



Offset til terminalskærm i Java

Ikke en artikel - men en Klasse til java der gør det nemt at skrive ud på avancerede måder når man kun opererer i dosvindue. Kun et værktøj du kan bruge som vi vil :-)

Skrevet den **19. Feb 2009** af **minau** I kategorien **Script / Andre**

```
/*
Dette er en klasse til java der gør det nemt at skrive ud på avancerede måder når man kun opererer i
dosvindue. Alle featurer skulle være forklaret. Men brug din fantasi - det skulle kunne kombineres på
diverse kreative måder :-)
```

```
*/

import java.util.*;
import java.io.*;
import java.lang.*;

// Offset ver 5.2.1
//   Gør det muligt at skrive ud på en doskærm hvor man vil
//
// - void SetSize(SKAL ANGIVES - valgfri: int Højden på skærmen) - hvis ingen parametre angives vil
brugerens kunne bestemme det.
//
// - void PrintBuf(); Udskriver Buffer på skærmen.
//
// - void Add(String der bliver skrevet til Buffer, int x koordinat for hvor det bliver skrevet fra (4 sekant), y
koordinat)
//
//           Er værdierne 0 vil elementet det bliver skrevet blive centreret.
OBS- Er y negativ vil der blive talt fra hvor der sidst er skrevet til -1
//
// - String StringToBox(String der returneres som inde i en kasse, int x space mellem txt og kan, int y space,
char (Valgfri) giver box med denne kant)
//
//           er værdien i space negativ vil spacingen blive målt fra kanten TIL
kassen
//
// - void AddBox(samme som Add, int x space mellem txt og kan, int y space, char (Valgfri) giver box med
denne kant)
//
//           er værdien i space negativ vil spacingen blive målt fra kanten TIL
kassen
//
// - String - MakeMenu(String[] array med de punkter der skal skrives til en menu - valgfri: int Space
mellem txt og linjer, char hvad vandrette linjer er lavet af, char hvad lodrette linjer er lavet af)
//
// - void AddMenu(int for den linje listen skal begynde på, Det samme som MakeMenu)
//
// - void AddBoxCenter(samme som AddBox - men teksten vil være centreret i boxen)
//
// - void Clear() - Ryder Buffer
//
```

```

// - void Clear(int x1, int y1 koordinat for øverste venstre hjørne, int x2,int y2 koordinater for nederste højre
hjørne)
//
//          Ryder en firkant fra hjørnerne (x1,y1) til (x2,y2)
//
// - void AddBufferMenu(String - skriver et menupunkt til en menubuffer)
//
// - String[] GetBufferMenu() - returnere et arrays af strings (fra AddMenuBuffer) der kan bruges som
punkter i menuen - cleaer AddMenuBuffer
//
// - String[] GetBufferMenu(ArrayList) - returnere et arrays af strings (fra det parsede) der kan bruges som
punkter i menuen - cleaer
//
// String CenterString(string der vil blive returneret hvor linierne står centreret over hinanden)
//
// int MaxString(Array) - returnere antallet af char i den længste instans af arrayet

```

```

public class Offset
{

```

```

    public static int BRED = 79; //bred = 79 og høj = 24 er standart for normalt
    public static int HOJ ;      // eller 79 og 24 for max for maximeret vindue - men dette sættes med
SetSize()
    public static int yhvor = 0;
    public static char[][] Buffer;
    public static ArrayList MenuBuffer = new ArrayList();

```

```

/* public static void main(String[] args) throws Exception
{

```

```

    //eksempel på hvordan forskellige ting kan se ud
    SetSize();

```

```

    //AddBox("012345623456789012345678901234567890123456789",1,1,0,0);
    //Add("20\n21\n22\n23\n24\n25\n26",1,20);
    //PrintBuf();
    //System.out.println(MakeList(new String[]
{"Førskjhlfs","kjglkjhdfljhlfs","kjglkjhdfljhlfs","kjglkjhdfljhlfs","kjglkjhdfljhlfs","kjglkjhdfljhl
fs","kjglkjhdfljkhldshkd dlkj kdls kldskljsdvxs"},1,'*','*'));
    //System.out.println(MakeList(new String[] {"Alldsfsgdsfgdsfgadfsf"},'*','*'));
    //PrintBuf();
    //SetSize();

    //Add("sdfsdfsdf",5,1);
    //AddBufferMenu("tingleing");
    //AddBufferMenu("tin2gleing"); //er god at kombinere med en forlykke når man nu ikke ved hvor
mange elementer der vil være i en menu
    //AddBufferMenu("dfsflg hdsfl kjghsdlg \njlksk lkdhflkjsdfg");
    //AddMenu(1,GetBufferMenu());
    //PrintBuf();
    //Clear();
    //PrintBuf();

```

```

}
*/
public static void AddBox(String txt, int x, int y, int xspace, int yspace){
    Add(StringToBox(txt,xspace,yspace),x,y);
}

public static void AddBox(String txt, int x, int y, int xspace, int yspace, char kant){
    Add(StringToBox(txt,xspace,yspace,kant),x,y);
}

public static void AddBoxCenter(String txt, int x, int y, int xspace, int yspace){
    Add(StringToBox(CenterString(txt),xspace,yspace),x,y);
}

public static void AddBoxCenter(String txt, int x, int y, int xspace, int yspace, char kant){
    Add(StringToBox(CenterString(txt),xspace,yspace,kant),x,y);
}

public static String CenterString(String txt){
    String[] linier = txt.split("\n");
    String spacer = "";
    int xmax = MaxString(linier);

    for(int j=0;j<linier.length;j++){
        spacer = "";
        for(int k=0;k<((xmax-linier[j].length())/2);k++){
            spacer+=" ";
        }

        linier[j] = spacer+linier[j];
    }

    String out = "";
    for(int l=0;l<linier.length;l++){
        out += linier[l]+"\\n";
    }

    return out;
}

public static String StringToBox(String txt, int xspace, int yspace){

```

```
return StringToBox(txt,xspace,yspace,(char)4);  
}
```

```
public static void AddBufferMenu(String item)  
{  
    MenuBuffer.add(item);  
}
```

```
public static String[] GetBufferMenu()  
{  
    String[] punkter = new String[MenuBuffer.size()];  
    for (int i = 0;i<MenuBuffer.size();i++)  
    {  
        punkter=(String)MenuBuffer.get(i);  
    }  
    MenuBuffer.clear();  
  
    return punkter;  
}
```

```
public static String[] GetBufferMenu(ArrayList Listen)  
{  
    String[] punkter = new String[Listen.size()];  
    for (int i = 0;i<Listen.size();i++)  
    {  
        punkter[i]=(String)Listen.get(i);  
    }  
  
    return punkter;  
}
```

```
public static void AddMenu(int y, String[] punkter)  
{  
    Add(MakeList(punkter, 1,'/','|'),1,y);  
}
```

```
public static void AddMenu(int y, String[] punkter,int yspacer, char hen, char ned)  
{  
    Add(MakeList(punkter,yspacer,hen,ned),1,y);  
}
```

```

public static String StringToBox(String txt, int xspace, int yspace, char kant){
    String boxString = "";
    String spacer = "";
    String top = "";
    int xmax = 0;
    int xspaceH = xspace;
    double temp = 0.0;
    String[] linier = txt.split("\n");

    xmax = MaxString(linier);

    if (xspace < 0){
        temp = (BRED - xmax + (2 * xspace)) / 2.0;
        xspace = (int) temp;
        xspaceH = xspace;
        if (0.5 < Math.round(temp) - xspace){ //Tjekker om der skal rundes op i højre side
            xspaceH++;
        }
    }

    if (yspace < 0){
        yspace = (HOJ - linier.length + (2 * yspace)) / 2;
    }

    for (int i0 = 0; i0 < (xmax + xspace + xspaceH + 2); i0++){ //Skriv øverste *
        top += kant;
    }

    for (int i4 = 0; i4 < yspace; i4++){ //Lav spacer
        spacer += kant;
        for (int i1 = 0; i1 < xspace; i1++){
            spacer += " ";
        }
        for (int i2 = 0; i2 < xmax; i2++){
            spacer += " ";
        }
        for (int i3 = 0; i3 < xspaceH; i3++){
            spacer += " ";
        }

        spacer += kant + "\n";
    }

    for (int i5 = 0; i5 < linier.length; i5++){ // skriv txt
        boxString += kant;
        for (int i6 = 0; i6 < xspace; i6++){
            boxString += " ";
        }
        boxString += linier[i5];
    }
}

```

```

    for(int i7=0;i7<(xspaceH+(xmax-linier[i5].length()));i7++){
        boxString += " ";
    }

    boxString += kant+"\n";
}

String box = top+"\n"+spacer+boxString+spacer+top;

return box;

}

public static void Add(String add, int x, int y){

    String[] linier = add.split("\n");
    String temp = "";
    int xmax = MaxString(linier);

    if(x<=0){ //0 for hvis der skal centreres på x axen
        x = (BRED-xmax)/2;
        if (x<0){x=0;} //Hvis strngen er for lang vil det give negativ - det går ikk
    }else{
        x=x-1; //ellers skal x trækkes én fra fordi det er i et array (starter på [0])
    }

    if(y==0){ //hvis y = 0 skal der centreres på y axen.
        y = (HOJ-linier.length)/2;
        if (y<0){y=0;} //Hvis strngen er for høj vil det give negativ - det går ikk
    } else{
        if (y<0){ //er y negativ betyder det at y skal tælle fra dér hvor der sidst er skrevet
            y = y+((y)*(-1)-1); //Minus én skriver til den række der lige er skrevet (derfor -1 i denne
linje) - da 0 jo var at centrere
        }
        y=y-1; //ellers skal y trækkes én fra fordi det er i et array (starter på [0])
    }

    temp = ""; // burde ikke være nødvendig - men bare for en sikkerhedsskyld
    if(x<0||BRED<x){temp+="fejl i koordinat x="+x;x=0;}
    if(y<0||HOJ<y){temp+=" fejl i y koordinat y="+y;y=0;}
    linier[0] = temp+linier[0]; //Ved Debug

    for(int i=0;i<linier.length;i++){
        if (x+linier[i].length()>BRED){ //tjekker om linjen er for bred til buffer

            // ved debugg: linier[i] = "linjen Der starter med \""+linier[i].substring(0,20)+"\" er
"+linier[i].length()+" bred (max "+BRED+"));
            linier[i] = linier[i].substring(0,BRED-x-2)+"->";
        }
    }
}

```

```

        for(int j=0;(j<linier[i].length() && j+x<BRED );j++){ // skriver til buffer tegn for tegn - lavef ->
        hvis linjen er for lang
                //Ved debug: System.out.println("x="+x+" - "+"y="+y+" - "+"i="+i+" -
        "+"j="+j); // Ved debug
                Buffer[y+i][x+j]=linier[i].charAt(j);

                if(y+i+1>yhvor){
                        yhvor=y+i;
                }
        }
        if(y+i+1>=HOJ){ // tjekker om den næste gang kommer til at skrive ud
        neden for buffer
                linier[linier.length-1]="Data er noteret, men der er ikke plads til at skrive mere.";
                y=y-1;
        }
}
}

```

```

}

```

```

public static void Clear(){
    Buffer = new char[HOJ][BRED];
    yhvor=0;
}

```

```

public static void Clear(int x1, int y1, int x2, int y2){
    //x1--;y1--;x2--;y2--; //Det bruger indtaster er i et koordinatsystem - og det er jo fra et array
    if(x1<=BRED && x2<=BRED && y1<=HOJ && y2<=HOJ && x1<=x2 && y1<=y2){
        for(int i = 0;i<(y2-y1+1);i++){
            for(int j = 0;j<(x2-x1+1);j++){
                //System.out.println("x1="+x1+" - "+"y1="+y1+"x2="+x2+" - "+"y2="+y1); // Ved debug
                Buffer[y1+i-1][x1+j-1]=" ".charAt(0);
            }
        }
    }
    }else{ System.out.println("Fejl i paramerteangivelsen af offset.clearArea");}
}

```

```

}

```

```

public static void PrintBuf(){
    String out = "\n";
    for(int ned = 0;ned<HOJ;ned++){
        for(int hen = 0;hen<BRED;hen++){
            out += Buffer[ned][hen] ;
        }
        out += "\n";
    }
}

```

```

}

System.out.print(out);

}

static int MaxString(String[] linier){ //returnere den længste linje i Arrayet
    int xmax = 0;
    for(int i=0;i<linier.length;i++){ //find bredden af txt
        if (linier[i].length()>xmax){
            xmax = linier[i].length();
        }
    }
    return xmax;
}

```

```

public static void SetSize() throws Exception {
    int min;
    int max;

    while(HOJ<10){

        for (int i =1;i<100;i++){
            System.out.println(i)    ;
        }
        min = GetInput.Int("Hvad er det mindste tal du kan se?");
        max = GetInput.Int("Hvad er det stoerste tal du kan se?");

        HOJ = max-min+1;
    }
    Buffer = new char[HOJ][BRED];

}

public static void SetSize(int y) throws Exception {
    HOJ = y;
    Buffer = new char[HOJ][BRED];

}

```

```

public static String MakeList(String[] punkter,int yspacer, char hen, char ned){
    int hvormange = punkter.length; //hvor mange punkter menuen skal have
    String bar = "";
    int i=0;
    int nmax = 0;

    for(i=0;i<BRED;i++){
        bar += hen ;    // Linje der adskiller felterne skrives
    }
}

```

```

int hvilkeSkrevet=0;

```



```

String map = "";
String out = "";
int skrevet=0;
int temp;

while(hvilkeSkrevet<punkter.length){ //GEntages til alle punkter er skrevet ud
    int langde = 1;
    while(hvilkeSkrevet<punkter.length){ //Finder du af hvor mange punkter der skal være i
dette niveao
        temp = MaxString(punkter[hvilkeSkrevet].split("\n"));
        if((langde+temp+((hvilkeSkrevet+1-skrevet)*3 ))<BRED){ //Hvis det at lægge næste punkt
(hvert punkt kræver 3 extra pladser hver (ned variabel + én spaci i hver side)) til i rækken IKKE
overskrider siden skal det gøres
            langde += temp;
            hvilkeSkrevet++;
        }else{
            break;
        }
    }
}

// De overskydende felter skal fordeles mellem de forkellige
punkter. x2 fordi der jo er space før og efter
int xspace = (int) (BRED-(langde+(hvilkeSkrevet-skrevet)))/ ((hvilkeSkrevet-skrevet)*2);

int luft=0;
/*
System.out.println("Hvor mange skrives i denne linje" +(hvilkeSkrevet-skrevet));
System.out.println("bred-langde+1 :"+(BRED-langde));
System.out.println("xspace:"+xspace);
*/
// find ud af hvor høj (linjeskift) der er maximalt i niveauet
for (i=0+skrevet;i<hvilkeSkrevet;i++){
    if (punkter[i].split("\n").length>nmax){
        nmax = punkter[i].split("\n").length;
    }
}

//System.out.println("yspacer: "+yspacer);
for(int g=0-yspacer;g<nmax+yspacer;g++){ //hvert niveao har yspacer+nmax(svare til
maximale antal linjer i instans)+yspace
    map += "\n";
    for (int l=0+skrevet;l<hvilkeSkrevet;l++){ //Hver linje i hele niveauet skal skrives ud é af
gangen.
        map += ned;

        for(int l1=0;l1<xspace;l1++){ //de tildelte felter skal skrives før txt
            map += " ";
        }
}

```

```

        if (g<0||g>=nmax){           // g (linje i niveauet) er uden for 0 til nmax vil det sige at det er
der hvor der skal være yspace
            luft=1;
        }else {
            if(punkter[l].split("\n").length>g){ // hvis der står noget i punktet på denne linje skrives det
ud
                map += punkter[l].split("\n")[g];
                for(int l4=0;l4<(MaxString(punkter[l].split("\n"))-punkter[l].split("\n")[g].length());l4++){
                    map += " ";           // Det skal fylde lige så meget som den linje i dette punkt der er
længst
                }
            }else{
                luft=1;           // Hvis der ikke står noget på denne linje i punktet - skal den fylde det
hele ud med luft
            }
        }

        if (luft>0){
            for(int l4=0;l4<MaxString(punkter[l].split("\n"));l4++){
                map += " "; // hvis det ikke står noget i denne linje eller hvis der er tale om spacing
skal der skrives luft
            }
            luft=0;
        }

        for(int l3=0;l3<xspace;l3++){ // spacing skal jo også skrives ud tilsidst
            map += " ";
        }

    }
    while(map.split("\n")[map.split("\n").length-1].length()<BRED-1){ // de felter der ikke er fyldt ud
skal skrives før at den sidste stjerne kan skrives
        map+=" ";
    }
    map +=ned;

}
skrevet= hvilkeSkrevet;
out += "\n"+bar+map;
map = "";
}
out += "\n"+bar;

return out;
}
}

```

// tanks and credits to arne_v :-)

// nanog --> hvordan vil du udbrede noget som du bare vil udbrede. Dette er set som en hjælp til dem der skal til at lave et eller andet i prompt - men (lige som jeg) drukner i at nægte at acceptere hvor kedeligt et program man bruger mange timer på bliver hvis der bare spørges linje for linje - uden [i]ldt grafisk.
// står der enlig at artikler skal være beskrivende?

Kommentar af nanoq d. 05. Mar 2004 | 1

Dette er ikke en artikel. En artikel skal man være beskrivende. Dette er blot et eksempel.

Kommentar af borriholt d. 05. Sep 2008 | 2

INGEN med respekt for sig selv gidder at udvikle et konsol program ... File --> New project og man har et GUI program .. Så klllasker man enn knap på trykker to gange på den og skriver noget kode .. .vupti så er man i gang ... SELV c++ verden har opdaget det, med c# godt nok 10 senere end os andre .. Men alle er velkommene i GUI kluppen