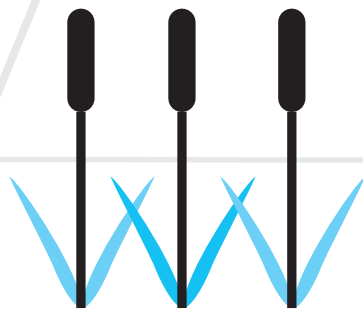
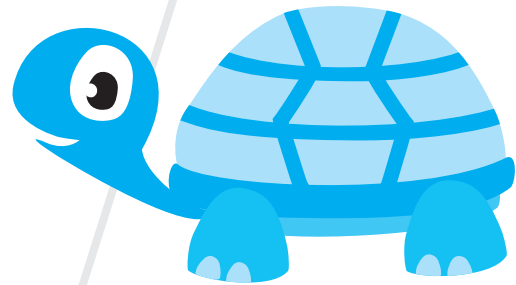
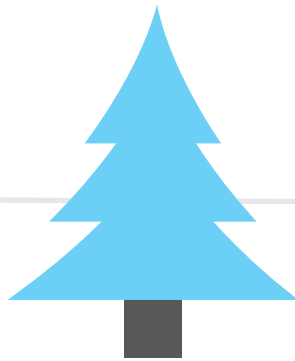


Programmering

KOMPLEMENT TILL MATTE DIREKT BORGEN UPPLAGA 2

Lektion 1



Sanoma Utbildning

POSTADRESS: Box 30091, 104 25 Stockholm

BESÖKSADRESS: Alströmergatan 12, Stockholm

HEMSIDA: www.sanomautbildning.se

E-POST: info@sanomautbildning.se

Order/läromedelsinformation

TELEFON: 08-587 642 10

TELEFAX: 08-587 642 02

Författare och redaktör: Linda Kempe

Grafisk form och illustrationer: Anna Hild

©2019 Sanoma Utbildning AB, Stockholm

Kopiering tillåten.

Till er som använder Programmering

Programmering är en komplettering till basläromedlet Matte Direkt Borgen åk 4-6. Kompletteringen innehåller analoga programmeringsuppgifter och följer Lgr11 reviderad 2018.

Materialet är uppbyggt med förslag på färdiga lektionsupplägg. *Programmering* är ett kopieringsunderlag och innehåller instruktioner till läraren, arbetsblad till eleverna samt facit. Under 2019 kommer flera lektioner läggas ut kontinuerligt.

I instruktionerna till läraren finns:

- Uppskattad tidsåtgång
- Moment ur Lgr11 reviderad 2018
- Arbetsgång. Rubrikerna A, B, C och D informerar om arbetsgång och kan skrivas på tavlan som instruktion till eleverna
- Extra uppgifter. Dessa kan variera i sin karaktär men består oftast av förslag på gemensam aktivitet, öva mera och/eller utmaning
- Facit

Arbetsbladens upplägg:

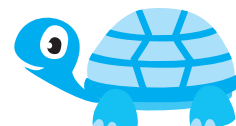
- Begreppsutor och faktarutor
- En progression mellan uppgifterna
- Eleverna får olika uppgifter där de ska hjälpa Turtle att ta sig fram i en visuell analog programmeringsmiljö

Lycka till!

Linda Kempe

Sanoma Utbildning

Hej!
Det är jag som
heter Turtle.



Skriva kod och hitta buggar

LEKTION

1

Tidsåtgång: ca 90 min.

I slutet av lektionen finns förslag på extra uppgifter som du kan använda vid behov eller vid andra tillfällen.

Moment ur Lgr11 reviderad 2018:

Algebra: Hur algoritmer kan skapas och användas vid programmering.
Programmering i visuella programmeringsmiljöer.

A. Gemensam introduktion

1. Diskussion: Vad i klassrummet är programmerat?

EXEMPEL: dator, bildskärm, tv, fjärrkontroll, tidsinställt fläktsystem. En del elever kan nämna datorspel och associerar programmering till att få programmera egna spel på dator. Här är det viktigt att behålla dessa elevers entusiasm samtidigt som du belyser att vi under lektionerna kommer att lära oss grunderna i programmering.

2. Praktisk övning: illustrering av "exakta instruktioner".

Material: Klossar.

INTRODUKTION: Berätta för eleverna vad en algoritm är. Du styr en dator med instruktioner som kallas algoritmer. Algoritmerna består av exakta steg-för-steg-instruktioner, som kallas för kod.

UPPGIFTEN: Programmeraren och maskinen. Dela in eleverna två och två. Placera dem rygg mot rygg och med varsin uppsättning av lika många klossar. Den som först är programmeraren bygger en konstruktion av sina klossar och beskriver för maskinen hur konstruktionen ser ut. Maskinen bygger en likadan av sina klossar. När eleverna är klar jämför de konstruktionerna med varandra. Blev de likadana? Var instruktionerna tydliga? Eleverna byter roller och upprepar uppgiften.

REFLEKTION ENSKILT/I PAR/ALLA Liknar konstruktionerna varandra? Om ja, vad bidrog till likadana konstruktioner? (Svar: exakta instruktioner/exakt kod) Om nej, vad kunde eleven gjort annorlunda? (svar: tydligare instruktioner/kod). Om konstruktionerna ser olika ut så innehöll koden fel instruktioner/buggar. Genom att gå tillbaka och rätta instruktionerna blir koden tydlig, buggarna försvinner och konstruktionerna blir likadana.

SAMMANFATTNING Vad lärde vi oss? Vi lärde oss att vi måste vara tydliga med de stegvisa instruktionerna. Samma sak gäller vid programmering av ett datorprogram.

3. Förstärk begreppen:

Skriv begreppen på tavlan. Detta underlättar för eleverna när de arbetar vidare med uppgifterna i arbetsbladet.

Algoritmer består av exakta steg-för-steg-instruktioner, som kallas för kod. Varje kod betyder något för datorn. Om datorprogrammet inte kan tolka eller tolkar instruktionerna fel, har det blivit en bugg. I denna praktiska övning använde vi oss av det talade språket som kod, men koden kunde även ha bestått av symboler. Datorprogrammen som vi använder oss av är skriven med programmeringsspråk.

B. Arbetsblad

Material: Blyertspennor och arbetsblad "Lektion 1".

I arbetsbladen får eleverna följa sköldpaddan Turtle i en analog programmeringsmiljö. Turtle kommer med sju olika uppgifter som eleverna ska utföra.

- *Uppdrag 1 och 2:* Eleverna lär sig "sväng höger" och "sväng vänster".

Förtydliga för eleverna: Koden "sväng höger" innebär att figuren roterar 90° runt sin egen axel. Den tar alltså inte ett steg till höger, vilket är en vanlig missuppfattning.

Skriva kod och hitta buggar LEKTION 1

En dator styrs med kommandon som kallas algoritmer. Algoritmerna består av exakta steg-för-steg-instruktioner som kallas för kod. Steg kan beskrivas som följande: Gå till höger/benken 65 exempel att klossar står på sin plats och vänta på 90° till höger.

Hjälj jag heter Turtle. Gå till mig framåt. Gå till mig framåt. Gå till mig framåt. Gå till mig framåt.

UPPGIFT 1: Att vilket håll ska sköldpaddan Turtle svänga för att komma fram till öjnet?

STRUKTUR: Börja med att läsa igenom alla regler med kod. Fyll i den vita koden på den tomma sköldpaddan.

Skriv till för sväng höger och V för sväng vänster.

- 1 Gå två steg framåt
- 2 Sväng _____
- 3 Gå tre steg framåt
- 4 Sväng _____
- 5 Gå två steg framåt

UPPGIFT 2: Att vilket håll ska Turtle svänga för att komma fram till öjnet?

Skriv till för sväng höger och V för sväng vänster.

- 1 Sväng _____
- 2 Gå två steg framåt
- 3 Sväng _____
- 4 Gå två steg framåt
- 5 Sväng _____
- 6 Gå tre steg framåt
- 7 Sväng _____
- 8 Gå tre steg framåt
- 9 Sväng _____
- 10 Gå två steg framåt
- 11 Sväng _____
- 12 Gå två steg framåt

Uppdrag 1 och 2

- **Uppdrag 3 och 4:** Eleverna läser kod, hittar bugg, skriver rätt kod.

Skriva kod och hitta buggar

När du ser en kod kan du se om den fungerar eller om den inte fungerar. Det är viktigt att kunna hitta buggar. Du kan se om du kan hitta buggar i koden och om du kan skriva rätt kod.

UPPDRAG 3: Hjälp Turtle att hitta till sitt hus. Hänga in buggen. Skriv rätt kod på skrivraden.

INSTRUKTION: Börja med att läsa igenom alla rader med kod. Hänga in buggen. Skriv in rätt kod.

Programmerarens kod: _____ Din kod: _____

1. Gå två steg framåt
2. Sväng vänster
3. Gå fem steg framåt
4. Sväng höger
5. Gå två steg framåt

UPPDRAG 4: Hjälp Turtle att hitta till sitt hus. Läs igenom all kod. Hänga in buggen och skriv rätt kod på den tomma skrivraden.

Programmerarens kod: _____ Din kod: _____

1. Gå två steg framåt
2. Sväng höger
3. Gå två steg framåt
4. Sväng höger
5. Gå två steg framåt

När du läser igenom koden och hittar en bugg, skriv den i den tomma rutan. Du kan se om du kan skriva en algoritm till den buggen.

Uppdrag 3 och 4

- **Uppdrag 7:** Eleverna behöver läsa alla kod, resonera och avgöra vilken kod som ger Turtle den kortaste vägen hem.

Skriva kod och hitta buggar

UPPDRAG 7: Turtle är svängig och behöver hjälp att hitta till sitt hus. Vilken Algoritm är rätt om du uppgift är att hitta den kortaste vägen för Turtle? Hänga in den rätta algoritmen.

ALGORITHM 1:

1. Sväng vänster
2. Gå fyra steg framåt
3. Sväng höger
4. Gå fyra steg framåt
5. Sväng höger
6. Gå ett steg framåt
7. Sväng höger
8. Gå ett steg framåt

ALGORITHM 2:

1. Gå fyra steg framåt
2. Sväng vänster
3. Gå tre steg framåt
4. Sväng höger
5. Gå ett steg framåt

ALGORITHM 3:

1. Sväng vänster
2. Gå fyra steg framåt
3. Sväng vänster
4. Gå fyra steg framåt
5. Sväng höger
6. Gå ett steg framåt
7. Sväng höger
8. Gå ett steg framåt

ALGORITHM 4:

1. Gå fem steg framåt
2. Sväng vänster
3. Gå tre steg framåt
4. Sväng höger
5. Gå två steg framåt

Uppdrag 7

- **Uppdrag 5 och 6:** Eleverna läser kod, hittar flera buggar, skriver rätt kod. Eleverna skriver egen kod med buggar, löser varandras kod och rättar koden.

Tips: Bestäm gärna paren i förväg så att ingen blir utanför.

Skriva kod och hitta buggar

UPPDRAG 5: Turtle har varit och badat vid sjön och ska nu gå hem för att äta middag. Kontrollera koden så att Turtle hittar hem till sitt hus. Ändra kod när det behövs.

Programmerarens kod: _____ Din kod: _____

1. Gå fem steg framåt
2. Sväng vänster
3. Gå två steg framåt
4. Sväng höger
5. Gå tre steg framåt
6. Sväng vänster
7. Gå tre steg framåt
8. Sväng höger
9. Gå ett steg framåt

UPPDRAG 6: Skriv en egen algoritm med buggar. Skriv minst fem rader med kod på skrivraden här nedanför, om du kallar algoritmen. Din kod ska innehålla två buggar, som inte får komma efter varandra. Buggen ska vara en bugg som inte ska lösas. Du ska kunna hitta dina buggar och skriva in rätt kod.

INSTRUKTIONER TILL DE KLASSMÄNNOR: Läs uppdraget. Läs igenom koden, ringa in buggarna och skriv in rätt kod på skrivraden.

Din kod: _____ Klassmännans kod: _____

1. _____
2. _____
3. _____
4. _____
5. _____
6. _____
7. _____

INSTRUKTIONER TILL DE KLASSMÄNNOR:

- Vad hade du en uppgift?
- Turtle är det viktigt att hitta och andra buggar?
- Hur ser det ut med buggar? Vad hade det?

Uppdrag 5 och 6

C. Genomgång av arbetsblad

D. Gemensam reflektion

- Vilka **begrepp** har vi lärt oss idag?
Svar: algoritm, kod och bugg.
- Kan ni ge exempel på begreppet **algoritm**?
Svar: exakta steg-för-steg-instruktioner.
- Kan ni ge exempel på vad begreppet **kod** innebär?
Svar: en instruktion.
- Kan ni ge exempel på vad begreppet **bugg** innebär?
Svar: en felaktig del i en kod.
- Vad har vi lärt oss idag?
Svaren kan variera.

EXTRA UPPGIFTER:

Material: Penna, papper eller räknehäftet.

AKTIVITET Gemensam uppgift. Tusenfoting. Alla elever ställer sig i en kö och ser framåt. Eleven som står först i kön vänder sig om med ansiktet mot de andra. Uppgiften för den eleven är att visa rörelser och säga instruktioner så att tusenfotingen/eleverna rör sig lika.

Utmaning: försök att enbart säga instruktioner.

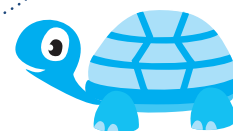
ÖVA MERA Enskild-/paruppgift. Skriv egna algoritmer. Eleverna skriver ner en algoritm, det vill säga den exakta steg-för-steg-instruktionen, för hur man borstar tänder, sitter ner på en stol eller tar på sig gummistövlar. Eleverna avslutar med att testa varandras algoritmer.

UTMANING Enskild-/paruppgift. Skriv egna algoritmer. Eleverna skriver ner algoritmen för 11+14. Algoritmen ska innehålla exakta steg-för-steg-instruktioner för hur de löser uppgiften. Eleverna diskuterar och jämför sina algoritmer. Ändra vid behov.

Skriva kod och hitta buggar

En dator styrs med instruktioner som kallas **algoritmer**. Algoritmerna består av exakta steg-för-steg-instruktioner, som kallas för **kod**. Varje kod betyder något för datorn. Sväng höger betyder till exempel att figuren står på sin plats och vrider sig 90° till höger.

Hej! Jag heter Turtle.
Hjälp mig med de olika
uppgifterna genom att
lösa uppdragen.



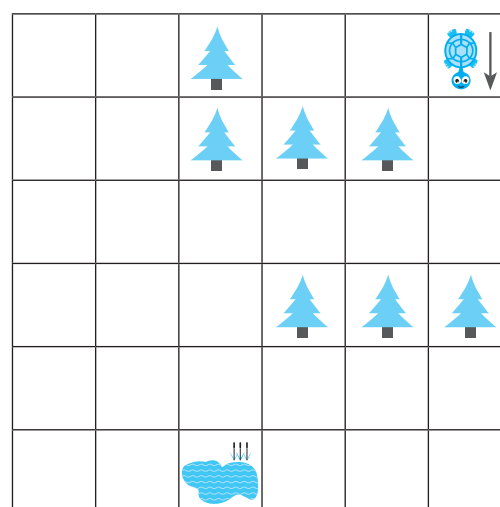
UPPDRAG 1: Åt vilket håll ska sköldpaddan Turtle svänga för att komma fram till sjön?

STRATEGI

Börja med att läsa igenom **alla** rader med kod.
Fyll i den rätta koden på den tomma skrivraden.

Skriv H för sväng höger och V för sväng vänster.

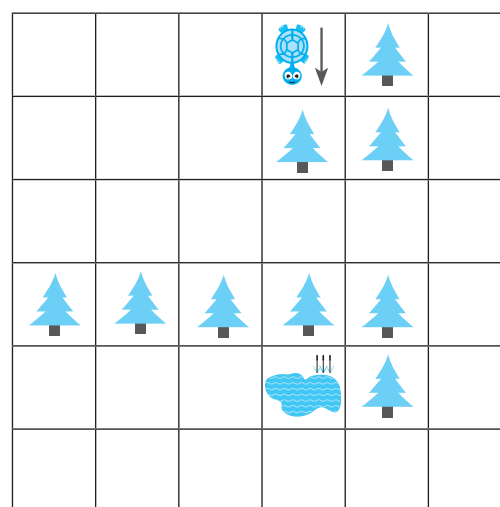
- 1 Gå två steg framåt
- 2 Sväng _____
- 3 Gå tre steg framåt
- 4 Sväng _____
- 5 Gå tre steg framåt



UPPDRAG 2: Åt vilket håll ska Turtle svänga för att komma fram till sjön?

Skriv H för sväng höger och V för sväng vänster.

- 1 Sväng _____
- 2 Gå ett steg framåt
- 3 Sväng _____
- 4 Gå två steg framåt
- 5 Sväng _____
- 6 Gå tre steg framåt
- 7 Sväng _____
- 8 Gå tre steg framåt
- 9 Sväng _____
- 10 Gå två steg framåt
- 11 Sväng _____
- 12 Gå ett steg framåt



Skriva kod och hitta buggar

När datorn inte kan läsa koden så fungerar inte datorprogrammet eller spelet.
Ett fel i en kod kallas för **bugg**. För att fixa den behöver programmeraren leta i koden
och hitta buggen. När datorprogrammet fungerar är koden rätt och buggen borta.

UPPDRAG 3: Hjälp Turtle att hitta till sitt hus. Ringa in buggen. Skriv rätt kod på skrivraden.

STRATEGI:

Börja med att läsa igenom **alla** rader med kod.

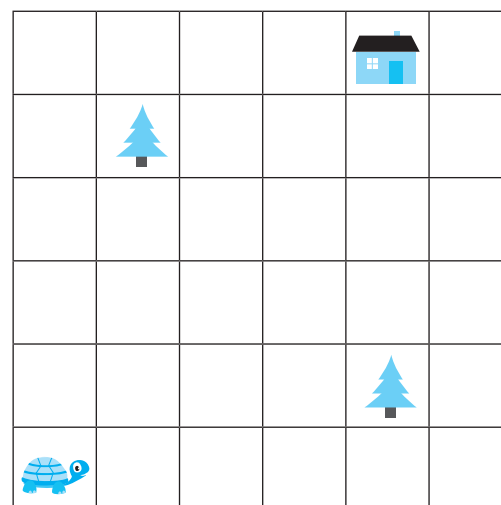
Ringa in buggen.

Skriv in rätt kod.

Programmerarens kod:

Din kod:

- 1 Gå två steg framåt
- 2 Sväng höger
- 3 Gå fem steg framåt
- 4 Sväng höger
- 5 Gå två steg framåt



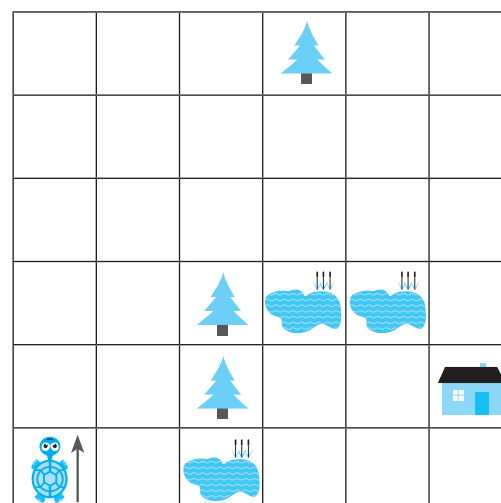
UPPDRAG 4: Hjälp Turtle att hitta till sitt hus. Läs igenom all kod.

Ringa in buggen och skriv rätt kod på den tomma skrivraden.

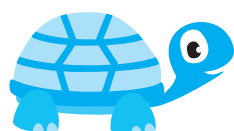
Programmerarens kod:

Din kod:

- 1 Gå tre steg framåt
- 2 Sväng höger
- 3 Gå två steg framåt
- 4 Sväng höger
- 5 Gå två steg framåt



När du läste igenom koden och
ändrade buggen, så blev koden en exakt
steg-för-steg-beskrivning.
Du har nu skapat en **algoritm**
till datorprogrammet.



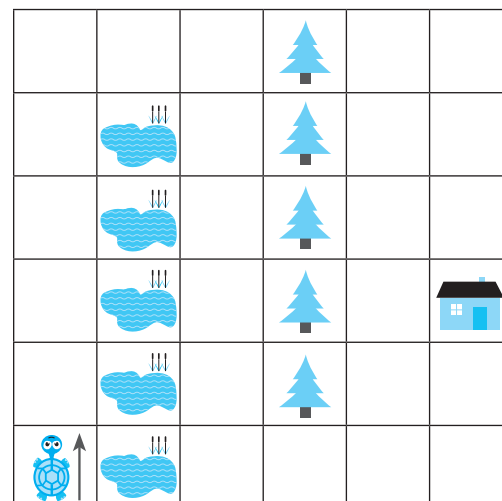
Skriva kod och hitta buggar

UPPDRAG 5: Turtle har varit och badat vid sjön och ska nu gå hem för att äta middag. Kontrollera koden så att Turtle hittar hem till sitt hus. Ändra kod när det behövs.

Programmerarens kod:

- 1 Gå fem steg framåt
- 2 Sväng vänster
- 3 Gå två steg framåt
- 4 Sväng höger
- 5 Gå tre steg framåt
- 6 Sväng vänster
- 7 Gå tre steg framåt
- 8 Sväng höger
- 9 Gå ett steg framåt

Din kod:



UPPDRAG 6: Skriv en egen algoritm med buggar.

Skriv minst fem rader med kod på skrivraderna här nedanför, en så kallad algoritm.

Din kod ska innehålla två buggar, som inte får komma efter varandra.

Rita in hinder och markera målet.

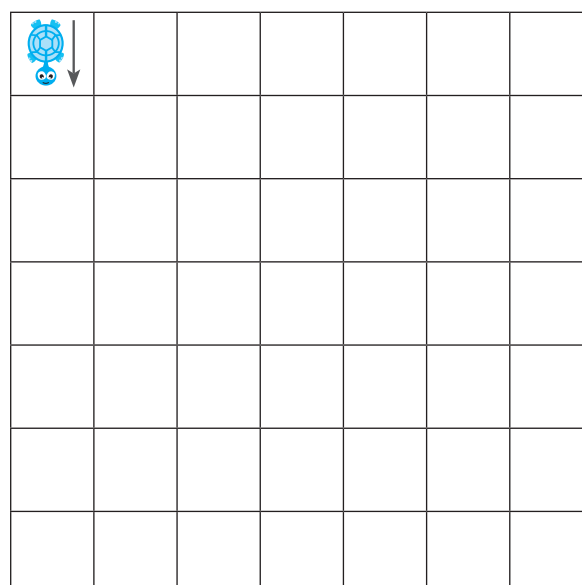
Be en klasskamrat att hitta dina buggar och skriva in rätt kod.

INSTRUKTIONER TILL DIN KLASSKAMRAT: Lös uppdraget. Läs igenom koden, ringa in buggarna och skriv in rätt kod på skrivraden.

Din kod:

- 1 _____
- 2 _____
- 3 _____
- 4 _____
- 5 _____
- 6 _____
- 7 _____

Klasskamratens kod:



DISKUTERA MED EN KLASSKAMRAT

- Vad lärde ni er av uppgiften?
- Varför är det viktigt att hitta och ändra buggar?
- Har ni varit med om buggar förut? Vad hände då?

Skriva kod och hitta buggar

UPPDRAG 7: Turtle är törstig och behöver hjälp att hitta till sjön. Vilken algoritm är rätt om din uppgift är att hitta den kortaste vägen för Turtle? Ringa in det rätta alternativet.

ALTERNATIV 1

- 1 Sväng vänster
- 2 Gå fyra steg framåt
- 3 Sväng höger
- 4 Gå fyra steg framåt
- 5 Sväng höger
- 6 Gå ett steg framåt
- 7 Sväng höger
- 8 Gå ett steg framåt

ALTERNATIV 2

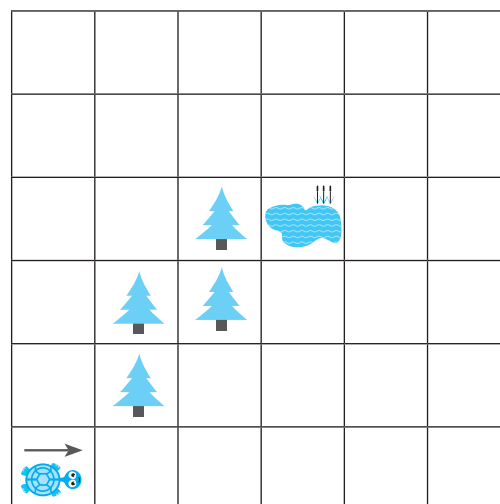
- 1 Gå fyra steg framåt
- 2 Sväng vänster
- 3 Gå tre steg framåt
- 4 Sväng vänster
- 5 Gå ett steg framåt.

ALTERNATIV 3

- 1 Sväng vänster
- 2 Gå fyra steg framåt
- 3 Sväng vänster
- 4 Gå fyra steg framåt
- 5 Sväng höger
- 6 Gå ett steg framåt
- 7 Sväng höger
- 8 Gå ett steg framåt.

ALTERNATIV 4

- 1 Gå fem steg framåt
- 2 Sväng vänster
- 3 Gå tre steg framåt
- 4 Sväng vänster
- 5 Gå två steg framåt.



FACIT översikt skriva kod och hitta buggar

Skriva kod och hitta buggar

LEKTION 1

En dator styrs med instruktioner som kallas algoritmer. Algoritmerna består av exakta steg-för-steg-instruktioner, som kallas för kod. Varje kod betyder något för datorn. Sväng höger betyder till exempel att figuren står på sin plats och vrider sig 90° till höger.



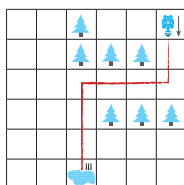
UPPDRAG 1: Åt vilket håll ska sköldpaddan Turtle svänga för att komma fram till sjön?

STRATEGI:

Börja med att läsa igenom alla rader med kod. Fyll i den rätta koden på den tomma skrivraden.

Skriv H för sväng höger och V för sväng vänster.

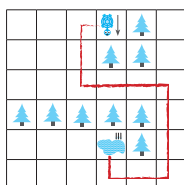
- Gå två steg framåt
- Sväng H
- Gå tre steg framåt
- Sväng V
- Gå tre steg framåt



UPPDRAG 2: Åt vilket håll ska Turtle svänga för att komma fram till sjön?

Skriv H för sväng höger och V för sväng vänster.

- Sväng H
- Gå ett steg framåt
- Sväng V
- Gå två steg framåt
- Sväng V
- Gå tre steg framåt
- Sväng H
- Gå tre steg framåt
- Sväng H
- Gå två steg framåt
- Sväng H
- Gå ett steg framåt



Matte Direkt Borgen - komplettering © Sanoma Utbildning 2018. Får kopieras. Lgr11: Hur algoritmer kan skapas och användas vid programmering i visuella programmeringsmiljöer.

1

Skriva kod och hitta buggar

LEKTION 1

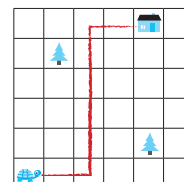
När datorn inte kan läsa koden så fungerar inte datorprogrammet eller spelet. Ett fel i en kod kallas för bugg. För att fixa den behöver programmeraren leta i koden och hitta buggen. När datorprogrammet fungerar är koden rätt och buggen borta.

UPPDRAG 3: Hjälp Turtle att hitta till sitt hus. Ringa in buggen. Skriv rätt kod på skrivraden.

STRATEGI:

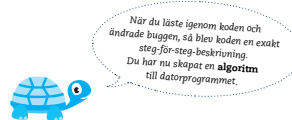
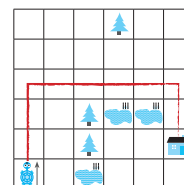
Börja med att läsa igenom alla rader med kod. Ringa in buggen. Skriv in rätt kod.

- Programmerarens kod: Din kod:
- Gå två steg framåt
 - Sväng höger
 - Gå fem steg framåt
 - Sväng höger
 - Gå två steg framåt
- VÄNSTER



UPPDRAG 4: Hjälp Turtle att hitta till sitt hus. Läs igenom all kod. Ringa in buggen och skriv rätt kod på den tomma skrivraden.

- Programmerarens kod: Din kod:
- Gå tre steg framåt
 - Sväng höger
 - Gå två steg framåt
 - Sväng höger
 - Gå två steg framåt
- FEM



Matte Direkt Borgen - komplettering © Sanoma Utbildning 2018. Får kopieras. Lgr11: Hur algoritmer kan skapas och användas vid programmering i visuella programmeringsmiljöer.

2

Uppdrag 1 och 2

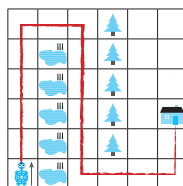
Uppdrag 3 och 4

Skriva kod och hitta buggar

LEKTION 1

UPPDRAG 5: Turtle har varit och badat vid sjön och ska nu gå hem för att äta middag. Kontrollera koden så att Turtle hittar hem till sitt hus. Ändra kod när det behövs.

- Programmerarens kod: Din kod:
- Gå fem steg framåt
 - Sväng vänster
 - Gå två steg framåt
 - Sväng höger
 - Gå tre steg framåt
 - Sväng vänster
 - Gå tre steg framåt
 - Sväng höger
 - Gå ett steg framåt
- HÖGER
- FEM
- VÄNSTER
- TVÅ



UPPDRAG 6: Skriv en egen algoritm med buggar.

Skriv minst fem rader med kod på skrivraderna här nedanför, en så kallad algoritm. Din kod ska innehålla två buggar, som inte får komma efter varandra. Rita in hinder och markera målet. Be en klasskamrat att hitta dina buggar och skriva in rätt kod.

INSTRUKTIONER TILL DIN KLASSKAMRAT: Lös uppdraget. Läs igenom koden, ringa in buggarna och skriv in rätt kod på skrivraden.

- Din kod: Klasskamratens kod:
- _____
 - _____
 - _____
 - _____
 - _____
 - _____
 - _____

DISKUTERA MED EN KLASSKAMRAT

- Vad lärde ni er av uppgiften?
- Varför är det viktigt att hitta och ändra buggar?
- Har ni varit med om buggar förut? Vad hände då?

Matte Direkt Borgen - komplettering © Sanoma Utbildning 2018. Får kopieras. Lgr11: Hur algoritmer kan skapas och användas vid programmering i visuella programmeringsmiljöer.

3

Skriva kod och hitta buggar

LEKTION 1

UPPDRAG 7: Turtle är törstig och behöver hjälp att hitta till sjön. Vilken algoritm är rätt om din uppgift är att hitta den kortaste vägen för Turtle? Ringa in det rätta alternativet.

ALTERNATIV 1

- Sväng vänster
- Gå fyra steg framåt
- Sväng höger
- Gå fyra steg framåt
- Sväng höger
- Gå ett steg framåt
- Sväng höger
- Gå ett steg framåt

ALTERNATIV 2

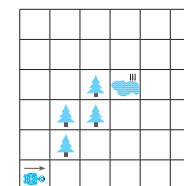
- Gå fyra steg framåt
- Sväng vänster
- Gå tre steg framåt
- Sväng vänster
- Gå ett steg framåt.

ALTERNATIV 3

- Sväng vänster
- Gå fyra steg framåt
- Sväng vänster
- Gå fyra steg framåt
- Sväng höger
- Gå ett steg framåt
- Sväng höger
- Gå ett steg framåt.

ALTERNATIV 4

- Gå fem steg framåt
- Sväng vänster
- Gå tre steg framåt
- Sväng vänster
- Gå två steg framåt.



Matte Direkt Borgen - komplettering © Sanoma Utbildning 2018. Får kopieras. Lgr11: Hur algoritmer kan skapas och användas vid programmering i visuella programmeringsmiljöer.

4

Uppdrag 5 och 6

Uppdrag 7

Skriva kod och hitta buggar

FACIT

LEKTION 1

En dator styrs med instruktioner som kallas **algoritmer**. Algoritmerna består av exakta steg-för-steg-instruktioner, som kallas för **kod**. Varje kod betyder något för datorn. Sväng höger betyder till exempel att figuren står på sin plats och vrider sig 90° till höger.

Hej! Jag heter Turtle. Hjälp mig med de olika uppgifterna genom att lösa uppdragen.



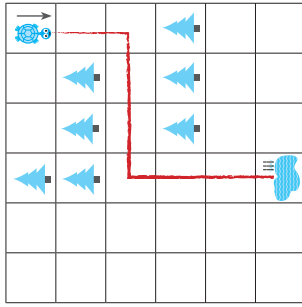
UPPDRAG 1: Åt vilket håll ska sköldpaddan Turtle svänga för att komma fram till sjön?

STRATEGI:

Börja med att läsa igenom **alla** rader med kod. Fyll i den rätta koden på den tomma skrivraden.

Skriv H för sväng höger och V för sväng vänster.

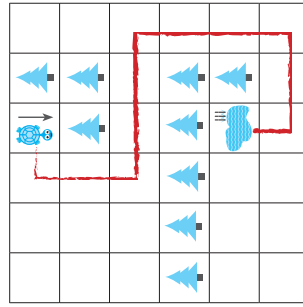
- Gå två steg framåt
- Sväng H
- Gå tre steg framåt
- Sväng V
- Gå tre steg framåt



UPPDRAG 2: Åt vilket håll ska Turtle svänga för att komma fram till sjön?

Skriv H för sväng höger och V för sväng vänster.

- Sväng H
- Gå ett steg framåt
- Sväng V
- Gå två steg framåt
- Sväng V
- Gå tre steg framåt
- Sväng H
- Gå tre steg framåt
- Sväng H
- Gå två steg framåt
- Sväng H
- Gå ett steg framåt



Skriva kod och hitta buggar

FACIT

LEKTION 1

När datorn inte kan läsa koden så fungerar inte datorprogrammet eller spelet. Ett fel i en kod kallas för **bugg**. För att fixa den behöver programmeraren leta i koden och hitta buggen. När datorprogrammet fungerar är koden rätt och buggen borta.

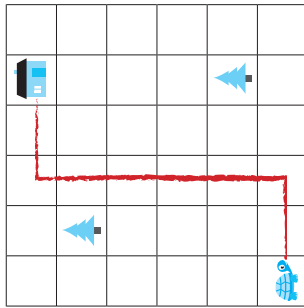
UPPDRAG 3: Hjälp Turtle att hitta till sitt hus. Ringa in buggen. Skriv rätt kod på skrivraden.

STRATEGI:

Börja med att läsa igenom **alla** rader med kod. Ringa in buggen. Skriv in rätt kod.

Programmerarens kod:

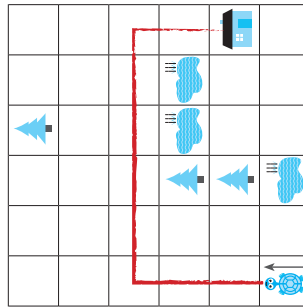
- Gå två steg framåt
- Sväng höger **Vänster**
- Gå fem steg framåt
- Sväng höger
- Gå två steg framåt



UPPDRAG 4: Hjälp Turtle att hitta till sitt hus. Läs igenom all kod. Ringa in buggen och skriv rätt kod på den tomma skrivraden.

Programmerarens kod:

- Gå tre steg framåt
- Sväng höger
- Gå två steg framåt **Fem**
- Sväng höger
- Gå två steg framåt



När du läste igenom koden och ändrade buggen, så blev koden en exakt steg-för-steg-beskrivning. Du har nu skapat en **algoritm** till datorprogrammet.



Skriva kod och hitta buggar

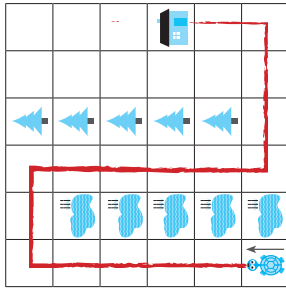
FACIT

LEKTION 1

UPPDRAG 5: Turtle har varit och badat vid sjön och ska nu gå hem för att äta middag. **Kontrollera koden** så att Turtle hittar hem till sitt hus. **Ändra kod** när det behövs.

Programmerarens kod:

- 1 Gå fem steg framåt
- 2 Sväng **vänster**
- 3 Gå två steg framåt
- 4 Sväng höger
- 5 Gå **tre** steg framåt
- 6 Sväng vänster
- 7 Gå tre steg framåt
- 8 Sväng **höger**
- 9 Gå **ett** steg framåt



UPPDRAG 6: **Skriv en egen algoritm med buggar.**

Skriv minst fem rader med kod på skrivraderna här nedanför, en så kallad algoritm.

Din kod ska innehålla två buggar, som inte får komma efter varandra.

Rita in hinder och markera målet.

Be en klasskamrat att hitta dina buggar och skriva in rätt kod.

INSTRUKTIONER TILL DIN KLASSKAMRAT: Lös uppdraget. Läs igenom koden, ringa in buggarna och skriv in rätt kod på skrivraderna.

Din kod:

- 1 _____
- 2 _____
- 3 _____
- 4 _____
- 5 _____
- 6 _____
- 7 _____



DISKUTERA MED EN KLASSKAMRAT

- Vad lärde ni er av uppgiften?
- Varför är det viktigt att hitta och ändra buggar?
- Har ni varit med om buggar förut? Vad hände då?

Matte Direkt Borgen – komplettering © Sanoma Utbildning 2018. Får kopieras.
Lgr11: Hur algoritmer kan skapas och användas vid programmering i visuella programmeringsmiljöer.

3

Skriva kod och hitta buggar

FACIT

LEKTION 1

UPPDRAG 7: Turtle är törstig och behöver hjälp att hitta till sjön. Vilken algoritm är **rätt om din uppgift** är att hitta den kortaste vägen för Turtle? Ringa in det **rätta alternativet**.

ALTERNATIV 1

- 1 Sväng vänster
- 2 Gå fyra steg framåt
- 3 Sväng höger
- 4 Gå fyra steg framåt
- 5 Sväng höger
- 6 Gå ett steg framåt
- 7 Sväng höger
- 8 Gå ett steg framåt

ALTERNATIV 2

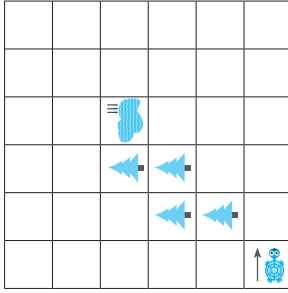
- 1 Gå fyra steg framåt
- 2 Sväng vänster
- 3 Gå tre steg framåt
- 4 Sväng vänster
- 5 Gå ett steg framåt.

ALTERNATIV 3

- 1 Sväng vänster
- 2 Gå fyra steg framåt
- 3 Sväng vänster
- 4 Gå fyra steg framåt
- 5 Sväng höger
- 6 Gå ett steg framåt
- 7 Sväng höger
- 8 Gå ett steg framåt.

ALTERNATIV 4

- 1 Gå fem steg framåt
- 2 Sväng vänster
- 3 Gå tre steg framåt
- 4 Sväng vänster
- 5 Gå två steg framåt.



Matte Direkt Borgen – komplettering © Sanoma Utbildning 2018. Får kopieras.
Lgr11: Hur algoritmer kan skapas och användas vid programmering i visuella programmeringsmiljöer.

4