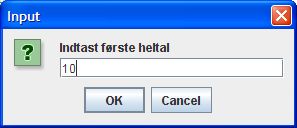
# 15.0 Brugergrænseflader: GUI (Grafic User Interface).

Denne lektion giver eksempler på dialogbokse, som vil give en pæn bruger-grænseflade.

Dette er ”ren grafik”, da det kun bidrager til det visuelle indtryk af pro­gram­met. Det har ingen indflydelse på det ”rigtige” program, der kan løse en op­gave for brugeren.

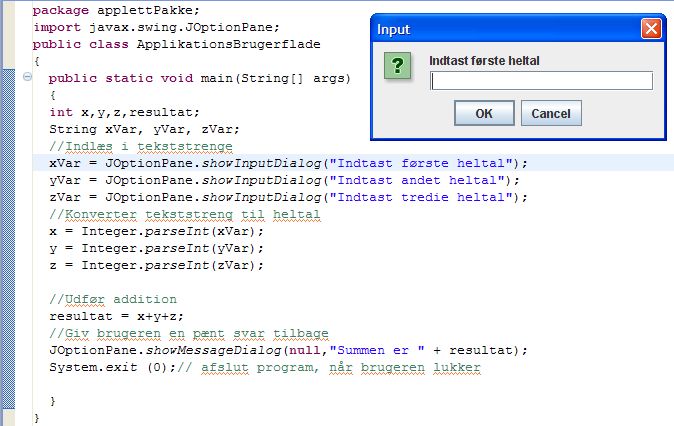
## 15.1 Brug af dialogbokse i en applikation.

I har længe måtte klare jer uden grafik, da vores applikationer[[1]](#footnote-1) er blevet af­vik­let i editoren *eclipse*’s afviklings-vindue *console*.

I programmet med navnet ”Applikations­Bruger­flade.java” vises et eksempel, der demon­stre­re dialogbokse, der kan modtage et indput:   
 *JOptionPane.showInputDialog("Valgfri tekst til boksen").*

og sende en meddelelse:  
 *JOptionPane.showMessageDialog(null, "meddelelse”);*

Først må vi importere biblioteket/pakken *javax.swing* som er en samling klasser skrevet til at lave vinduer.  
Derefter navn­giver vi vores program og starter en *main*-metode ligesom vi plejer. I linierne 7 og 8 definerer vi variabler. Bemærk at vi bl.a.får brug for variable af typen *String*. Dvs. tekst-strenge, der består af en række tegn – *char*.  
I linie 10-12 benytter vi os af klassen *JOptionPane* ved at kalde metoden *showInputDialog( )* fra denne klasse. Som argumenter til metoden sender vi den tekst, vi ønsker vist til brugeren.  
I linie 14-16 konverteres tekststrengen til heltal. (Se næste afsnit for ind­læs­ning af doubler).



I

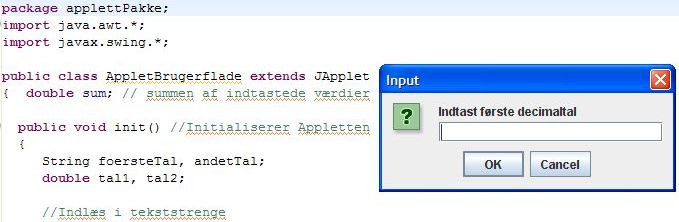
Argumentet – inde i parentesen – er her af typen *String*, mens metodens output er af typen *integer*.

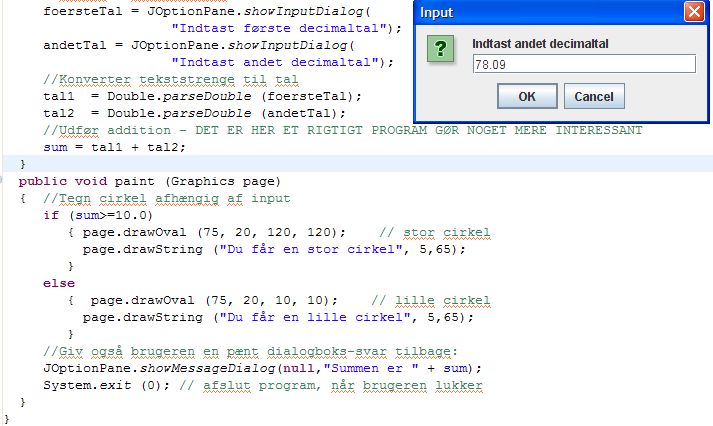
linie 19 udføres det egentlig program. Det kunne være en 2.grads­lignings­løser som vi har lavet tidligere. Her er det blot en simpel addition.

I linie 21 gives et svar – en meddelelse til brugeren -, og endelig lukkes alle vinduer og programmet afsluttes i linie 22, når brugeren klikker på OK.

## GUI_AppletInit15.2 Brug af dialogbokse til en applett.(se lektinsplanerne 17 og 18 om Appletter)

På næste side er vist et eksempel på hvordan dialogbokse også kan bruges sammen med en applett.   
Indtastningen boksene kaldes fra *init( )*– metoden. Det er den metode der initialiserer appletten.



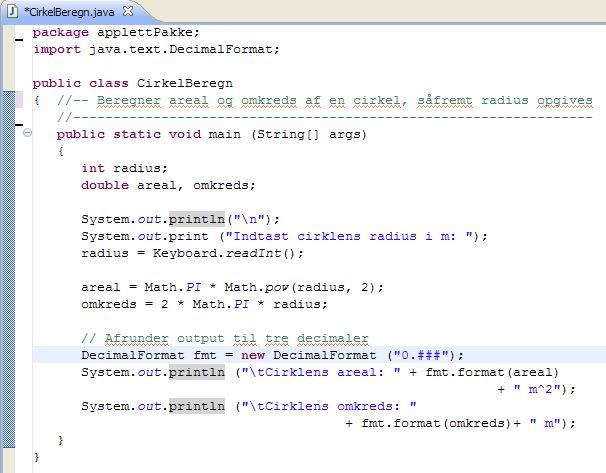


I *paint( )* – metoden tegnes appletten. I dette eksempel har indtastningerne betyd­ning for summen og om der vælges stor eller lille cirkel.

## 15.3 Afrunding af decimaltal

Til sidst et kort eksempel på hvordan decimaltal kan formateres, så betydende cifre før komma kan bestemmes samt antallet af decimaler.

Bemærk at det er nødvendigt at skabe et objekt fra klassen *DecimalFormat*. Idet vi beder om at få skabt et sådant objekt, angiver vi typen ved at sende et argument med til kontruktøren:  
 (linie 20) *DecimalFormat fmt = new DecimalFormat ("0.###")*Antallet af 00 og ## angiver hvor mange pladser vi ønsker henholdsvis før og efter kommaet. Når vi så har et tal, vi vil have formateret på præcis denne måde, kalder vi metoden: ( linie 21) *DecimalFormat.format(areal)*

**

**Opgave 15.1**   
Find dit gamle program der løser 2.-gradslininger. Tilføj dialogbokse og afrund evt. skæringer før de serveres i en boks.

\_\_\_ \_\_\_ \_\_\_\_

1. En applikation er et program, der kan køres selvstændigt - blot man har en java-oversætter. Applikationer kan kendes på *Main*-metoden. [↑](#footnote-ref-1)