

Ordinarie tentamen för ID1004 Objektorienterad programmering, 11 mars 2014, 14-19

Denna tentamen examinerar 3.5 högskolepoäng av kursen.

Inga hjälpmedel är tillåtna.

Tentamen består av en obligatorisk del och en extra del. För ett godkänt betyg på hela tentamen måste den obligatoriska delen vara godkänd. Detta berättigar till betyg E på tentamen. För högre betyg kan ytterligare uppgifter lösas i den extra delen. Examinator förbehåller sig möjligheten att justera tentamensbetyget med hänsyn till en bedömning av helhetsintrycket.

Den obligatoriska delen ger maximalt 41 poäng. För godkänt krävs 27 poäng eller mer.

Läs noga igenom alla frågorna först och planera tiden.

Om inget annat uttryckligen sägs så visas och efterfrågas kod och uttryck ur Java version 6 eller 7. Var noga med metodens returtyper.

Obligatorisk del

Fråga 1 (6p)

Deklarera och initiera de nedan beskrivna variablerna. Skriv en skiss till en klass som heter `Varor` med en default-konstruktör och placera in variablerna på sina korrekta platser.

- a) En statisk heltalsvariabel som är åtkomlig från alla klasser. Variabeln ska heta `antal` och dess första värde ska vara noll.
- b) En klassvariabel för decimaltal som kan ärvas men inte är åtkomlig från andra klasser. Variabeln ska heta `pris` och initieras till ett.
- c) En strängvariabel som är lokal i konstruktorn. Variabeln ska heta `visning` och initieras till null.

Fråga 2 (3p)

Skriv en metod som tar emot en referens till en array av heltal. Metoden ska returnera det sista elementet i arrayen. Om det inte finns något sista tal, ska metoden returnera -1. Metoden ska heta `sistaTalet` och vara möjlig att anropa från alla klasser utan att behöva skapa ett objekt.

Ett arrayobjekt har en konstant `length` som anger hur många element arrayen rymmer. En arrays index börjar alltid på noll. Det är inte nödvändigt att skriva någon omgivande klassdeklaration.

Fråga 3 (2p + 2p)

Ett svenskt personnummer innehåller årtal (2), månad(2), dag(2), ett skiljetecken (-/+), ett tresiffrigt löpnummer(3) och en kontrollsiffra(1). Skiljetecknet är '-' såvida inte personen är äldre än 100 år varvid '+' används istället, till exempel 740908-1123.

- Föreslå en lämplig datastruktur för att lagra ett personnummer i Java. Datastrukturen ska vara minimal och representera personnummerformatet ovan och ingenting därutöver.
- Givet den föreslagna datastrukturen, skriv en metod
`public boolean hundraÅring(/* datatyp */ personnummer)`
som returnerar true om personen är äldre än 100 år, och false annars.

Fråga 4 (1p+1p)

Givet följande kodfragment:

```
int a = 10;           // 1
int b = a + 1;        // 2
int c = 3;            // 3
a = a * c;            // 4
c = c - 1;            // 5
c += a;              // 6
a = b - 1;           // 7
System.out.println(a); // 8
```

- Vilket tal skriver programmet ut?
- Vilka rader kan tas bort utan att utskriften ändras?

Fråga 5 (3p)

Skriv en metod som heter `udda`. Metoden ska ta emot ett heltal och returnera true om talet är udda och false om talet är jämnt. För heltalet n gäller: $n \bmod 2 = 1$ om talet är udda.

Fråga 6 (4p)

Skriv en metod som heter `antaLUdda`. Metoden ska ta emot en array av heltal och returnera antalet udda tal i arrayen. Använd ett anrop till metoden `udda` från fråga 5.

Fråga 7 (2p)

Fyll i sanningstabellens tredje kolumn (svara på separat papper).

a	b	$((a \ \&\& \ !b) \ \ (!a \ \&\& \ b))$
false	false	
true	false	
false	true	
true	true	

Fråga 8 (2p+2p+ 3p)

- a) Skriv metoden `double medel(double[] v)` som returnerar medelvärdet av talen i arrayen `v`.
- b) Skriv metoden `int antalMindre(double m, double [] v)` som returnerar antal tal i array `v` som är mindre än värdet `m`.
- c) Skriv med hjälp av de två metoderna ovan, metoden `double [] underMedel(double [] v)` som returnerar en ny array som innehåller de tal i `v` som är mindre än medelvärdet.

Fråga 9 (1p+2p+1p+3p+1p+2p)

Med utgångspunkt från klasserna och programmet Q9 i bilaga 1:

- a) Rita en graf över klasshierarkin.
- b) Studera `main`-metoden i klass Q9: varför går det att tilldela arrayen `shapes` objekt ur andra klasser än `Shape`?
- c) Vad händer om man lägger till kod som gör `new Shape()`?
- d) Programmet genererar utskriften nedan: förklara hur det går till.

```
{java.awt.Point[x=0,y=0], java.awt.Point[x=0,y=10]}  
{java.awt.Point[x=0,y=0], java.awt.Point[x=0,y=10], java.awt.Point[x=10,y=10]}  
C:java.awt.Point[x=10,y=10] r=5
```
- e) Klass `Circle` en egen `toString`-metod. Vad kan det finnas för skäl till det?
- f) Utöka klasshierarkin med klassen `Rectangle`. Konstruktorn ska acceptera dessa fyra parametrar: `int x`, `int y`, `int width`, `int height`, där `x,y` är koordinaten för det övre vänstra hörnet, `width` är rektangelns bredd och `height` är rektangelns höjd.