

Tentamen för ID1004 Objektorienterad programmering (vilande kurs), 29 oktober 2013, 9-13

Denna tentamen examinerar 3.5 högskolepoäng av kursen.

Inga hjälpmedel är tillåtna.

Tentamen består av tre sektioner. För ett godkänt betyg på hela tentamen måste varje sektion vara godkänt besvarad. Tentamensbetyget A-F bestäms därefter i huvudsak av den ackumulerade poängsumman (korrekt besvarade frågor) enligt tabellen nedan. Examinator förbehåller sig möjligheten att vid gränsfall justera tentamensbetyget med hänsyn till en bedömning av helhetsintrycket, inklusive att om så anses påkallat ge betyget Fx. Betyget Fx innebär en möjlighet att enligt senare överenskommelse komplettera tentamen till högst betyg E.

Poäng	0-15	16-19	20-23	24-28	29-32	33-36
Tentamensbetyg	F/Fx	E	D	C	B	A

Läs igenom alla frågorna först och planera tiden. Försök att besvara alla frågor så väl som möjligt.

Om inget annat uttryckligen sägs så visas och efterfrågas kod kompatibel med Java version 6 och 7.

Grundläggande syntaktiska konstruktioner och begrepp (16p)

1. (3p) Vilka av följande namn är syntaktiskt giltiga identifierare i Java? Om något namn är ogiltigt, varför är det så?

- a) szf01qxxnt [giltig]
- b) 5n_3b1k [ogiltig, börjar med siffra]
- c) Qp k\$u2 [ogiltig, innehåller otillåtet tecken, space]

2. (2p) Vilka texter producerar kodraderna nedan?

- a) `System.out.println ("Talet är " + (3 + 8));` [Talet är 11]
- b) `System.out.println ("Talet är " + 3 + 8);` [Talet är 38]

3. (5p) Givet följande variabeldeklarationer, besvara varje fråga:

```
int cycles, lastLengths = 0;
final int MY_YEAR = 0;
int tonnes, rate, year = 0;
```

- a) Hur många variabler är deklarerade? [6 st]
- b) Vad har variablerna för typ? [int]
- c) Vilka av variablerna initieras explicit? [lastLengths, year, MY_YEAR]
- d) Är följande uttryck tillåtet? Motivera svaret:
`year = 20;` [Ja, det är en vanlig variabel]
- e) Är följande uttryck tillåtet? Motivera svaret:
`MY_YEAR = 40;` [Nej, MY_YEAR är deklarerad som final och redan tilldelad.]

4. (1p) Ordna följande datatyper i stigande storleksordning (med storlek avses antal bitar i representationen):

float byte long short int [byte, short, int&float, long]

5. (5p) Visa och förklara resultaten av dessa uttryck:

- | | |
|-----------------------|--|
| a) $13 \% 4$ | [13 modulo 4 = 1, rest vid heltalsdivision] |
| b) $10 / 4$ | [10 dividerat med 4 = 2, heltalsdivision] |
| c) $12 + 6 * 4$ | [12 + 24 = 36] |
| d) $(12 + 6) * 4$ | [18 * 4 = 72] |
| e) $15 \% 5 + 5 \% 5$ | [0 + 0 = 0] |

Grundläggande datalogiska begrepp och relationer (11p)

6. (2p) Givet följande sekvens av programinstruktioner:

```
1. double x, y, k, b;  
2. x = 3.14;  
3. y = -7.9125;  
4. k = -2.101;  
5. b = 13.17;  
6. y = k * x + b;
```

- a) En av raderna är överflödig och kan plockas bort. Vilken, och varför det? [rad 3, y läses inte]
b) Vilka rader går att kasta om ordningen på utan att beräkningen ändras? [2,3,4,5]

7. (1p) Vilken text kommer att skrivas ut av denna kod?

```
int small = 100;  
int medium = 75;  
int big = 42;  
if ((small < big) || (medium > small)) {  
    System.out.println("PEW");  
}  
else if (big > medium) {  
    System.out.println("DIE");  
}  
else {  
    System.out.println("PIE");  
}
```

[PIE]

8. (2p) Komplettera nedanstående metod så att den implementerar den logiska funktionen exclusive-OR, (ömsesidigt uteslutande), dvs att en och endast en av parametrarna är sann:

```
public boolean xor(boolean p, boolean q) {  
  
    [return (p && !q) || (!p && q);]  
}
```

9. (2p) Vilken talserie skrivs ut av följande kod?

```
int x = 22;
int y = 13;
while (y < x) {
    x = (x + y) / 2;
    System.out.println(x);
}
```

[17, 15, 14, 13]

10. (4p) Arrayen `final int [] nums` innehåller tal från en okänd källa.

a) Skriv ett kodfragment som räknar antal tal i `nums` som är mindre än noll.

```
int nofNegative = 0;
for (int n : nums) {
    if (n < 0) nofNegative++;
}
```

b) Skriv ett kodfragment som sätter variablerna `int minIndex, maxIndex`; till index för minsta respektive största värdet i `nums`.

```
int minIndex = 0;
int maxIndex = 0;
int minValue = nums[0];
int maxValue = nums[0];
for (int i = 1; i < nums.length; i++) {
    if (nums[i] < minValue) {
        minValue = nums[i];
        minIndex = i;
    }
    if (maxValue < nums[i]) {
        maxValue = nums[i];
        maxIndex = i;
    }
}
```

Grundläggande objektorienteringsbegrepp och komplexitet (10p)

11. (3p) Försvara användningen av begreppet inkapsling i objektorienterad programmering.

Inkapsling skyddar klassen och dess objekt från insyn och manipulation på andra sätt än genom dess gränssnitt. Genom att använda deklarationer som `public`, `default`, `protected` och `private` kan programmeraren förhindra att kod skriven för andra klasser ser eller ärver attribut (fält, metoder och inre klasser) som inte skall vara synliga utåt. Detta minskar risken för programmeringsfel och beroenden mellan klasser som försvårar underhåll av programkoden.

12. (2p) I objektorienterad programmering, vad är det för skillnader mellan:

```
public class MyClass { ... }
```

och

```
MyClass mc = new MyClass();
```

Det första är en deklaration av en klass, en specifikation av dess attribut och tjänster i form av fält, metoder och inre klasser. För varje klass finns endast en enda deklaration och med undantag för statiska fält och metoder så finns det ingen körbar kod förrän en instans av klassen skapats. Det andra är en variabeldeklaration (mc) och en instansiering (new). En instans av MyClass skapas och en referens till objektet tilldelas variabeln mc. Det går nu att följa referensen och använda klassens instans genom att anropa metoder. Det går även att vid behov skapa fler instanser.

13. (3p) I objektorienterad programmering, vad menas med ett gränssnitt?

Ett gränssnitt kan antingen vara en interface-klass i Java, eller den mängd med publika attribut (fält, metoder och inre klasser) som en klass exponerar mot andra klasser. Gränssnittet tillåter en väldefinierad användning av klassens instanser, och gör det samtidigt möjligt att underhålla implementationen bakom gränssnittet utan att anropande kod behöver modifieras. En interface-klass i Java är speciell i den bemärkelsen att den definierar vilka metoder som skall vara synliga i objektet och vilka signaturer och returvärden dessa ska ha. Interface-klassen säger dock inte hur metoderna ska implementeras, det är upp till den implementerande klassen.

(fråga 14 på nästa sida)

14. (2p) Vid digital ljudbehandling behöver man ofta addera (mixa) samman signaler med varandra. I kodexemplen nedan ska ett antal arrayer samlade i `input` mixas ner i en enda array, `mixBuffer`. Eftersom `mixBuffer` återanvänds måste eventuella data i den skrivas över, och det gör man enklast genom att kopiera första input och sedan addera till de andra. Nedan visas två alternativa sätt att programmera nedmixningen. Är det någon skillnad mellan dem effektivitetsmässigt? I så fall varför?

Koden antar att alla arrayer som innehåller signaler är av samma längd (1024 element), är initierade och redan har använts flera gånger.

```
float [] mixBuffer;
float [][] input;

// Alt a
for (int i = 0; i < input.length; i++) { // För varje input
    for (int j = 0; j < mixBuffer.length; j++) { // Lägg till i mix
        if (i == 0) {
            mixBuffer[j] = input[i][j]; // Kopiera om det är första input
        } else {
            mixBuffer[j] += input[i][j]; // annars addera input
        }
    }
}

// Alt b
for (int j = 0; j < mixBuffer.length; j++) { // Kopiera första input
    mixBuffer[j] = input[0][j];
}
for (int i = 1; i < input.length; i++) { // Addera resterande inputs
    for (int j = 0; j < mixBuffer.length; j++) {
        mixBuffer[j] += input[i][j];
    }
}
```

Om vi antar att det finns $n = \text{input.length}$ arrayer med insignal, så kräver uppgiften 1024 kopieringar och $(n-1) \cdot 1024$ additioner. Båda alternativen utför detta, så i det avseendet är de likvärdiga.

Alternativ a har ett test i sin inre loop om i är lika med noll. Detta test kommer att utföras för varje kopiering och addition och vara sant 1024 gånger och falskt $(n-1) \cdot 1024$ gånger. Alternativ b utför inget sådant test utan har bara de kopieringar och additioner som faktiskt behövs. Därför är alternativ b mer effektivt.