
Instuderingsfrågor inför slutprovet i C#

Provet omfattar: Kursboken *Programmering 1+ med C#*, kap 6: sid 111-156.
Övningar: Kap 6, sid 151-156.
Lösningar: sid 230-234.

Svar till alla frågor finns i kursboken [Programmering 1+ med C#](#).

Kap 6 Kontrollstrukturer , sid 111-156

1. Med vilken sats skriver man i C# ett *enkelt* val?
2. Med vilken sats skriver man i C# ett *tvåvägsval*?
3. Med vilken sats kan man skriva i C# ett *flervägsval*?
4. Skriv ett program som läser in tre heltal, hittar och skriver ut det *största* av dem. Modifiera programmet så att det blir det *minsta* av de tre talen. Använd endast enkelt val.

Titta på koden **GuessIfElse** i kursboken sid 128. Försök att besvara frågorna teoretiskt, annars testa koden:

5. Vad händer om man, när programmet **GuessIfElse** körs, matar in tal > 20 eller < 1 ?
6. Vad händer om man, när programmet **GuessIfElse** körs, av misstag matar in en bokstav istället för tal?
7. Val mellan hur många alternativ görs i programmet **GuessIfElse**?
8. Hur många villkor har **if-else**-satserna? Varför står efter **else** inget villkor?
9. Kan man i programmet **GuessIfElse** ersätta **if-else**-satserna med separata, enkla **if**-satser?
I så fall gör det och testa!
10. Vad är nackdelarna med denna version av Gissa tal-spelet?

Titta på koden **GuessDo** i kursboken sid 132. Försök att besvara frågorna teoretiskt, annars testa koden:

11. Vad är den praktiska skillnaden mellan programmen **GuessIfElse** och **GuessDo**?
12. Kan man i koden se hur många varv **do**-loopen kommer att ha?
13. När avslutas **do**-loopen i **GuessDo**? Kan den bli en evighetsloop? Motivera ditt svar.
14. På vilket sätt styr villkoret i **do**-loopen programmet **GuessDo**?
15. Ersätt **switch**-satsen i programmet **GuessDo** med en nästlad **if-else**-sats enligt koden i **GuessIfElse**.
16. Kan man i programmet **GuessDo** ersätta **do**-loopen med en **while**-loop? Om ja, gör det.
17. Efter du har testat *Gissa tal* både med **do**- och **while**-loopen, vilken av dem föredrar du och varför?
18. Vad är nackdelarna i Gissa tal-spelet, version 2?

Titta på koden **GuessDoRand** i kursboken sid 136. Försök att besvara frågorna teoretiskt, annars testa koden:

19. Vilken typ av tal slumpar metoden **Next(1, 21)** och i vilket intervall hamnar slumpalen?
20. Vilken typ av tal slumpar metoden **Next()** och i vilket intervall hamnar slumpalen? Ange det störst och det minst möjliga. Vet du inte, testa! Titta även på programmet **DoRand** i kursboken sid 134.
21. Var någonstans är metoden **Next()** definierad? Hur ser man det i koden?
22. Val mellan hur många alternativ görs i varje varv av **do**-loopen i programmet **GuessDoRand**?
23. På vilket sätt realiserar programmet användarens ev. önskemål om att avbryta spelet och avslöja programmets hemliga tal?

Testa och experimentera gärna även i fortsättningen med programmen i Visual Studio:

24. Skriv en **while**-sats som skriver ut de 50 första positiva heltalen.
25. Skriv en **while**-sats som beräknar och skriver ut summan $1 + 2 + 3 + \dots + 100$.
26. Varför talar man i programmeringssammanhang om *pseudoslumtal*?
27. Vad är en *evighetsloop* och hur kan man förhindra en sådan?
28. Kan man i koden se hur många varv en **for**-loop kommer att ha?
29. Hur många varv har **for**-satserna i programmet **ForRandom** på sid 143-144?
30. Beskriv med egna ord **for**-satsens flödesplan på sid 142.
31. Vad är den väsentliga skillnaden mellan en **for**- och en **while**- resp. **do**-sats?
32. Vad ersätter i C#s Pythons **range(,)** i **for**-satsens huvud?
33. Är **for**-satsens räknare giltig även efter **for**-satsens?
34. Till vilket värde initieras **for**-satsens räknare?
35. Ser man i koden att **for**-satsens räknare uppdateras?
36. Kan **for**-satsen bli en evighetsloop? Motivera ditt svar.
37. Skriv en **for**-sats som skriver ut de 50 första positiva heltalen.
38. Skriv en **for**-sats som beräknar och skriver ut summan $1 + 2 + 3 + \dots + 100$.