Länk till Lektion. Gy. Tillämpad programmering – Avancerad programmering

Provkonstruktion

Årskurs: Gymnasiet
Ämne: Tillämpad programmering
Tema: Avancerad programmering

Syfte

Syftet med provet är att bedöma elevernas kunskaper och förståelse för avancerade programmeringskoncept, inklusive rekursion och iterativa metoder, samt deras förmåga att tillämpa dessa tekniker för att lösa komplexa problem.

Koppling till styrdokument

Centralt innehåll

Denna lektion fokuserar på avancerade programmeringskoncept, inklusive rekursion och iterativa metoder samt grundläggande programmeringsmönster. Eleverna får lära sig att använda dessa tekniker för att lösa mer komplexa problem och förstå hur de kan effektivisera programkod.

Kunskapskrav

Eleven ska kunna använda avancerade programmeringsmetoder och förstå hur dessa kan tillämpas för att lösa problem och optimera kod.

Prov

Faktafrågor

1. Vad är rekursion? A) När en funktion anropar sig själv B) När en funktion anropar en annan funktion C) När en funktion körs flera gånger D) När en funktion skapar en ny instans Rätt svar: **A**

2. Vilket av följande är ett exempel på ett problem som ofta löses med rekursion? A) Sortering av en lista B) Beräkning av Fibonacci-sekvensen C) Sökning i en databas D) Styrning av en loop Rätt svar: **B**

3. Vad kännetecknar en iterativ metod? A) Den anropar sig själv B) Den använder loopar för att upprepa en process C) Den sparar resultat i en databas D) Den körs endast en gång Rätt svar: **B**

4. Vilket av följande är ett designmönster? A) Rekursion B) Loop C) Observer D) Iteration Rätt svar: **C**

5. Vilken nackdel kan uppstå vid användning av rekursion? A) Den är alltid ineffektiv B) Den kan leda till stack overflow C) Den kan inte användas för komplexa problem D) Den är alltid långsam Rätt svar: **B**

6. Vilken typ av problem är ofta bättre att lösa med iterativa metoder? A) Problem med många återkommande anrop B) Problem som kräver minneshantering C) Enkla loopar och upprepningar D) Problem som har flera lösningar Rätt svar: **C**

7. Vad är syftet med designmönster? A) Att optimera databaser B) Att spara tid vid programmering C) Att skapa kod som är lättare att återanvända D) Att skräddarsy programvara Rätt svar: **C**

8. Vilket av följande beskriver en basfall i rekursion? A) Ett villkor för att stoppa rekursionen B) När en funktion anropar sig själv för första gången C) En iterativ version av en funktion D) En funktion som inte har några villkor Rätt svar: **A**

9. Vad händer om basfallet inte hanteras i en rekursiv funktion? A) Funktionen körs alltid framgångsrikt B) En felkod genereras utan att stoppa C) Funktionen slutar köra efter ett tag D) Det kan leda till att programmet kraschar Rätt svar: **D**

10. Vad är en fördel med rekursion? A) Det är alltid snabbare B) Den gör koden mer läsbar för komplexa problem C) Den kräver mindre minne D) Det är enklare att implementera Rätt svar: **B**

11. Vad är skillnaden mellan rekursion och iteration? A) Rekursion är alltid mer effektiv B) Iteration kan leda till stack overflow C) Rekursion använder anrop medan iteration använder loopar D) Det finns ingen skillnad Rätt svar: **C**

12. Varför kan iterativ kod vara att föredra i vissa fall? A) Den alltid är enklare att förstå B) Den kan kräva mer minne C) Den kan vara snabbare och mer effektiv D) Den är alltid mer flexibel Rätt svar: **C**

13. Vilken typ av problem kan vara mer lämpad för rekursiva metoder? A) Problem som har en enkel secvens B) Problemlösning som diversifierar anrop C) Problem som innebär

multipla loopar D) Hierarkiska problem som trädstrukturer Rätt svar: **D**

14. Vad är en potential problem med att använda designmönster? A) De är alltid svåra att implementera B) De kan leda till överdesign av systemet C) De har inga praktiska tillämpningar D) De är inte applicerbara Rätt svar: **B**

15. Vad är syftet med att analysera olika programmeringsmetoder? A) För att kunna ignorera dem B) För att förstå deras för- och nackdelar C) För att diskutera med andra D) För att memorera dem Rätt svar: **B**

Resonerande frågor

1. Diskutera för- och nackdelarna med att använda rekursion jämfört med iteration. Syftet är att ge eleverna möjlighet att kritiskt bedöma olika metoder och deras effektivitet.

2. Ge exempel på ett programmeringsproblem där rekursion är mer lämplig än iteration och förklara varför. Denna fråga uppmuntrar en djupare förståelse för tillämpningarna av rekursion.

3. Hur kan designmönster förbättra programstruktur och underhåll? Denna fråga syftar till att få eleverna att resonera kring konceptuell förståelse av programdesign.

4. Kanaliserar din respons på diskussioner kring hur rekursion kan förbättra kodens läsbarhet. Här ges eleverna möjlighet att argumentera för kodens estetik och praktiska tillämpning.

5. Vilken inverkan har rekursion på prestanda och minnesanvändning? En insikt i dessa frågor kan hjälpa eleverna att förstå de tekniska begränsningarna med rekursion.

6. Ge en kort beskrivning av ett projekt där du skulle använda båda metoder. Förklara valet av metod. Detta ger eleverna chansen att tänka applicerande kring tidigare lärdomar.

7. Reflektera över hur du skulle förklara dessa koncept för en person utan programmeringsbakgrund. Genom denna fråga uppmanas elever att förenkla komplexa idéer och presentera dem pedagogiskt.

8. Hur visar designmönster den evolutionära aspekten av programmering? Med denna fråga uppmuntras eleverna att tänka långsiktigt när det gäller programutveckling och design.

Bedömning

Provet bedöms med poängsystem. För att uppnå betyget E krävs minst 8 poäng totalt. För betyg C krävs 12 poäng, därav minst 3 poäng från resonerande frågor. För betyg A krävs 18 poäng, med minst 5 poäng från resonerande frågor.