
Instuderingsfrågor I inför tentan i kursen Algoritmer, ...

Svar till alla frågor kan hittas i [kursboken](#).

Frågorna omfattar **kap 1-3, sid 5-133** i kursboken.

Kap 1 Algoritmer och programmering, sid 5-60

1. Vilket var det första programmeringsspråk som utvecklades för de första datorerna?
Vilka egenskaper hade det? Vad är dess största skillnad till dagens programmeringsspråk?
2. Vad karaktäriserar de programmeringsspråk som kallades för *lågnivåspråk*? Varför "*låg*"?
3. Vilket var det första *högnivåspråket*? Varför "*hög*"?
4. Redogör för skillnaderna mellan begreppen *assemblering*, *kompilering* och *interpretering*.
5. Nämn ett exempel på programmeringsspråk som använde en av metoderna i fråga 4.
6. Vad var det första användningsområdet för programmering?
7. Finns det fortfarande kod som används som är skriven i något av de första programmeringsspråken?
Nämn några sådana samt deras användningsområde.
8. Vilket var det första programmeringsspråk som introducerade *kontrollstrukturer* i programmeringen?
9. Vad menas med *deklarativ programmering*? Är C# ett deklarativt språk?
10. Nämn några underkategorier till deklarativ programmering.
11. Vad karaktäriserar det som kallas för *spaghettiprogrammering*? Vad är huvudkritiken mot den?
12. Vilket programmeringstekniskt koncept kan ersätta spaghettiprogrammering?
13. Vad är den traditionella, procedurala synen på programmering som rådde på 60- och 70-talet?
14. Vad är den objektorienterade synen på programmering som kom upp på 80-talet?
15. Mellan vilka två programmeringsspråk går historiskt skiljelinjen mellan procedural och objektorienterad programmering? När ungefär inträffade övergången?
16. Vad var anledningen till paradigmskiftet inom programutveckling?
17. Vilka för- och nackdelar har enligt din åsikt den procedurala synen på programmering?
18. Vilka för- och nackdelar har enligt din åsikt den objektorienterade synen på programmering?
19. Definiera med egna ord begreppet *algoritm*. Nämn några exempel.
20. Vad är skillnaden mellan en algoritm och ett datorprogram?
21. Nämn några exempel på *beskrivning* av algoritmer. Vilka av dem är relevanta för programmering?
22. Vad är skillnaden mellan *instruktioner* och *villkor* i algoritmer?
23. Vilka är de tre grundläggande *kontrollstrukturerna* i alla procedurala programmeringsspråk?

24. Algoritmen för platsbyte kodas i [kursboken](#) på sid 44. Är programmet **MiniSort** objektorienterat?
25. Varför misslyckas försöket att modularisera **MiniSort**?
26. Beskriv med egna ord parameteröverföringsmetoden *värdeanrop* (*Call by value*).
27. Beskriv med egna ord parameteröverföringsmetoden *referensanrop* (*Call by reference*).
28. Vilka ändringar måste göras i koden, för att lyckas med modulariseringen av **MiniSort**?
29. Hur måste en parameter i en metod deklarerars för att bli en utparameter?
30. Vad är skillnaden mellan en metods returvärde och metodens utparametrar?
31. Varför (och när) behövs utparametrar, när man kan exportera data från en metod via returvärdet?
32. Hur kan man deklarerar en array som parameter i en metod?
33. Var någonstans måste storleken på en array anges när den skickas som parameter till en metod?
34. Vilken parameteröverföringsmetod använder C# automatiskt på en array som parameter?

Kap 2 Logik för blivande programmerare, sid 61-85

35. Varför är ämnet *Logik* relevant för programmering? I vilka delar av ett program används logik?
36. Vad är skillnaden mellan en *instruktion* och ett *villkor*?
37. Är *logiskt uttryck* synonym till *villkor*?
38. Ge några exempel på logiska uttryck. Är `temp <= 10` ett logiskt uttryck?
39. Ange koden i C# som kan lagra värdet till ett logiskt uttryck?
40. Vad skilljer ett *sammansatt* från ett *enkelt* logiskt uttryck?
41. Nämn några exempel på logiska operatorer. Är `==` en logisk operator?
42. Skriv ett sammansatt logiskt uttryck som placerar talet `t` inom intervallet [1, 15].
43. Skriv ett sammansatt logiskt uttryck som placerar talet `t` utanför intervallet [1, 15].
44. Bestäm manuellt sanningsvärdena hos de följande logiska uttrycken:
 - a) `(8 < 7) && (true || false)`
 - b) `!(3 < 3.01) || (!(0==0) && true)`
 - c) `(true || !false) && !(4*5==1) && false`
45. Beskriv med egna ord vad en *sanningstabell* är och vad den kan användas till.
46. Döp om i programmet **GuessNEG** (kursboken, sid 69) variabeln `wrongGuess` till `rightGuess`. Tilldela till den det logiska uttrycket `guessedNo == secretNo`, dvs ta bort negationen. Gör de ändringar i programmet som behövs så att programmet fungerar på samma sätt som förut.
47. Formulera som en logisk lag det vardagliga exemplet:

"Jag dricker kaffe utan socker OCH utan mjölk."

betyder samma sak som

"Jag dricker kaffe varken med socker ELLER med mjölk."

Vad kallas lagen i logiken?

48. Hur kan man manuellt bevisa logiska lagar?
49. Har likhetstecknet i *De Morgans lagar* (kursboken, sid 77) samma betydelse som i matematiken? Vad kallas det i logiken?

Kap 3 Datastrukturer och abstrakta datatyper, sid 86-133

50. **Fish** är en klass (kursboken, sid 106). Vad skapas i programmet **ArrayOfRef** (sid 107) med satsen:

```
Fish[] f = new Fish[5];
```

en array av referenser eller en array av objekt? Hur ser man det i koden?

51. Vad skapas i samma program som i fråga 50 med satsen:

```
f[i] = new Fish();
```

ett objekt, en array av objekt, en referens eller en array av referenser?

52. Vilket *defaultvärde* får referensvariabler automatiskt i C#?
53. Beskriv med egna ord vad **null** betyder i C#. Vad kan det användas till?
54. Vilket bibliotek behöver man inkludera i sitt program för att kunna generera slumpstal i C#?
55. Vad gör anropet **r.Next()** där **r** är referens till ett objekt av klassen **Random**?
56. Vad gör anropet **r.Next(a, b)** där **r** refererar till ett **c**-objekt och **a, b** är två heltal?
57. På vilken elementär algoritm bygger bubbsorteringsalgoritmen?
58. Vilken parameteröverföringsmetod tillämpas automatiskt i programmet **BubbleTest** vid anrop av metoderna **RandArray()** och **sort()**?
59. Vad vinner man på att använda *generiska metoder*?
60. Hur definieras en variabel datatyp i generiska metoder? När bestäms den aktuella datatypen?
61. Vilka typer av data kan sorteras med den generiska metoden **G_sort()**?
62. Hur löses sorteringsordningen i den generiska metoden **G_sort()** när data inte är tal?
63. Vilken kod måste man i C# skriva, för att förskjuta ett tecken **n** steg i teckentabellen?
64. Vilken kod i C# ger tecknet till en ASCII-kod?
65. Hur kan man i C# ta ut ett tecken ur en sträng?
66. Vilket tecken tar ut **text[0]** ur strängen **text = 'Gustav'**?
67. I projektet *Lönespecifikation* (sid 93) genomfördes objektorienterad modellering. Beskriv med egna ord de fyra stegen i denna modellering.

Instuderingsfrågor II inför tentan i kursen Algoritmer, ...

Svar till alla frågor kan hittas i [kursboken](#).

Andra källor: [Mängder](#) och [Databaser](#).

Kap 4 Tillämpningar, sid 134-156

1. Vad är motivationen för filhantering i programmering?
2. Vilka två operationer står i centrum för filhantering?
3. Vilket bibliotek behövs för att kunna använda filhanteringsklasser i ett C# program?
4. Vilken klass ur biblioteket i fråga 3 behövs för att skriva från ett C# program till filer?
5. När och var någonstans i datorns fil- och mappsystem skapas filen **Textfil.txt** med satsen:

```
StreamWriter f_Write = new StreamWriter("Textfil.txt");
```

6. Vad händer om du exekverar satsen i fråga 5 om filen **Textfil.txt** redan finns?
7. Vilken del av satsen i fråga 5 skapar filhanteringsobjektet och vad är referensen till objektet?
8. Skriv kod som behövs för att skriva texten "**Hallo!**" till filen som skapas i fråga 5.
9. Skriv kod som stänger filen **Textfil.txt**. Vad händer när filen stängs efter användning?
10. Vilken klass behövs för att läsa från filer till ett C# program?
11. Vad händer i satsen:

```
StreamReader f_Read = new StreamReader("Textfil.txt");
```

12. Vad heter datamedlemmen som representerar filslutstecknet i i klassen **StreamReader**?
13. Av vilken datatyp är filslutstecknet i i klassen **StreamReader**?
14. Skriv kod som läser allt innehåll från filen som nämns i fråga 11.
15. Modifiera satsen i fråga 5 för att lägga till text till filen **Textfil.txt** utan att det gamla innehållet raderas.
16. Modifiera klassen **RandPasswd** ([kursboken](#), sid 148) genom att implementera en ny lösenordpolicy för att generera slumplösenord: 3 gemener, 2 versaler och 2 specialtecken. Inkludera tecknen ? och @ i versalerna, för att de är ”grannar” i ASCII-tabellen.
17. Testa den nya policyn i fråga 16 för att skriva ut de nya slumplösenorden samt tillhörande användarnamn till en fil.
18. Modifiera programmet **EncryptText** ([kursboken](#), sid 152) genom att implementera en ny krypteringsmetod: Definiera krypteringen med funktionen $y = kx + m$, och anropa den med $k = 3$ och $m = -40$. Ordna dekrypteringen med den *inversa* funktionen $y = (x - m) / k$.
19. Testa den nya krypteringsmetoden i fråga 18 för att skriva den krypterade texten till en ny fil.

Kap 5 Datastrukturer i relationsdatabaser, sid 157-225

20. Begreppet *databas* kan betyda olika i olika sammanhang.
Redogör för de två mest relevanta betydelser (sammanhang) som begreppet används inom IT.
21. Vilken databasmodell har i praktiken visat sig vara den mest effektiva strukturen för lagring av information? Med avseende på vilka kriterier har den visat sin överlägsenhet?
22. Beskriv med egna ord de viktigaste egenskaperna hos *relationsdatabaser*. Vad är deras minsta modul?
23. När är en mängd väldefinierad?
24. Vilka operationer kan utföras på mängder? Med vilket grafiskt verktyg kan man visualisera dem?
25. Vad är en *Cartesisk produkt* av två mängder?
26. Definiera begreppet *relation* mellan två mängder.
27. På vilket sätt kan man beteckna en tabell som en *relation*? I så fall relation mellan vilka mängder?
28. Vad består en tabell av?
29. Om en tom tabell kan jämföras med en klass, vad i tabellen är motsvarigheten till klassens datamedlemmar? Vad kan ett objekt av denna klass jämföras med?
30. Vad innebär *modularisering* i relationsdatabaser?
31. Varför leder modularisering till att etablera relationer mellan tabeller?
32. Vad är en *primärnyckel* i en tabell? Varför är den nödvändig?
33. Vad är en *främmande nyckel* i en tabell? När är den nödvändig?
34. Förklara med ett exempel begreppet *relation mellan tabeller*.
35. Vad är en *databashanterare*? Nämn några exempel. Med vilken har vi arbetat under kursen?
36. Har alla databashanterare stöd för SQL?
37. Är SQL ett programmeringsspråk?
38. Skriv en SQL-sats som ska visa kolumnerna **department_name** och **manager_id** i tabellen **departments**.
39. Kan man med en **SELECT**-sats skapa en tabell?
40. Skriv en SQL-sats som ska visa de anställdas efternamn som arbetar på avdelning 80. Företagets anställda antas vara registrerade i tabellen **employees** som har bl.a. kolumnerna **department_id** och **last_name**.
41. Skriv en SQL-sats som ska visa de anställdas för- och efternamn sorterade efter deras anställningsdatum. Företagets anställda antas vara registrerade i tabellen **employees** som har bl.a. kolumnerna **first_name**, **last_name** och **hire_date**.
42. Skriv en SQL-sats som ska skapa en tabell som heter **Kursdeltagare** och har kolumnerna **ElevId**, **Namn**, **Email** och **KursId**. Av dessa ska kolumnen **ElevId** vara primärnyckel och Identity. Ingen kolumn ska tillåtas innehålla tomma celler.