

Ordinarie tentamen för ID1004 Objektorienterad programmering, 7 april 2015

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 34 poäng. För godkänt krävs 22 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 senare. Var noga med metodens returtyper.

## Obligatorisk del

### Fråga 1 (6p)

Deklarera och initiera de nedan beskrivna variablerna:

1. En statisk variabel med dubbel flyttalsprecision som heter `alfa` och som är initierad till halva värdet av `Math.PI`.  
`public static double alfa = Math.PI / 2d;`
2. En klassvariabel som heter `brus` och som kan ta heltalsvärden i området `-100 – +100`. Variabeln ska endast vara åtkomlig i den klass där den är definierad, och inte använda mer än två byte minne.  
`private short brus = 0;`  
`private byte brus = 0;`
3. En referensvariabel till instanser av klassen `StringBuilder`. Variabeln ska gå att ärva, men inte vara synlig utanför klassens package. Variabeln ska ha namnet `sb` och initieras till att inte referera till någonting.  
`protected StringBuilder sb = null;`

### Fråga 2 (5p)

Givet denna variabeldeklaration:

```
protected float [] r = new float[200];
```

1. Vilken datatyp har variabeln `r`?  
`Referens till array of float.`
2. Vilken elementtyp har `r`?  
`float`

3. Hur många element har r?  
200
4. Vilket index har första elementet?  
0
5. Vilket index har sista elementet?  
199

### Fråga 3 (2p)

Deklarera med en passende datatyp en variabel kallad `antal` och tilldela den en array med plats för 101 heltal.

```
int [] antal = new int [101];
```

### Fråga 4 (1p)

Vilket sanningsvärde beräknas detta boolska uttryck till?

```
(true && false) || !false  
true
```

### Fråga 5 (2p)

Givet denna kod:

```
int a = /* något värde */ ;  
int b = /* något värde */ ;
```

Konstruera ett villkorsuttryck som är sant om och endast om variabeln `a` är större än 7 samtidigt som variabeln `b` är i området `-5 – +5` (inklusive).

```
(a > 7) && (-5 <= b) && (b <= 5)
```

### Fråga 6 (2p)

Skriv en if-sats som tilldelar variabeln `ep` värdet 0 om `ep` är större än eller lika med variabeln `epEnd`.

```
if (epEnd <= ep)  
    ep = 0;
```

### Fråga 7 (1p)

Hur många varv gör denna loop?

```
int i = 5;  
while (i < 10) {  
    i++;  
}
```

Den gör 5 varv.

### Fråga 8 (2p)

Skriv en for-loop som summerar alla tal i arrayen `int [] ar`. Lägg summan i variabeln `int sum`.

```
int sum = ar[0];
```

```
for (int i = 1; i < ar.length; i++)  
    sum += ar[i];
```

### Fråga 9 (1p)

Vilken returtyp har följande metoddeklaration?

```
public static float floor (float d);  
float
```

### Fråga 10 (2p)

Arean för en cirkel är  $A=\pi r^2$ . Skriv en metod `area` som tar emot radien i dubbel flyttalsprecision och returnerar arean för motsvarande cirkel.

```
public double area(double r) {  
    return Math.PI * r * r;  
}
```

**Fråga 11(10p)**

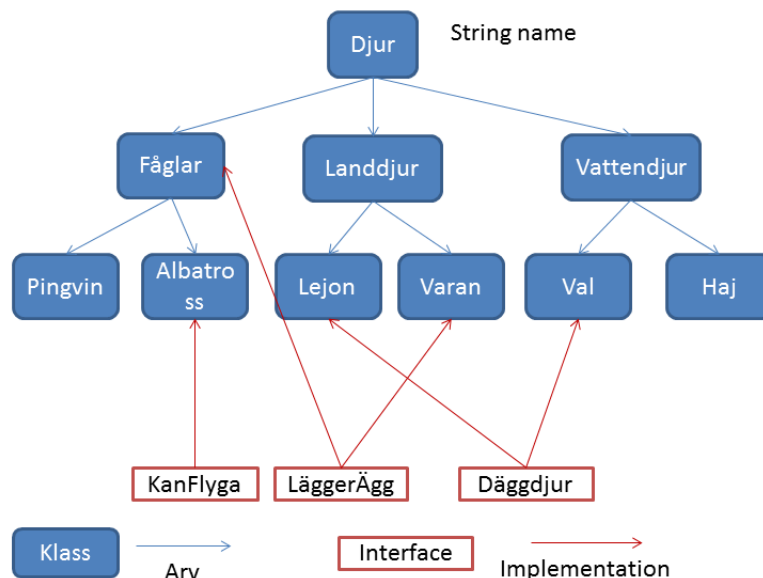
Använd endast klasser, interface, arv och polymorfism (inga extra variabler utöver namnet) för att modellera följande med hjälp av Java:

- Det finns djur, och alla djur har ett namn (t ex lejon, val, pingvin).
- Djur kan vara fåglar, landdjur eller vattendjur.
- Pingviner och albatrossar är fåglar.
- Lejon och varaner är landdjur.
- Valar och hajar är vattendjur.
- Albatrossar kan flyga men det kan inte pingviner.
- Lejon och valar är däggdjur.
- Fåglar och varaner lägger ägg.

Att göra:

- Rita en graf över modellen. Visa tydligt vad som är klass, interface och arv. Det är tillåtet, men inget krav, att även skriva Java-kod som implementerar modellen.
- Antag att variabeln `LinkedList<Djur> allaDjur` existerar och innehåller djur ur modellen. Skriv en loop som går igenom `allaDjur` och skriver ut namnen på de djur som är både landdjur och lägger ägg.

```
for (Djur d : allaDjur)
    if (d instanceof Landdjur)
        if (d instanceof LägerÄgg)
            System.out.println(d.name);
```





## Extra del

Med godkänd obligatorisk del, ger poäng på extradelen: 7-10 D, 11-14 C, 15-18 B och 19-22 A.

### Fråga 12 (8p)

Kasinospelet Roulette spelas med hjälp av ett snurrande hjul, en kula och en spelplan med olika fält. Ett Europeiskt roulettehjul har 37 olika fack som kulan kan landa i, numrerade 0, 1, 2, ... 36. Numren 1-36 är antingen svarta eller röda. Nollan är grön, och om den kommer upp så vinner kasinot. Det finns flera olika sätt att spela på; man kan satsa på ett enskilt nummer, rött eller svart, udda eller jämt, högt eller lågt, eller på flera nummer samtidigt.

Vi antar att ett online-kasino för sin webbsajt behöver ett antal statiska Java-metoder som hjälper till att avgöra om ett antal vinstalternativ är sanna eller inte. Parametern  $k$  är i samtliga fall det nummer som kulan har valt.

```
// manque, den lägre halvan av nummerområdet, 1–18 (inklusive)

public static boolean manque (int k)
    public static boolean manque (int k) {
        return (1 <= k) && (k <= 18);
    }

// passe, den övre halvan av nummerområdet, 19–36 (inklusive)

public static boolean passe (int k)
    public static boolean passe (int k) {
        return (19 <= k) && (k <= 36);
    }

// pair, ett jämnt nummer

public static boolean pair (int k)
    private static boolean odd (int k) {
        return k % 2 == 1;
    }

    public static boolean pair (int k) {
        return (0 < k) && !odd(k);
    }

// impair, ett udda nummer

public static boolean impair (int k)
    public static boolean impair (int k) {
        return (0 < k) && odd(k);
    }

// rouge, ett rött nummer
```

```
public static boolean rouge (int k)
    private static boolean range1 (int k) {
        return ((1 <= k) && (k <= 10)) || ((19 <= k) && (k <= 28));
    }

    private static boolean range2 (int k) {
        return ((11 <= k) && (k <= 18)) || ((29 <= k) && (k <= 36));
    }

    public static boolean rouge (int k) {
        return (range1(k) && odd(k)) || (range2(k) && !odd(k));
    }

// noir, ett svart nummer

public static boolean noir (int k)
    public static boolean noir (int k) {
        return (range1(k) && !odd(k)) || (range2(k) && odd(k));
    }
```

Rött eller svart avgörs av dessa regler:

- I områdena 1—10, 19—28 är udda nummer röda och jämna nummer svarta.
- I områdena 11—18, 29—36 är udda nummer svarta och jämna nummer röda.

Kom ihåg att om kulan landar på 0 så vinner kasinot och inget av vinstalternativen ovan är sant.

Skriv implementationer för samtliga 6 metoder. Använd privata hjälpmetoder där så är påkallat.

**Fråga 13 (6p)**

Givet följande interface:

```
public interface Filter<T> {  
    public boolean accept (T e);  
}
```

och detta program:

```
public class Solution {  
  
    public static String [] sieve (Filter<String> fs,  
                                   String [] candidates)  
    {  
        /* KOD FATTAS HÄR */  
        int count = 0;  
        for (String s : candidates)  
            if (fs.accept(s))  
                count++;  
        String [] rtn = new String [count];  
        count = 0;  
        for (String s : candidates)  
            if (fs.accept(s))  
                rtn[count++] = s;  
        return rtn;  
    }  
  
    public static void main(String[] args) {  
  
        String [] candidates =  
            {"adam", "sonny", "beata", "sixten", "hugo"};  
  
        Filter<String> fs = new Filter<String>() {  
            public boolean accept (String s) {  
                return s.startsWith("s");  
            }  
        };  
        String [] selected = sieve(fs, candidates);  
  
        System.out.println(java.util.Arrays.toString(selected));  
    }  
}
```

- a) Komplettera kroppen i metoden `sieve` så att metoden returnerar en array med exakt de strängar som väljs ut av filtret `fs`.
- b) Vilka namn kommer programmet att skriva ut?  
"sonny" och "sixten"