
Instuderingsfrågor inför prov i Programmering 3

Svar till alla frågor kan hittas i kursboken [Programmering 3 med C#](#).

Provet omfattar kursboken: *Programmering 3 med C#, kap 1-3, sid 8-109.*

Kap 1 Fördjupning i programmering, sid 8-47

1. Vad bestod den tekniska innovationen av i den datormodell *John von Neumann* utvecklade 1944?
2. Vilket var det första programmeringsspråk som utvecklades för de första datorerna?
Vilka egenskaper hade det? Vad är dess största skillnad till dagens programmeringsspråk?
3. Vad karaktäriserar de programmeringsspråk som kallades för *lågnivåspråk*? Varför "låg"?
4. Vilket var det första *högnivåspråket*? Varför "hög"?
5. Redogör för skillnaderna mellan begreppen *assemblering*, *kompilering* och *interpretering*.
6. Nämn ett exempel på programmeringsspråk som använde en av metoderna i fråga 5.
7. Vad var det första användningsområdet för programmering?
8. Finns det fortfarande kod som används som är skriven i något av de första programmeringsspråken?
Nämn några sådana samt deras användningsområde.
9. Vilket var det första programmeringsspråk som introducerade *kontrollstrukturer* i programmeringen?
10. Vad menas med *deklarativ programmering*? Är C# ett deklarativt språk?
11. Nämn några underkategorier till deklarativ programmering.
12. Beskriv med egna ord *händelsestyrd programmering*. Nämn exempel.
13. Vad karaktäriserar det som kallas för *spaghettiprogrammering*? Vad är huvudkritiken mot den?
14. Vilket programmeringstekniskt koncept kan ersätta spaghettiprogrammering?
15. Vad är den traditionella, procedurala synen på programmering som rådde på 60- och 70-talet?
16. Vad är den objektorienterade synen på programmering som kom upp på 80-talet?
17. Mellan vilka två programmeringsspråk går historiskt skiljelinjen mellan procedural och objektorienterad programmering? När ungefär inträffade övergången?
18. Vad var anledningen till paradigmskiftet inom programutveckling?
19. Vilka för- och nackdelar har enligt din åsikt den procedurala synen på programmering?
20. Vilka för- och nackdelar har enligt din åsikt den objektorienterade synen på programmering?
21. Definiera med egna ord begreppet *algoritm*. Nämn några exempel.
22. Vad är skillnaden mellan en algoritm och ett datorprogram?
23. Nämn några exempel på *beskrivning* av algoritmer. Vilka av dem är relevanta för programmering?
24. Vad är skillnaden mellan *instruktioner* och *villkor* i algoritmer?
25. Vilka är de tre grundläggande *kontrollstrukturerna* i alla procedurala programmeringsspråk?
26. *Algoritmen för platsbyte* kodas i [kursboken](#) på sid 29. Är programmet **MiniSort** objektorienterat?
27. Varför misslyckas försöket att modularisera **MiniSort**?
28. Beskriv med egna ord parameteröverföringsmetoden *värdeanrop* (*Call by value*).
29. Beskriv med egna ord parameteröverföringsmetoden *referensanrop* (*Call by reference*).
30. Vilka ändringar måste göras i koden, för att lyckas med modulariseringen av **MiniSort**?
31. Hur måste en parameter i en metod deklarerars för att bli en utparameter?
32. Vad är skillnaden mellan en metods returvärde och metodens utparametrar?
33. Varför (och när) behövs utparametrar, när man kan exportera data från en metod via returvärdet?
34. Hur kan man deklarerar en array som parameter i en metod?
35. Var någonstans måste storleken på en array anges när den skickas som parameter till en metod?
36. Vilken parameteröverföringsmetod använder C# automatiskt på en array som parameter?

Kap 2 Logik för programmerare, sid 48-67

37. (Fortsättning följer)

Kap 3 Tillämpningar av OOP, sid 68-109

38. (Fortsättning följer)