

OBJEKTORIENTERAD PROGRAMVARUUTVECKLING

för IT1 (TDA545) och GU (DIT710)

OBS! Det kan finnas kurser med samma eller liknande namn på olika utbildningslinjer. Denna tentamen gäller *endast* för den eller de utbildningslinjer som anges ovan. Kontrollera därför noga att denna tentamen gäller för den utbildningslinje du själv går på.

TID 14.00-18.00

Ansvarig: Jan Skansholm, tel. 772 10 12 eller 0707-163230

Betygsgränser: Sammanlagt maximalt 60 poäng.
På tentamen ges graderade betyg:
CTH: 3:a 24 poäng, 4:a 36 poäng och 5:a 48 poäng
GU: G 24 poäng, VG 48 poäng

Hjälpmedel: Skansholm, *Java direkt med Swing*, valfri upplaga, 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 den (anonyma) kod du fått av tentamensvakten 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) a) Abstrakta klasser och gränssnitt (interface) har många likheter. Man kan därför fråga sig om båda konstruktionerna verkligen behövs. Nämn något som man kan göra med abstrakta klasser, men inte med gränssnitt. Nämn också något som man kan göra med gränssnitt, men inte med abstrakta klasser.

(2 p)

b) Som du vet får en instansmetod anropa andra metoder (både instansmetoder och klassmetoder) i samma klass direkt, utan att man skriver något objektnamn framför namnet på metoden som anropas. Men varför får en klassmetod inte anropa en instansmetod i samma klass på detta enkla sätt?

(2 p)

c) I ett program har man lagt följande rader:

```
String s1 = "HEJ";
String s2 = s1;
String s3 = "hej".toUpperCase();
if (s1==s2)
    System.out.println("lika");
else
    System.out.println("olika");
if (s1==s3)
    System.out.println("lika");
else
    System.out.println("olika");
```

Ange vilken utskrift programmet bör ge och förklara varför!

(2 p)

Uppgift 2) Två positiva heltal i och j kallas för *relativa primtal* om det inte finns något heltal större än 1 som de båda är delbara med. (Talen 16 och 21 är t.ex. relativa primtal, men däremot inte talen 18 och 21 eftersom de båda är delbara med 3.) Skriv en klassmetod `relPrim` som undersöker om två positiva heltal är relativa primtal. Om så är fallet skall metoden som resultat ge värdet `true`, annars värdet `false`.

Skriv sedan ett fullständigt program som gång på gång läser in ett par heltal från *en* dialogruta (Obs! inte två dialogrutor) och som undersöker om de två talen är relativa primtal eller inte. Programmet *måste* använda sig av funktionen `relPrim`. Resultatet skall meddelas i en annan dialogruta. När användaren klickar på knappen *Avbryt* i den första dialogrutan skall programmet avslutas.

(10 p))

Uppgift 3) Som du vet kan ett komplext tal beskrivas med hjälp av en realdel och en imaginärdel. Din uppgift är nu att definiera en klass i Java som beskriver begreppet komplext tal. Klassen skall (minst) ha följande konstruktörer och metoder:

- en defaultkonstruktör,
- en kopieringskonstruktör,
- en konstruktör som har en realdel och en imaginärdel som parametrar,
- två metoder som används för att avläsa ett komplext tals realdel respektive imaginärdel,
- en metod som adderar ett annat komplext tal till det aktuella komplexa talet (det aktuella komplexa talet skall alltså förändras),
- en metod som multiplicerar det aktuella komplexa talet med ett annat komplext tal (det aktuella komplexa talet skall alltså förändras). Tänk på att $i \cdot i = -1$,

- en metod med namnet `equals` som undersöker om det aktuella komplexa talet är lika med ett annat komplext tal. (När man jämför variabler av typerna `float` eller `double` bör man tänka på att värdena lagras approximativt. Någon bit kan därför vara olika för variabler som man logiskt uppfattar som "lika". När man jämför två sådana variabler bör man inte jämföra dem direkt, utan istället se om absolutbeloppet av deras skillnad är mycket litet, t.ex. 10^{-15} . I så fall kan man uppfatta variablerna som lika.),
- en metod `toString` som returnerar det komplexa talet som en text med formen "`a+b*i`" eller "`a-b*i`", beroende på om imaginärdelen är positiv eller inte.

Det skall inte vara möjligt att utifrån avläsa eller ändra ett komplext tal direkt. Allt måste göras via konstruktörer eller metoder.

(10 p)

Uppgift 4) Konstruera en klass `Lag` som beskriver fotbollslag som deltar i en serie. Klassen skall ha utifrån oåtkomliga instansvariabler för klubbnamnet, antal spelade matcher, totala antalet gjorda mål, totala antalet insläppta mål och antal poäng. Vinst ger 3 poäng, oavgjort 1 poäng och förlust 0 poäng. Klassen skall ha en konstruktor med vilken man kan initiera ett lag med ett visst namn. Dessutom skall det finnas en metod med namnet `nyMatch` som man anropar för att registrera ett matchresultat. Metoden skall ha två parametrar: antal mål som laget gjort i matchen respektive antalet mål som man släppt in. (Man får tre poäng för en vunnen match, en poäng för en oavgjord och noll poäng för en förlorad.) Det skall också finnas en metod `toString` som ger en text som motsvarar en rad i en tabell, t.ex.

```
"GAIS      3      3      2      4"
```

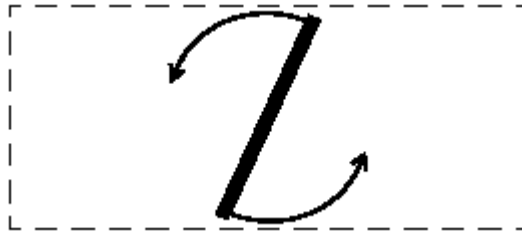
Utforma klassen `Lag` så att objekt av denna typ blir naturligt jämförbara. Jämförelserna skall göras så att det lag som har högst poäng kommer först. Om två lag har samma poäng skall det lag som har bäst målskillnad (skillnaden mellan gjorda och insläppta mål) komma först. Om två lag har samma poäng och målskillnad skall lagen hamna i bokstavsordning.

(10 p)

Uppgift 5) I det s.k. p-språket dubbleras alla vokaler (aouåeyäö) och ett 'p' placeras mellan de dubblade vokaler. Konsonanter och övriga tecken är oförändrade. Skriv ett program som läser en textfil innehållande ett hemligt meddelande skrivet i p-språket och som skriver ut det hemliga meddelandet i klartext i ett textfönster. Namnet på textfilen skall ges på kommandoraden som argument till metoden `main`. Om filen t.ex. innehåller texten "Apallapa läpär sipig Japavapa foport." så skall programmet skriva ut "Alla lär sig Java fort.". Utforma programmet så att det fungerar även när den hemliga texten (som i exemplet) innehåller både stora och små bokstäver. Om filen skulle råka innehålla felaktig text som inte är korrekt p-språk skall programmet ge en felutskrift och avslutas. Programmet skall också ge en felutskrift om felaktigt antal parametrar ges till programmet eller om filen inte existerar.

(12 p)

Uppgift 6) Konstruera en grafisk komponent som är en snurrande linje enligt figuren till här nedanför. Pilarna är inte en del av figuren utan markerar bara hur linjen ska rotera.



Den streckade linjen är inte heller en del av figuren utan markerar komponentens storlek. Komponenten ska vara en Swing-komponent och ha följande egenskaper:

- Linjen ska alltid fylla ut komponenten så gott det går men ändå vara lika lång oavsett vilken vinkel den för tillfället har. Med andra ord är det den minsta av bredden och höjden som avgör hur lång linjen blir.
- Linjen ska alltid vara centrerad i komponenten.
- Linjen ska hela tiden snurra automatiskt.
- Det ska finnas en metod `setRotationSpeed(double omega)` med vilken man kan ändra rotationens hastighet, uttryckt i varv per sekund. Man skall också kunna ange detta som parameter till konstruktorn när man skapar en ny komponent.

(12 p)