bland andra. Datorn skulle användas i planering, genomförande och uppföljning av undervisningen av såväl lärare som elever. Inom ramen för detta uppdrag genomfördes flera stora framgångsrika satsningar: Skoldatanätet, IT i Skolan (ITiS), satsningen på European Schoolnet och de skolprojekt som initierats av Stiftelsen för kunskaps- och kompetensutveckling (KK-stiftelsen). Sedan mitten av 1990-talet har tekniken utvecklats ytterligare. Datorerna har blivit kraftfullare och mycket billigare. Trådlösa nätverk blev vanligare vilket bland annat ledde till att CD-ROM-baserade material, till exempel spel, ersattes med interaktiva och nätbaserade tjänster. Många lärare och elever fick varsin dator vilket gjorde det möjligt att använda IT som ett naturligt verktyg på lektioner och därmed på riktigt i undervisning.

Vi är nu i en fjärde fas som karakteriseras av att omfatta alla elever, vilket inte var fallet i tidigare försök med datorer i skolan. Numera är fokus på att utveckla elevers kunskap i programmering som en del av den digitala kompetensen.

År 2006 släppte EU ett ramverk kring digital kompetens, Digital Competence Framework for citizens, DigComp 2.0 (Vuorikari, m.fl., 2016). I ramverket tydliggörs de digitala kompetenserna som en medborgare behöver ha för att kunna verka i ett digitalt samhälle. Utifrån detta ramverk beskrivs, i Skolverkets kommentarmaterial (Skolverket, 2017), den digitala kompetensen utifrån fyra aspekter:

- att förstå digitaliseringens påverkan på samhället
- att kunna använda och förstå digitala verktyg och medier
- att ha ett kritiskt och ansvarsfullt förhållningssätt
- att kunna lösa problem och omsätta idéer i handling

Begreppen digitala verktyg och digitala medier

Ett **digitalt verktyg** kan vara både en fysisk enhet, till exempel en dator, smart telefon, lärplatta eller en kamera, men ett digitalt verktyg kan även vara en programvara eller en internetbaserad tjänst. För de flesta är begreppen datorplatta eller surfplatta mer använda än begreppet lärplatta, men inom utbildning används gärna det senare för att betona att syftet är att underlätta och förstärka lärandet. Vilket begrepp vi väljer beror därför på sammanhanget.

När det gäller programmering sker en ständig utveckling, exempelvis utvecklas programspråk eller ersätts av nya. I läro- och kursplanerna nämns därför inte specifika programspråk eller programmeringsmiljöer utan eleverna ska genom undervisningen

utveckla en generell förståelse för programmering. Denna förståelse kan skapas genom att till exempel utgå från ett visst språk eller en viss programmeringsmiljö. Eleverna har sedan möjlighet att använda sina generella kunskaper om och erfarenheter av programmering för att gå vidare i sin kunskapsutveckling. Vissa forskare menar att den moderna utvecklingen av teknik och nyare programmeringsmiljöer betyder att eleverna kan lägga mer fokus på de logiska tankeprocesserna som krävs för programmering än på färdigheter i att kunna programmera genom att använda ett visst programspråk (Rolandsson, 2015).

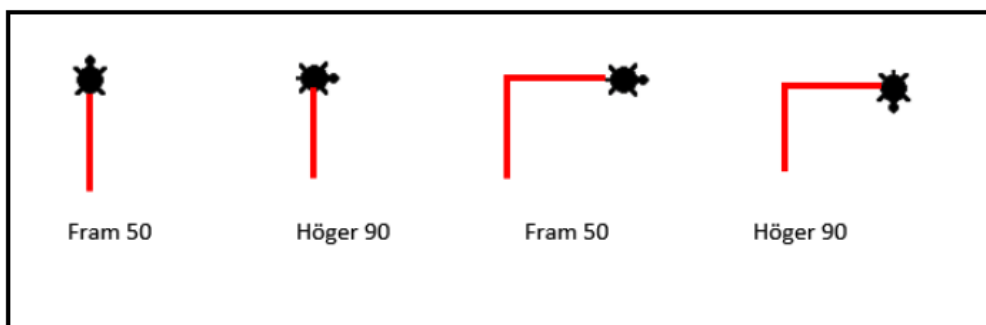
Med **digitala medier** avses plattformar och verktyg som kan användas för kommunikation och skapande, till exempel e-tidningar, webbtjänster eller specifika appar (Bers, 2020).

Programmering och matematikundervisning

Under 1980-talet utvecklades de första programmeringsmiljöerna som anpassades för programmering i skolan. En central person var Seymour Papert från Massachusetts Institute of Technology (MIT) i Boston, USA. Papert var matematiker, datavetare och pedagog och utvecklade programspråket Logo (Papert, 1980). En av de centrala delarna i Logo var Turtle graphics. Där skulle eleverna styra en sköldpadda (turtle) med tre tekniker: ange en position, ange en riktning och använda en penna. Pennan hade en färg och en bredd och kunde lyftas upp och ner. Genom att ge enkla instruktioner om hur sköldpaddan skulle flytta sig och ändra pennans egenskaper kunde eleverna få datorn att rita upp olika grafiska spår. Se exempel i Figur 1.

Figur 1

Sköldpadda programmerad för att skapa en geometrisk form



Genom att introducera hur entydiga stegvisa instruktioner kan uttryckas och följas ges eleverna möjlighet att utveckla förståelse för hur programmering kan användas. Det kan till exempel handla om att eleverna får tydliga stegvisa instruktioner för att utföra vissa bestämda rörelser eller förflytta sig i en bestämd bana. Genom innehållet symbolers användning vid stegvisa instruktioner möter eleverna även de symboler som utgör en del

av det programspråk som eleverna senare kan möta. Här kan eleverna till exempel inledningsvis få möjlighet att själva komma överens om vilka symboler som behövs för att beteckna ett antal upprepningar. De stegvisa instruktionerna kan ges med hjälp av till exempel pilar, ord, bilder, siffror eller bokstäver som läggs eller skrivs i en viss ordning. Fokus blir på vikten av att instruktioner ges i rätt ordning. Om en sekvens av instruktioner upprepas, formas dessa instruktioner till en loop.

Det finns forskningsresultat som visar att många lärare ser stora kunskapsvinster i att låta eleverna arbeta med analoga programmeringsövningar (Sung, m.fl., 2017). Resultatet visar att elever som i hög grad genomfört övningar av praktisk karaktär skapade program med större noggrannhet, än elever som deltagit i undervisning med lägre grad av praktiska övningar. Forskarna påpekar dock att läraren har en avgörande roll eftersom läraren kan utforma undervisningen genom att arbeta på många olika abstraktionsnivåer och använda olika undervisningsmetoder.

Tinkering

Ett flertal forskare hävdar att en lärandemiljö som tillåter eleverna att testa saker och idéer i en ostrukturerad process, **tinkering**, gynnar lärandet av programmering samt skapar nyfikenhet och kreativitet (Quan & Gupta, 2018). Tinkering är både en process och en inriktning och kännetecknas av lekfullhet där eleven kontinuerligt utvärderar sina mål, utforskar nya vägar och föreställer sig nya möjligheter.

Inom didaktisk forskning presenteras två olika sidor av tinkeringprocessen, dels den fria aktiviteten där elever skapar eller ändrar ett befintligt datorprogram, dels aktiviteter där eleverna får testa, göra fel, samt ge och få feedback från andra elever, lärare eller dator. Det finns forskare som gör en distinktion mellan tinkers och planerare (Blikstein, m.fl., 2014). Tinkers testar och gör en serie stegvisa förändringar i sina program för att skapa en färdig lösning. Planerare däremot är mer systematiska, de tar fram och genomför en handlingsplan för att kunna göra slutliga ändringar i skapandet av en algoritm eller ett program. Att låta eleverna utvecklas som både tinkers och planerare skapar möjligheter för dem att förstå olika faser i programmeringsprocessen. Användningen av analog eller digital programmering kan stödja både tinkers och planerare eftersom miljön erbjuder elever möjligheter att både testa sig fram och gå systematiskt tillväga. Detta spelar en central roll i vad eleverna har möjlighet att förstå när programmering används i matematikundervisningen.

Tinkering tillåter att idéer kolliderar och det är just i dessa kollisionpunkter som kreativitet uppstår. Eleverna får självtillit genom att de tillåts experimentera, ta risker och utforska egna idéer (Quan & Gupta, 2018). Detta leder till att de börjar se sig själva som programmerare, med egna bra idéer som de kan förvandla till verklighet. Detta i sin tur kan bidra till att eleverna utvecklar färdigheter som kan användas senare i

programmering. Dessa färdigheter är centrala dimensioner för att lära sig programmera och inkluderar:

- Vana vid att hantera komplexitet
- Uthållighet vid arbete med svåra eller stora problem
- Tolerans för tvetydighet eller osäkerhet
- Förmåga att hantera öppna problemuppgifter
- Förmåga att kommunicera och samarbeta med andra för att komma fram till en gemensam lösning

Tinkering är nära förknippad med logiskt resonemang, eftersom eleverna testar sina instruktioner och skapar erfarenheter som handlar om orsak och effekt. I undervisningen bör eleverna därför få testa sina instruktioner i utforskande lekbaserade aktiviteter som baseras på analog eller digital programmering. Så småningom blir denna utforskning målmedveten genom prövning och förbättring.

Programmering genom pedagogisk lek

Wood (2014) identifierar tre förhållningssätt till den pedagogiska leken. Leken kan vara:

- elevinitierad: lek initieras av eleven trots att den är begränsad av vilka val som erbjuds
- lärarguidad: elevens lek värderas utifrån att den är till nytta för lärande och utveckling
- tekniskt inriktad: lek och lärande är något instrumentellt, det vill säga läroplansstyrt

Dessa tre förhållningssätt är användbara i programmering och kan kopplas till både analog och digital programmering. Programmeringen kan innebära att elever själva får initiera, skapa, testa och utveckla enkla spel i en tinkeringprocess. I spel, utvecklar eleverna, förutom sin sociala och språkliga kompetens, även sin koncentrationsförmåga och sin kommunikativa kompetens. Genom att låta elever programmera enkla spel, som de känner igen från vardagen, ges utrymme för egna beslut, nya insikter och eleven utmanas i sitt eget tänkande (Parekh & Gee, 2018).

Referenser

Bers, M. U. (2020). *Coding as a playground: Programming and computational thinking in the early childhood classroom*. Routledge. <https://doi.org/10.4324/9781003022602>

Blikstein, P., Worsley, M., Piech, C., Sahami, M., Cooper, S., & Koller, D. (2014). Programming pluralism: using learning analytics to detect patterns in the learning of

computer programming. *Journal of the Learning Sciences*, 23(4), 561–599.
<https://doi.org/10.1080/10508406.2014.954750>

Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. Basic Books.

Quan, M. G., & Gupta, A. (2018). Tensions in the productivity of design task tinkering. *Journal of Engineering Education*, 20(109), 88–106. <https://doi.org/10.1002/jee.20303>

Riis, U. (1991). *Skolan och datorn: satsningen Datorn som pedagogiskt hjälpmedel 1988–1991*. Tema Teknik och social förändring, Rapport 24, Universitetet i Linköping.

Sung, W., Ahn, J., & Black, J. B. (2017). Introducing computational thinking to young learners: practicing computational perspectives through embodiment in mathematics education. *Technology, Knowledge and Learning*, 22(3), 443–463.
<https://doi.org/10.1007/s10758-017-9328-x>

Wood, E. (2014). The play-pedagogy interface in contemporary debates. I L. Brooker, M. Blaise & S. Edwards (Red.), *Handbook of play and learning in early childhood* (s. 145–156). Sage.

Matematik – anpassad grundskola åk 7–9
Modul: Programmering i anpassad grundskola
Del 2: Datorer och programmering i undervisningen – då och nu

Aktiviteter – följa instruktioner

Constanta Olteanu, Linnéuniversitetet

Helen Larsson & Ida Ekeback, Rosenlundsskolan Jönköping

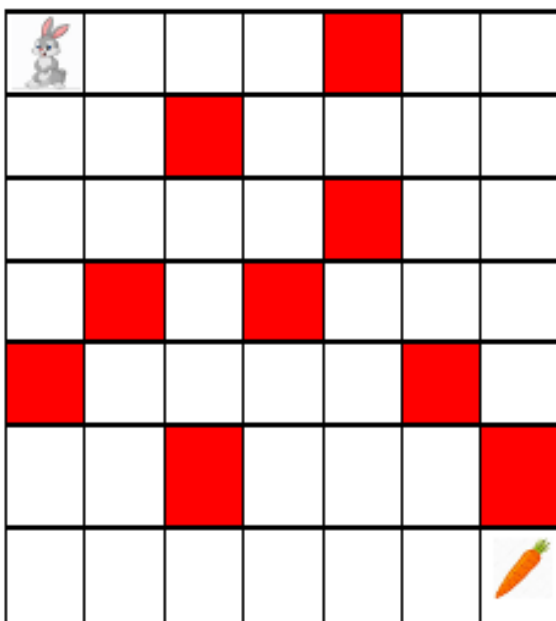
Analog

Aktivitet: Labyrintspel

Syfte: Genomföra en aktivitet med tinkering i fokus.

Matematiskt innehåll: Lägesord, geometriska figurer, enkla mönster.

Förberedelser: Kopiera labyrinten och tabellen i aktiviteten nedan. Varje elev behöver två kopior.



Symbol	Förklaring

Genomförande: Förklara för eleverna att kaninen vill komma fram till moroten, men att den inte får hoppa över eller stanna på de röda fälten.

Eleverna börjar med att arbeta individuellt. Varje elev antecknar i tabellen den valda vägen. Eleven skapar egna symboler för detta ändamål (exempelvis pilar och siffror).

Därefter arbetar eleverna i par och byter tabeller med varandra. Den ena eleven förklarar innebörden i symbolerna som användes i tabellen för den andra eleven. Den andra eleven följer instruktionerna som den förste eleven har skapat.

Därefter byter eleverna rollerna.

Diskutera:

- Går kaninen automatiskt framåt ett steg när den svänger eller står den kvar i samma ruta och svänger?
- Hur många olika vägar finns det?
- Vilken är den kortaste vägen?
- Vilken väg måste kaninen svänga flest gånger?

Digital

Aktivitet: Skapa en blinkande symbol

Syfte: Genomföra en aktivitet med digitala verktyg samt med tinkering i fokus.

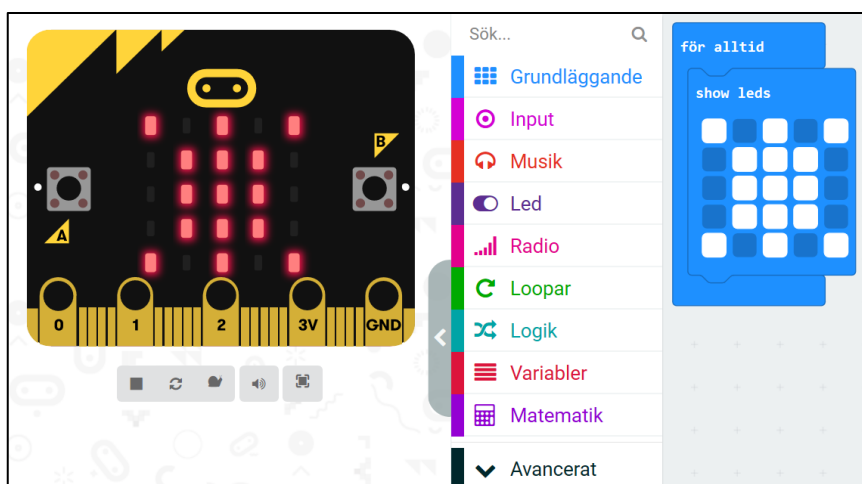
Matematiskt innehåll: Enkla mönster.

Utrustning: Digitalt verktyg.

Förberedelser: I aktiviteten nedan används programmeringsmiljön micro:bit. Gå till länken: tiigbg.se/koda/a1

Skriv ut mallar som elever kan ha till hjälp när de själva skapar en symbol. Mallarna ska ha en 5x5 pixlig ruta.

En del elever kan även behöva mallar som visar hur en symbol kan se ut. Skapa och skriv ut.



Genomförande: Eleven ska, genom att testa och prova sig fram, komma underfund med hur man kan få micro:bit att visa en egen ritad symbol och att sedan försöka att få den att blinka.

Diskutera:

- Hur kom ni på hur ni skulle göra?
- Hur gjorde ni för att få er micro:bit att visa en symbol?
- Hur gjorde ni för att få symbolen att blinka.
- Var det något som var extra klurigt och som ni inte förstod?

Fördjupning: Arbeta med if-satser på micro:bit. Till exempel genom att programmera micro:bit till att göra något annat efter att symbolen har blinkat tio gånger.

Del 2: Moment B – kollegialt arbete

Diskutera

- I texten ges en kort historisk bild av införandet av datorer i skolan. Vilka egna erfarenheter, som elev/lärare, har ni av detta?
- Vilka möjligheter och utmaningar ser ni med införandet av digitala verktyg i matematikundervisningen?
- Hur förhåller sig tinkering till den matematikundervisning som ni har idag?

Förbered en aktivitet/lektion

Förbered en aktivitet med tinkering i fokus. Låt elevernas kreativitet, upptäckarlust och nyfikenhet vara drivkraften i aktiviteten. Använd något av de förslag som presenteras i dokumentet "Aktiviteter – följ instruktioner" eller välj en egen aktivitet. Anpassa aktiviteten till era grupper och diskutera hur ni rent praktiskt ska genomföra den. Aktiviteten kan genomföras analogt eller digitalt beroende på era elevers erfarenheter och förkunskaper.

Del 2: Moment C – aktivitet

Genomför aktiviteten i era klasser/grupper. Observera på vilket sätt eleverna tar sig an uppgiften och vilka strategier de använder för att komma fram till en lösning. Gör korta anteckningar och ta med till moment D.

Del 2: Moment D – gemensam uppföljning

Diskutera

- Hur fungerade aktiviteten? Vilka anpassningar behövde ni göra?
- Vilka tekniska och matematiska begrepp behövde förklaras?
- Hur påverkade arbetsformen, tinkering, elevernas möjligheter att lära sig?
- Vad bör ändras om ni vill använda samma aktivitet en annan gång?

Reflektera

Avsätt de sista tio minuterna för att reflektera över:

- Vad gjorde jag/vi?
- Vad lärde jag mig?

Sammanfatta tillsammans arbetet med denna del i några punkter. Anteckna också de idéer ni vill använda i kommande arbete och tänk på att spara era anteckningar eftersom ni kommer att använda dem i del 4.