



# EN BANKAPPLIKATION

## Sammanfattning

En liten uppgift där det skall skapas en applikation som simulerar händelser för ett bankkonto. I uppgiften så tränar du på hantering av variabler och arrayer, in- och utmaningar, selektioner och iterationer, metoder samt pseudokod och aktivitetsdiagram

Johan Hällgren

## KUNSKAPSKRAV

I denna uppgift så bedöms du efter följande kunskapskrav. Alla kunskapskrav behöver inte vara möjliga att nås på alla nivåer.

| Kunskapskrav          | Betyget E   | Betyget C   | Betyget A  |
|-----------------------|---|---|--|
| <b>1. Planering</b>   | Eleven formulerar och planerar i <b>samråd</b> med handledare programmeringsuppgifter med pseudokod eller diagramteknik.<br><br>I planeringen väljer eleven <b>med viss säkerhet</b> kontrollstrukturer, metoder, variabler, datastrukturer och algoritmer som är adekvata för uppgiften. | Eleven formulerar och planerar <b>efter samråd</b> med handledare programmeringsuppgifter med pseudokod eller diagramteknik.<br><br>I planeringen väljer eleven <b>med viss säkerhet</b> kontrollstrukturer, metoder, variabler, datastrukturer och algoritmer som är adekvata för uppgiften. | Eleven formulerar och planerar <b>efter samråd</b> med handledare programmeringsuppgifter med pseudokod eller diagramteknik.<br><br>I planeringen väljer eleven <b>med säkerhet</b> kontrollstrukturer, metoder, variabler, datastrukturer och algoritmer som är adekvata för uppgiften <b>samt motiverar utförligt sina val</b> . |
| <b>2. Kodkvalitet</b> | I sin programmering skapar eleven med konsekvent kodningsstil och tydlig namngivning korrekt, strukturerad och <b>enkelt</b> kommenterad källkod med <b>tillfredsställande</b> resultat.  | I sin programmering skapar eleven med konsekvent kodningsstil och tydlig namngivning korrekt, strukturerad och <b>noggrant</b> kommenterad källkod med <b>tillfredsställande</b> resultat.  | I sin programmering skapar eleven med konsekvent kodningsstil och tydlig namngivning korrekt, strukturerad och <b>noggrant och utförligt</b> kommenterad källkod med <b>gott</b> resultat.   |
| <b>3. Interaktion</b> | Dessutom väljer eleven <b>med viss säkerhet</b> ett uttryckssätt som är anpassat för att på ett <b>tillfredsställande</b> sätt interagera med den avsedda användaren.   | Dessutom väljer eleven <b>med viss säkerhet</b> ett uttryckssätt som är anpassat för att på ett <b>tillfredsställande</b> sätt interagera med den avsedda användaren.   | Dessutom väljer eleven <b>med säkerhet</b> ett uttryckssätt som är anpassat för att på ett <b>gott</b> sätt interagera med den avsedda användaren.   |
| <b>4. Produkten</b>   | Elevens färdiga program eller skript är utförda med <b>tillfredsställande</b> resultat i ett eller flera programspråk som är stabila och robusta <b>i program av enkel karaktär</b> .   | Elevens färdiga program eller skript är utförda med <b>tillfredsställande</b> resultat i ett eller flera programspråk som är stabila och robusta.   | Elevens färdiga program eller skript är utförda med <b>gott</b> resultat i ett eller flera programspråk som är stabila och robusta <b>i program av komplex karaktär</b> .  |
| <b>5. Felsökning</b>  | Eleven anpassar <b>med viss säkerhet</b> sin planering av programmeringsuppgiften och utför felsökning av <b>enkla</b> syntaxfel.   | Eleven anpassar <b>med viss säkerhet</b> sin planering av programmeringsuppgiften och utför <b>på ett systematiskt sätt</b> felsökning av syntaxfel, <b>körtidsfel och programmeringslogiska fel</b> .  | Eleven anpassar <b>med säkerhet</b> sin planering av programmeringsuppgiften och utför <b>på ett systematiskt och effektivt sätt</b> felsökning av syntaxfel, <b>körtidsfel och programmeringslogiska fel</b> .  |
| <b>6. Utvärdering</b> | Innan programmeringsuppgiften avslutas utvärderar eleven med <b>enkla</b> omdömen programmets prestanda och ändamålsenlighet i <b>någon</b> situation och i <b>något</b> sammanhang.  | Innan programmeringsuppgiften avslutas utvärderar eleven med <b>nyanserade</b> omdömen programmets prestanda och ändamålsenlighet i <b>några</b> situationer och sammanhang.  | Innan programmeringsuppgiften avslutas utvärderar eleven med <b>nyanserade</b> omdömen <b>och med förslag på förbättringar</b> programmets prestanda och ändamålsenlighet i <b>flera</b> situationer och sammanhang.   |
| <b>7. Terminologi</b> | Eleven kommunicerar om programmeringsuppgiften och dess utvärdering och använder då <b>med viss säkerhet</b> datavetenskapliga begrepp.   | Eleven kommunicerar om programmeringsuppgiften och dess utvärdering och använder då <b>med viss säkerhet</b> datavetenskapliga begrepp.   | Eleven kommunicerar om programmeringsuppgiften och dess utvärdering och använder då <b>med säkerhet</b> datavetenskapliga begrepp.   |
| <b>9. Principer</b>   | Eleven redogör <b>översiktligt</b> för principer för att uppnå god kvalitet vid skapandet av datorprogram.  | Eleven redogör <b>utförligt</b> för principer för att uppnå god kvalitet vid skapandet av datorprogram.   | Eleven redogör <b>utförligt och nyanserat</b> för principer för att uppnå god kvalitet vid skapandet av datorprogram.  |

## UPPGIFTEN

Denna uppgift ges i två delar, en grunduppgift och en vidareutbyggnad av applikationen. Grunduppgiften består av en presentation, en del med lite mer hjälp och slutligen en del där du som elev får ganska mycket hjälp genom att vi gör delar av uppgiften i form av gemensam programmeringsträning, om intresse finns för detta.

## PRESENTATION

Du skall bygga en bankapplikation som simulerar ett bankkonto. I applikationen skall det gå att göra flera uttag och flera insättningar innan användaren väljer att avsluta programmet. När applikationen startar skall det finnas 1000kr på kontot, när applikationen avslutas så försvinner alla transaktioner och applikationen startar om från ursprungsläget nästa gång.

Alla transaktioner, heltal, skall lagras i en array. Aktuellt saldo skall beräknas i en metod utifrån de lagrade transaktionerna. Ditt saldo får aldrig vara negativt, det går alltså inte att göra ett större uttag än vad du har som saldo på ditt konto. Hur du löser detta skall du visa med pseudokod och aktivitetsdiagram.

Applikationen skall kunna hantera insättningar, uttag, visa saldo och transaktioner och skall gå att avsluta på ett kontrollerat sätt.

## LITE MERA HJÄLP

Här kommer ett exempel på lösning, använd vissa eller alla tips för att bygga din egen lösning.

- Börja med att bygga en programloop som gör att du hela tiden kommer tillbaka till menyn när du har slutfört en insättning, uttag eller annan uppgift. Följande menyalternativ är lämpliga;
  - 1. Insättning
  - 2. Uttag
  - 3. Saldo
  - 4. Transaktioner
  - 0. Avsluta
- Välj lämpligt inmatningssätt för menyn, förslagsvis char från en Scanner.
- Skapa en array av typen int med alla transaktioner. Det räcker att skapa 20 komponenter i arrayen.
- Skapa en metod som beräknar saldot utifrån transaktionerna i arrayen.
- Skapa funktionalitet för insättning och uttag, för uttag behöver du kontrollera att uttaget är mindre än saldot för att kunna godkännas. Lagra sedan ett uttag som ett negativt tal. Kontrollen för uttag skall planeras med aktivitetsdiagram och pseudokod, se läroboken för hjälp.
- Menyvalet saldo skriver ut saldot och menyvalet transaktioner listar alla transaktioner som är gjorda.

## MYCKET HJÄLP

Denna uppgift kan vi, om intresse finns, genomföra tillsammans som en gemensam programmeringsträning.

## VIDAREUTBYGGNAD

Den grundläggande uppgiften går att bygga ut på många olika sätt, här följer ett antal möjligheter av varierande svårighetsgrad;

- Gör menyutskriften som en metod som anropas när den behövs.
- Fundera på hur menyn skall fungera vid inmatning av andra menyalternativ än de som angivits. Felhantering kommer vi jobba mer med i nästa uppgift.
- En array är ju statisk i antalet komponenter vilket innebär att när den är full så kan inte fler transaktioner göras, ett dynamiskt alternativ är olika former av samlingar där ArrayList är den samling som är mest lik en array. Läs igenom kapitel 14.1 i boken och se om du kan bygga om transaktionsarrayen till en sådan typ.
- Händelser som upprepas är lämpliga att skapa metoder för, inmatning och validering av belopp är en sådan sak som borde genomföras i en metod.