

Examination in

Objektorienterad programvaruutveckling IT1 TDA545

DAY: FRIDAY DATE: 2011-10-21 TIME: 14.00-18.00

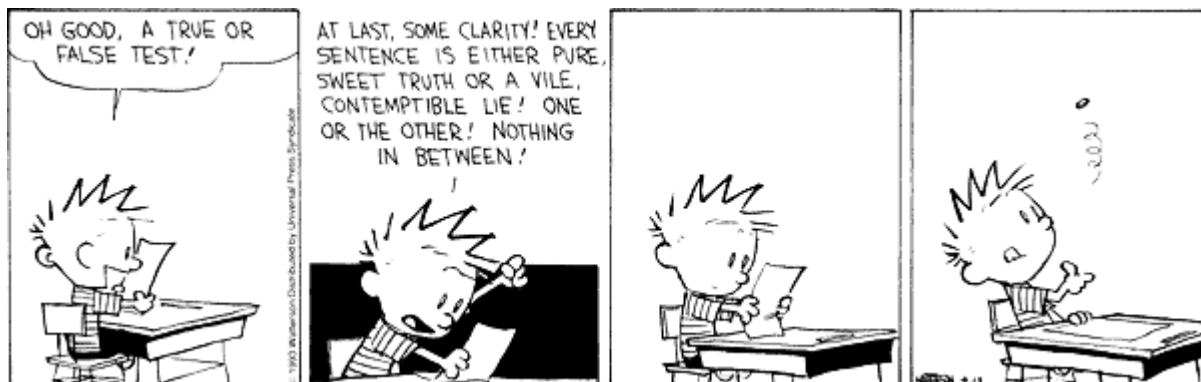
ROOM: M

Responsible teacher: Erland Holmström phone 1007, home 0708-710600
Results: Are sent by mail from Ladok.
Solutions: Are eventually posted on homepage.
Inspection of grading: The exam can be found in our study expedition after posting of results.
Time for complaints about grading are announced on homepage after the result are published or mail me and we find a time.
Grade limits: CTH: 3=28p, 4=38p, 5= 48p, max 60p
Aids on the exam: En sida **“Tentamnehjälpmedel – språkkonstruktioner i Java”**
utanför på denna får man också skriva vad man vill **på de vita ytorna**
textrutorna. Lappen lämnas in med tentan.

Observe:

- Start by reading through all questions so you can ask questions when I come. I usually will come after appr. 2 hours.
- All answers must be motivated when applicable.
- Write legible! Draw figures. Solutions that are difficult to read are not evaluated!
- Answer concisely and to the point.
- The advice and directions given during course must be followed.
- Programs should be written in Java, indent properly, use comments and so on.
- Start every new problem on a new sheet of paper.

Good Luck!

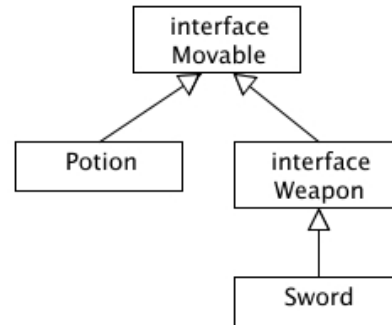


tryckfelen rättade

Problem 1. Uppvärmning:

- Antag `int i=2; int j=3;` Vad har följande uttryck för värde? (`double`) (`j/i`)
- Antag att `i` och `j` är `int`'s större än 0, vad är det minsta värdet på `i%j`?
- Antag att variabeln `i` innehåller ett tal i intervallet 0..99. Skriv satser som lagrar entalsiffran i variabeln `ones` och tiotalssiffran i variabeln `tens`.
Ex: `i=57` skall ge `ones = 7` och `tens = 5`.
- Antag att vi har arvshierarkin enligt figur till höger och definitionerna:

```
Movable movable;
Weapon weapon;
Sword sword;
Potion potion;
Sword klassen innehåller en metod:
    public void sharpen()...
Potion innehåller
    public Potion mix(Potion
                      other)...
```



Weapon innehåller

```
public void break()...
```

Vilka av de följande satserna är legala? (dvs ok att göra)

- | | |
|---|--|
| a) <code>movable = sword;</code> | e) <code>((Sword)movable).sharpen();</code> |
| b) <code>sword = weapon;</code> | f) <code>potion.mix(movable);</code> |
| c) <code>weapon = (Sword)potion;</code> | g) <code>potion.mix((Potion)movable);</code> |
| d) <code>weapon.sharpen();</code> | h) <code>sword.break();</code> |

(8p)

Problem 2. Testar: fält, loopar, metoder, parameteröverföring

Vad är det tänkt att följande program skall skriva ut och vad skriver det ut?
Förklara.

```
public class TestPassByValue {
    public static void swap(int n1, int n2) {
        int temp = n1; n1 = n2; n2 = temp;
    }
    public static void re(int[] a) {
        for (int i=0; i<a.length/2; i++) {
            swap(a[i], a[a.length-1-i]);
        }
    }
    public static void main(String[] args) {
        int[] a = {1,2,3,4,5,6,7,8,9};
        re(a);
        for (int i=0; i<a.length; i++) {
            System.out.print( (a[i]) + " " );
        }
        System.out.println();
    }
}
```

(3p)

Problem 3. Testar: förståelse av beräkningsordning och prioritet

Normal beräkningsordning är ju vänster – höger om inte prioritet bryter den. Men vad innebär det att programmet nedan skriver ut?

(Koden i den här uppgiften är konstruerad för att testa förståelsen av beräkningsordning och prioritet men att skriva kod med sånt här användande av globala variabler och funktioner som uppdaterar dom rekommenderas verkligen inte!)

```
public class Falsarium {
    static int i;
    public static int f() {
        i = i + 1;
        return i;
    }
    public static void main (String [] args) {
        int[] a = new int[7];
        int[][] aa = new int[7][7];
        //a) #####
        i = 5;
        i = i+f();
        System.out.println("a) i= " + i);
        //b) #####
        i = 5;
        i = f()+i;
        System.out.println("b) i= " + i);
        //c) #####
        i = 5;
        a[i] = f();
        System.out.println("c) i= " + i);
        for (int j=0; j<a.length; j++) {
            System.out.print(" " + a[j]);
        }
        System.out.println();
        //d) #####
        i = 5;
        aa[i][f()] = f();
        System.out.println("d) i= " + i);
        for (int j=0; j<aa.length; j++) {
            for (int k=0; k<aa[j].length; k++) {
                System.out.print(" " + aa[j][k]);
            }
            System.out.println();
        }
        System.out.println();
    }
} // end Falsarium
```

(4p)

Observera att lappen Exam Help skall lämnas in med tentan.

Problem 4. Testar: rekursion

Antag att din dator klarar av rekursion men inte vanlig addition :-).

Den kan bara addera/subtrahera *ett* till ett annat (valfritt) tal. Om man skall addera 9+3 måste man alltså låta datorn addera "1" tre gånger, dvs (((9+1)+1)+1) dvs som (9+3) = (10+2) = (11+1) = (12+0) = 12.

Skriv en rekursiv funktion som givet två heltal ≥ 0 returnerar deras summa.

Om något av talen är negativt skall en `IllegalArgumentException` kastas.

Antalet rekursiva anrop skall minimeras, alltså skall 3+9 och 9+3 beräknas på samma sätt nämligen som (((9+1)+1)+1) och inte som (((((((((3+1)+1)+1)+1)+1)+1)+1)+1). Och absolut inte som (9+(1+(1+1))).

(6p)

Problem 5. Testar: loopar, interface, interface som parameter,

I den här uppgiften väljer du att svara på **en** av deluppgifterna. Svårighetsgraden är stigande, liksom poängen. *Glöm inte ange vilken av deluppgifterna du svarar på, det är just den som kommer att poängsättas.*

Antag att $f(i)$ är definierad enligt `double f(double i)`. (Den matematiska benämningen "funktion" motsvarar en metod som returnerar ett värde i Java.)

a) Skriv en funktion, `sigma1`, som givet ett positivt tal n beräknar $\sum_{i=0}^n i$ (2p)

b) Skriv en funktion, `sigma2`, som beräknar $\sum_{i=0}^n f(i)$. f är en given (dvs deklarerad tidigare i programmet), men, just nu, okänd funktion (tex x^2). (Du skall alltså INTE skriva en funktion som summerar x^2 utan en som summerar funktionen f .) (3p)

c) Antag nu att vi vill summera *olika* funktioners värde i ett intervall enligt:

$\sum_{i=a}^b f(i)$ Skriv en funktion, `sigma3`, som tar en funktion och ett

heltalsintervall (a och b) som parametrar och ger summan av f i det intervallet. Utgå från ett interface för f som du också skriver (kalla det för `FunctionF`).

Skriv också en funktion (`FunkX2`) som implementerar interfacet och ett huvudprogram som adderar funktionen x^2 mellan $a=1$ och $b=5$. (7p)

d) Här blir det lite svårare men klarar ni c) så borde ni klara även detta, bara generalisera lite till. Men det kan också vara strategiskt bättre att göra c om du är osäker på lösningen här.

Utöka funktionen (kalla den `sigma4`) så att även operatorn (dvs "addition" i uppgifterna ovan) kan varieras så att man kan addera, subtrahera, multiplicera, ... funktionsvärdena.

Tips1: Du behöver ett interface till för operatorn.

Skriv också klasser som implementerar interfacet och ett huvudprogram som *multipliserar* funktionen x^2 mellan 1 och 5. (dvs $1^2 * 2^2 * 3^2 * \dots 5^2$)

Tips2: När man adderar/subtraherar så använder man 0.0 som

"identitetselement" dvs man kan börja med $0.0 + \dots$ i loopen. När man

multipliserar/dividerar så måste man använda 1.0 som identitetselement

(annars blir ju resultatet alltid 0.0). Man kan lösa det genom att lägga till en

metod i det nya interfacet `getIdentityElement()`. (12p) (totalt max 12p)

Problem 6. Testar: fält, booleaner, swing, händelsehantering, grafik, klasser

a) Implementera följande klass:

```
public class FiverModel {
    public FiverModel()
    public boolean status(int x, int y)
    public void chose(int x, int y)
}
```

Klassen har en instansvariabel, ett tvådimensionellt fält med 5x5 booleaner som initieras till false.

Metoden `status` returnerar värdet på rad x och kolumn y.

Metoden `chose` ändrar värdet på de 5 positionerna

(x,y), (x-1,y), (x+1,y), (x,y-1) och (x,y+1)

om positionerna existerar i matrisen.

Tips: `chose` måste se upp om man anger en punkt x,y på randen dvs första/sista rad respektive kolumn annars får man `ArrayIndexOutOfBoundsException`.

Antingen måste man testa ordentligt eller också kan man låta fältet vara en kolumn större än det ser ut när man använder klassen dvs fältet är 7x7 här. Då måste man å andra sidan dölja det för omvärlden som ju skall tro att det är 5x5.

b) I spelet Fiver skall man försöka konvertera alla vita fyrkanter till svarta. När man klickar på en knapp så byter den färg precis som dess grannar på sidorna och ovan och nedanför. När programmet startar får man upp en 5x5 kvadrat med vita knappar som i fig 1.

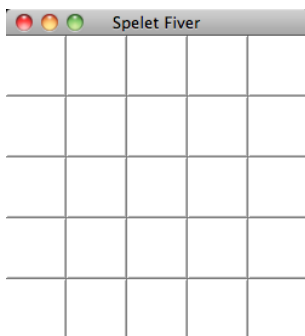


fig 1

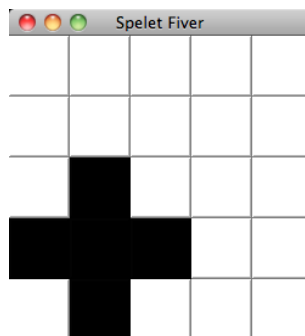


fig 2

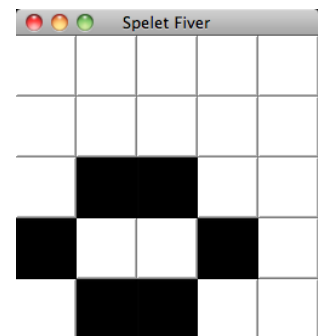


fig 3

I fig 2 har jag klickat på knappen på rad 4 och kolumn 2 (numrerade 1..5) och i fig 3 har jag också klickat på 4,3.

Implementera spelet Fiver så det fungerar på det sättet. Några krav:

- Använd klassen i a) utan att ändra i den.
- Du skall rita i en *panel* (`FiverPanel`) vars konstruktor tar en `FiverModel` som parameter och allt skall startas från ett *huvudprogram* med en `JFrame`.
- Du skall ge knapparna en storlek och en ram (ramen gör dessutom att default utseendet försvinner. I alla fall på min mac så spelar det stor roll som du ser ovan).
- Det är inte ok att skapa en hanterare för varje knapp.

Tips:

- Du kan använda en knappens bakgrundsfärg om du gör den ogenomskinlig.
- Det är ok med en "ful"-lösning i `actionPerformed` för att hitta vilken knapp som tryckts, tex att söka igenom knapparna.

Nu är det bara en uppgift kvar..

Problem 7. Testar: exceptions, enkla kontrollstrukturer, användande av färdiga klasser

En A/D omvandlare är kopplat till någon form av analog signal tex en voltmätare, och omvandlar den analoga signalen till digitala värden tex i form av en double.

Antag att en A/D-omvandlare representeras med en klass med signaturen

```
public class RealADConverter implements ADConverter{

    public RealADConverter(double lower, double upper) {...}
    public double read() throws UnderVoltage, OverVoltage
    {...}
} // end RealADConverter
```

Metoden `read()` läser A/D-omvandlarens värden och returnerar dem. Om värdena ligger utanför intervallet `lower..upper` kastas en exception, `UnderVoltage` eller `OverVoltage`. Man kan alltså bestämma mellan vilka värden man accepterar att mätvärdena skall vara vid konstruktionen av objektet. Denna klass är given.

Du skall skriva en metod som skapar och använder ett sådant objekt

```
public void runExperiment(int maxNoise,
                          int lower, int upper) ...
```

`runExperiment` anropar kontinuerligt `read()` och skriver ut det värde som returneras. När totalt `maxNoise` exceptions rests från `read()` så skall exekveringen avbrytas och en exception, `TooMuchNoise`, kastas av `runExperiment`.

Alla exceptions (`UnderVoltage`, `OverVoltage`, `TooMuchNoise`) finns färdiga att använda.

(7p)

Observera att lappen Exam Help skall lämnas in med tentan.

(Jag vill se vad ni skriver så jag kan göra en bättre "Exam Help", ni får tillbaka den med tentan igen.)

Du har väl svarat på kursenkäten?