

TENTAMEN I PROGRAMMERING

Ansvarig:	Jan Skansholm
Betygsgränser:	Sammanlagt maximalt 60 poäng. På tentamen ges graderade betyg: 3:a 24 poäng, 4:a 36 poäng och 5:a 48 poäng
Hjälpmedel:	Skansholm, <i>Java direkt</i> , Studentlitteratur (understrykningar och mindre anteckningar i boken är tillåtna)

Inga kalkylatorer är tillåtna.

Tänk på:

- att skriva tydligt och disponera papperet på ett lämpligt sätt.
- att börja varje ny uppgift på nytt blad. Skriv endast på en sida av papperet
- Skriv ditt personnummer på *alla* blad.
- De råd och anvisningar som givits under kursen skall följas vid programkonstruktionerna. Det innebär bl.a. att onödigt komplicerade, långa och/eller ostrukturerade lösningar i värsta fall ej bedöms.

Uppgift 1) Om man inte hade haft tillgång till standardklassen `Math` hade man kunnat använda s.k. Maclaurin-serier för att beräkna värdet av vissa vanliga funktioner. Funktionen *sin* kan t.ex. beräknas med följande oändliga serie:

$$\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \frac{x^9}{9!} - \dots$$

Skriv ett klass `c` som innehåller en klassmetod `sin`. Metoden skall ha en parameter `x` av typen `double` och beräkna och returnera värdet *sin(x)*. Beräkningen skall ske med hjälp av serien ovan. Börja med att beräkna den första termen och beräkna sedan nästa term utgående från den förra. Upprepa detta tills den senaste termen som beräknats till beloppet blir mindre än 10^{-5} .

(10 p)

Uppgift 2) En engelsk motsvarighet till det svenska rövarspråket är *Pig Latin*. I detta språk bildas ord genom att man flyttar om och lägger till bokstäver enligt följande regler:

- Om den första bokstaven i ett ord är en vokal läggs ändelsen “*yay*” till sist i ordet. Ordet “*are*” blir t.ex. “*areyay*”. Observera att bokstaven *y* räknas som en konsonant på engelska.
- Om den första bokstaven i ett ord är en konsonant flyttas den första bokstaven till slutet av ordet och ändelsen “*ay*” läggs till allra sist. Ordet “*can*” blir t.ex. “*ancay*”. Undantag från denna regel är ord som börjar med vissa kombinationer av två bokstäver. Se nästa regel.
- Om den första bokstaven i ett ord är en konsonant och den andra bokstaven är någon av bokstäverna *hlprt* flyttas de två första bokstäverna till slutet av ordet och ändelsen “*ay*” läggs till allra sist. Ordet “*think*” blir t.ex. “*inkthay*”.

- a) Skriv en klassmetod `toPig` som får ett engelskt ord som parameter. Som resultat skall metoden ge ordet översatt till *Pig Latin*. I resultatet skall alla eventuella stora bokstäver vara översatta till motsvarande små bokstäver.

(6 p)

- b) Skriv ett program som läser en textfil som innehåller vanlig engelsk text. Programmet skall översätta texten till *Pig Latin* och skriva ut resultatet i en annan fil. Namnen på de två filerna skall ges som argument på kommandoraden när man startar programmet. Skulle det inte gå att öppna någon av filerna skall programmet ge en felutskrift och avslutas.

Översättning av orden till *Pig Latin* skall göras med hjälp av metoden `toPig` i uppgift a. (Det får du göra även om du inte löst den uppgiften.) I filen som läses kan det finnas tecken som inte är bokstäver, t.ex. punkt och frågetecken, omedelbart efter ett ord. Om så är fallet skall detta tecken inte skickas med i anropet av metoden `toPig`. Tecknet skall dock skrivas ut direkt efter det översatta ordet. Dessutom skall då en ny rad påbörjas i utskriften

Exempel. Om infilen innehåller texten:

```
Is this a Java program?
Yes it is!
```

Så skall följande skrivas till utfilen:

```
isyay isthay ayay avajay ogrampray?
esyay ityay isyay!
```

(6 p)

- Uppgift 3) Konstruera en klassmetod `randArray` som skapar ett fält i vilket den placerar ut heltal på ett slumpmässigt sätt. Metoden skall som resultat ge det fält som skapats. Metoden skall ha två parametrar, med namnen `max` och `upprepa`, av typen `int`. Den första parametern `max` anger vilka heltal som får förekomma i fältet. Fältet får bara innehålla heltal i intervallet 1 till `max`. Den andra parametern `upprepa` anger hur många gånger varje heltal skall förekomma i fältet. Längden på det resulterande fältet blir alltså `max*upprepa`. Varje anrop av metoden skall förstås ge olika resultat, även för samma parametrar.

Tips. Använd dig av metoden `random` i paketet `Math`.

Ett par exempel: Anropet `randArray(5,2)` skall som resultat ge ett fält med längden 10 där talen 1 till 5 finns slumpmässigt placerade. Varje tal finns 2 gånger. Resultatet kan t.ex. bli fältet: `[4, 2, 1, 1, 5, 3, 4, 2, 5, 3]`.

Anropet `randArray(3,4)` skall som resultat ge ett fält med längden 12 där talen 1 till 3 finns slumpmässigt placerade. Varje tal finns 4 gånger. Resultatet kan t.ex. bli fältet: `[1, 3, 2, 3, 2, 2, 1, 3, 3, 1, 1, 2]`.

(12 p)

- Uppgift 4) Som bekant kan klassen `CompoundBorder` användas när man vill sätta samman två ramar till en enda. Uppgiften är att konstruera en egen subclass till klassen `CompoundBorder`. Den nya klassen skall heta `MultiBorder`. Den skall kunna användas när man vill sätta samman ett godtyckligt antal ramar till en enda. Den nya klassen skall endast innehålla en konstruktor vilken har ett fält av typen `Border[]` som parameter. (`Border` är ett gränssnitt som implementeras av samtliga ram-klasser i `Swing`). Konstruktorns parameter innehåller referenser till de ramar som skall sättas samman. I konstruktorn skall den första ramen i fältet ligga ytterst och den sista innerst.

Tips: I klassen `CompoundBorder` finns två instansvariabler som är deklarerade på följande sätt:

```
protected Border insideBorder, outsideBorder;
```

Dessa skall innehålla referenser till de två ramar som sätts samman. I klassen `CompoundBorder` finns dessutom en parameterlös konstruktor som sätter dessa variabler till värdet `null`. (Det är alltså tillåtet för variablerna att ha värdet `null` om färre än två ramar sätts samman.)

(6 p)

- Uppgift 5) I denna uppgift skall man konstruera ett program som genererar ett fönster som ser ut som bilden på nästa sida. I centrum finns en panel på vilken texten *Java* placerats. Panelen omges av en ram som består av 10 delramar. Dessa delramar har ritats i 10 olika nyanser, från svart, via grått till vitt. Vad som inte framgår av bilden är att dessa delramar rör sig kontinuerligt. Fem gånger per sekund förskjuts varje delram ett steg utåt. Den yttersta delramen placeras då innerst. Detta ger illusionen av en rörelse inifrån och ut. Uppgiften skall utföras i två steg:

- a) Konstruera en subclass till klassen `JPanel`. Den nya klassen skall heta `LivePanel`. En komponent av denna klass skall fungera som en vanlig `JPanel`-komponent men den skall omges av en flerfärgad ram som rör sig så som beskrivits ovan.

Det skall finnas en konstruktor som skall ha två parametrar. Den första skall vara ett fält med referenser till objekt av standardklassen `Color` och den andra en `int` som anger hur tjock ramen skall vara (totalt för alla delramarna). Antalet delramar skall vara lika med antalet element i den första parametern och färgen för de olika delramarna anges av fältets element. Det första elementet i fältet anger färgen för den yttersta delramen och det sista elementet för den innersta delramen.

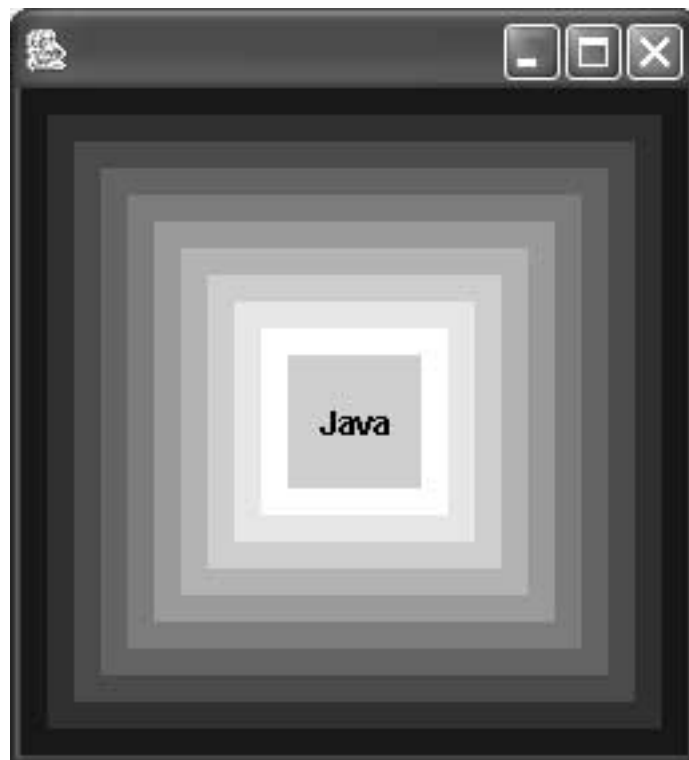
Förutom konstruktorn skall klassen `LivePanel` innehålla en metod med namnet `moveIn`. Denna har en parameter av typen `boolean` som anger om delramarna skall röra sig utifrån och in eller inifrån och ut. Värdet `true` anger rörelse inåt. Om man inte har anropat metoden `moveIn` skall rörelsen ske inåt.

Vid konstruktionen klassen `LivePanel` skall klassen `MultiBorder` från uppgift 4 användas.

Tips: Antag att det skall finnas n st delramar. Skapa då n st objekt av klassen `MultiBorder` vilka vardera innehåller n st delramar med de olika färgerna. För varje `MultiBorder`-objekt skall de olika delramarna ligga förskjutna ett steg. Anropa sedan metoden `setBorder` fem gånger per sekund och byt till ett nytt `MultiBorder`-objekt vid varje anrop.

(12 p)

- b) Skriv ett program som testat den nya klassen `LivePanel` från a-uppgiften. (Det får du göra även om du inte löst den uppgiften.) Programmet skall generera nedanstående fönster. En komponent av klassen `LivePanel` skall placeras mitt i fönstret. Denna skall initieras så att den omges av 10 delramar i 10 olika gråskalor från svart till vitt. Mitt i fönstret skall texten *Java* visas. Utforma fönstret så att `LivePanel`-komponenten (men förmodligen inte fönstret) blir kvadratisk med storleken 250 x 250 pixlar. Delramarna skall tillsammans ha tjockleken 100 pixlar.



(8 p)