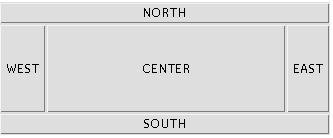
## 20.0 Grafisk brugergrænseflade (GUI, graphical user interface) med swing og awt(abstract windows toolkit)

En container kan indeholde komponenter og styre hvordan disse vises på skærmen. Her vil vi benytte JFrame og JPanel. Alle containere har en *layout-manager*, der styrer hvordan de enkelte komponenter placeres. Alle containere har egenskaben layout, der kan sættes med metoden setLayout(Layout l).

* **Null-Layout**(ingen styring)
* **FlowLayout**. Placerer komponenter ligesom centreret tekst. Når der ikke er plads i første række fortsættes på næste. FlowLayout er standard i paneler og applets.

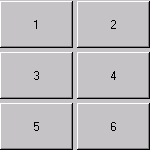
Kilde: javabog.dk

* **BorderLayout**. Tager højde for vinduets størrelse og tilpasser komponenternes størrelse efter aktuelle plads i vinduet/panelet.

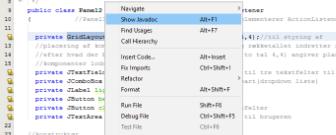
Kilde: javabog.dk

Hvis man skal have flere komponenter i et område, placerer man først et JPanel, og komponenterne tilføjes så panelet. BorderLayout er standard for et vindue.

* **GridLayout**. Med metoden setLayout(new GridLayout(3,2); oprettes et gridlayout med 3 rækker og 2 søjler. Komponenter som tilføjes, placeres rækkevis. Her eksempel med 6 knapper som tilføjes fra 1 til 6.

Kilde: <https://docs.oracle.com/javase/8/docs/api/java/awt/GridLayout.html>

Husk i NetBeans kan man markere højreklikke på kommando, som man gerne vil have Javadokumentatinen for, og så vælge Show Javadoc



* **GridBagLayout**. Her kan man styre hvordan komponenter skal placere i panelet/vinduet efter et usynligt gitter. En komponent kan en eller flere celler i højde og bredde.

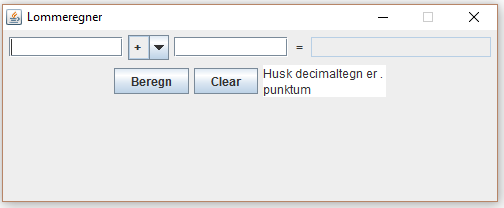
Kilde: javabog.dk

Se javabog.dk eller Oracles dokumentation: <https://docs.oracle.com/javase/8/docs/api/java/awt/GridBagLayout.html>

### *20.1* *Eksempel på lille lommeregner lavet med FlowLayout.*

Lommeregneren har to indtastningsfelter og et resultatfelt. Man kan vælge mellem de fire regningsarter + - \* og /. Der er en Beregn knap og en Clear knap samt et tekstareal til meddelelser.

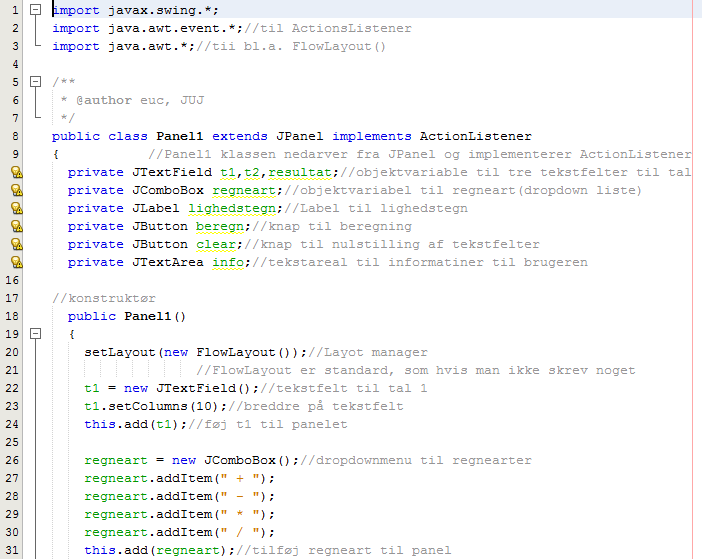
I følgende eksempel er der anvendt FlowLayout til styring af komponenter. Ved at sætte størrelsen på vinduet, får man placeret komponenterne nogenlunde som man gerne vil have det.

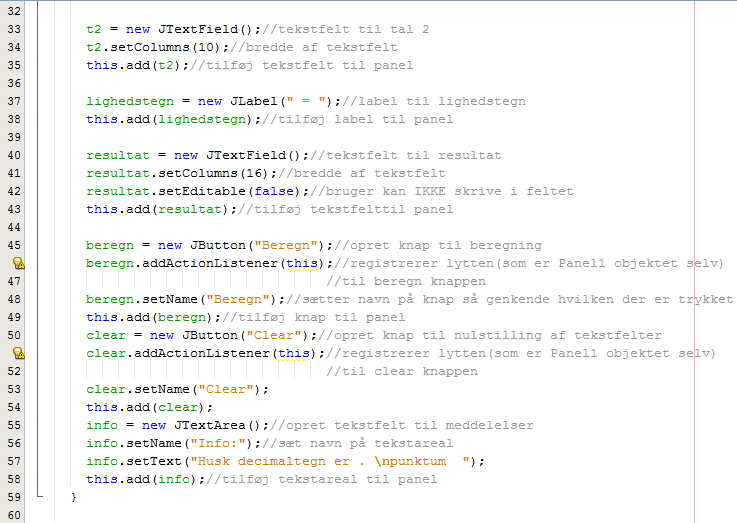


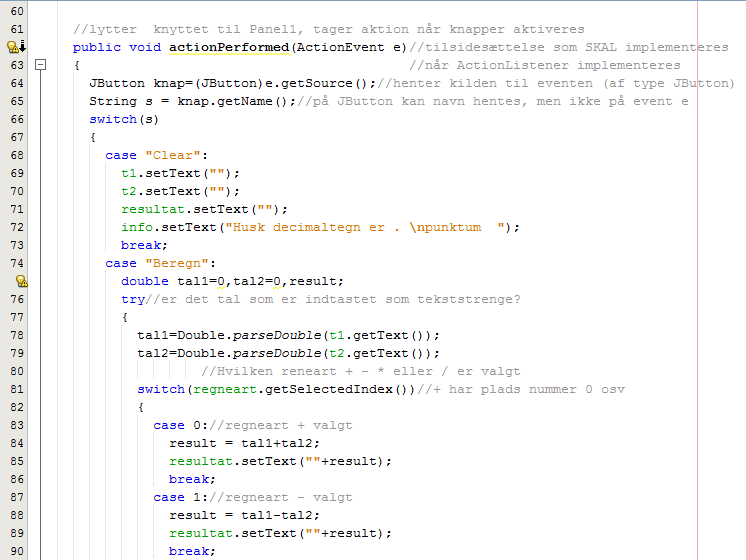
Her følger koden for vinduet(her startes programmet)

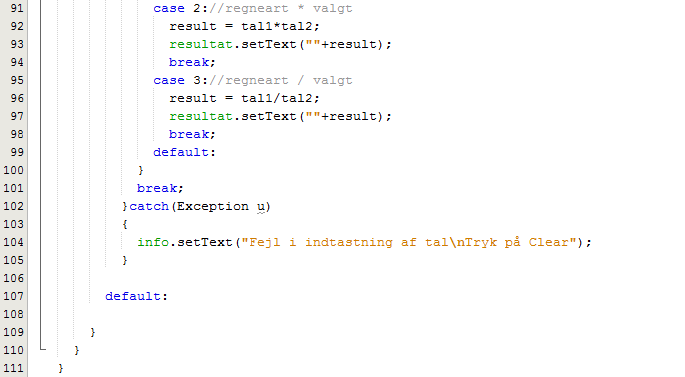


og her følger koden for panelet.









I linje 8 nedarver Panel1 klassen fra JPanel(dvs. Panel1 er et JPanel) og samtidig implementeres en ActionsListener på Panel1 klassen. Dvs. Panel1 er også en ActionListener. I linje 46 beregn.addActionListener(this); knyttes lytteren(som er panelet selv, this er panel objektet) til knappen Beregn.

Tilsvarende for Clear knappen i linje 51.

Under kørslen vil afvikleren kalde metoden

public void actionPerformed(ActionEvent e)

hvis der klikkes med musen på en af knapperne. ActionEvent objekt e har en metode, getSource(), som returner objektet som eventen er sket på. I dette tilfælde et JButton objekt(Beregn eller Clear). Eksplicit typekonvertering til JButton giver adgang til metoder på JButton objekt. I dette tilfælde benyttes getName().

JButton knap=(JButton)e.getSource();//henter kilden til eventen (af type JButton)

String s = knap.getName();//på JButton kan navn hentes, men ikke på event e

Udfra navnet finder programmet ud af, om det Beregn eller Clear knappen som er aktiveret.

I linje 20 sættes Layoutet i Panelet til FlowLalayout

setLayout(new FlowLayout());//Layot manager

//FlowLayout er standard, som hvis man ikke skrev noget

Efterhånden som komponenterne er oprettet og tilpasset, tilføjes disse i den rækkefølge, som ønskes. F.eks. første komponent i linje 21

this.add(t1);//føj t1 til panelet

Bemærk at beregningen af resultatet er pakket ind i en try…catch blok.

Brugerens indtastning af tal læses som String og skal så eksplicit typekonverteres til double. F.eks. linje 78 og 79

tal1=Double.parseDouble(t1.getText());

tal2=Double.parseDouble(t2.getText());

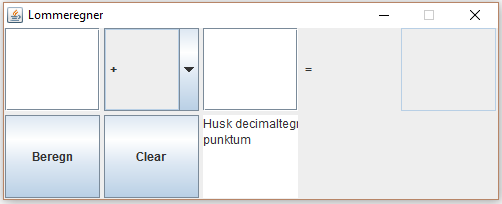
Tilsvarende konverteres reultatet fra tal til String med ””+variabel

resultat.setText(""+result);

Hvis der er noget som forhindrer beregningen, fanges fejlen, og i catch delen skrives meddelelse til brugeren i tekstareal feltet. Linje 104

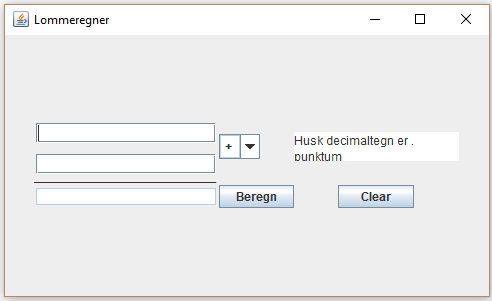
info.setText("Fejl i indtastning af tal\nTryk på Clear");

### *20.2* *Eksempel på lille lommeregner lavet med GridLayout.*



Som det fremgår, får alle komponenter tildelt samme plads, så i dette tilfælde er GridLayout ikke særligt anvendeligt. Men hvis man har en række komponenter f.eks. JButtons eller JLabels som er ens i størrelse og som skal placeres i en matrix er GridLayout den helt rigtige.

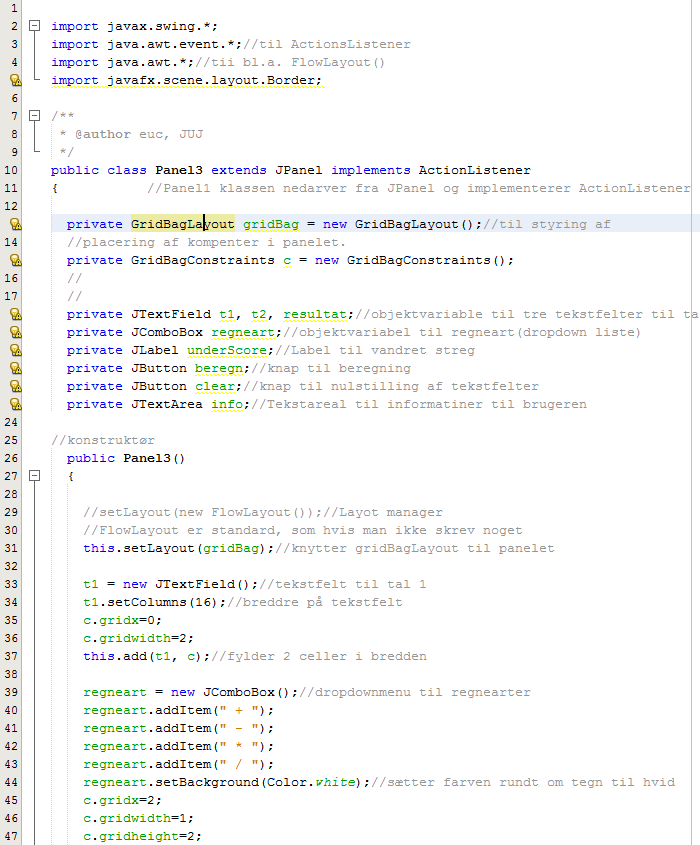
### *20.3* *Eksempel på lille lommeregner lavet med GridBagLayout.*

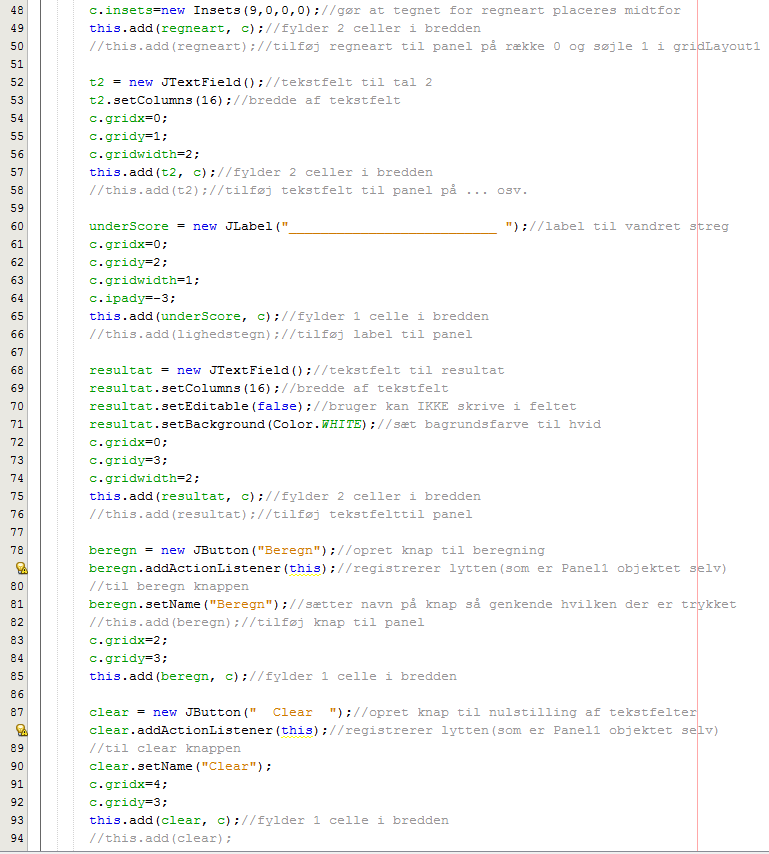


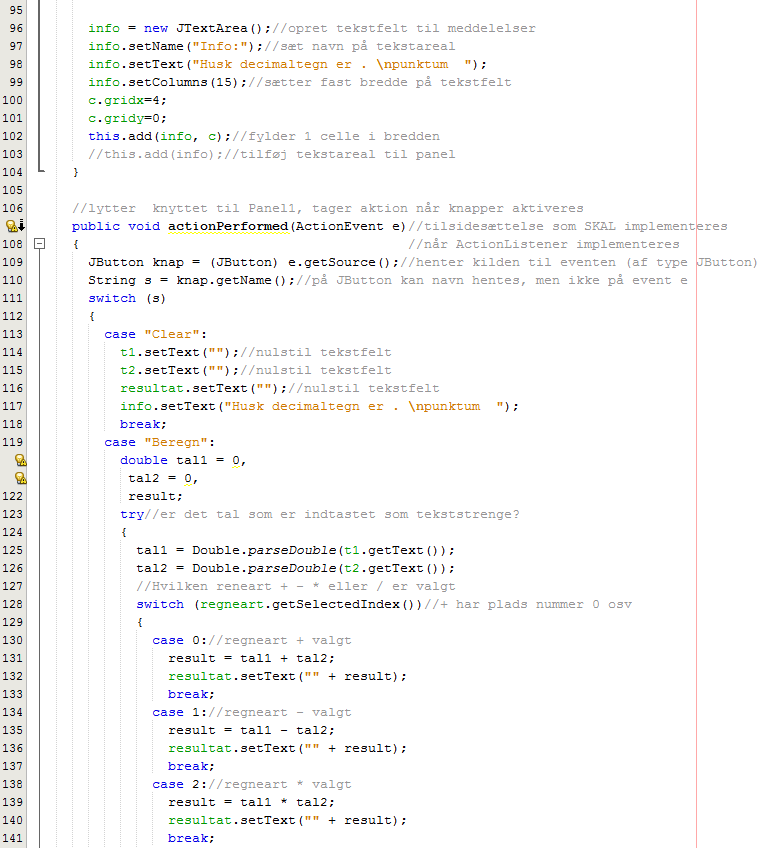
Her koden for vinduet

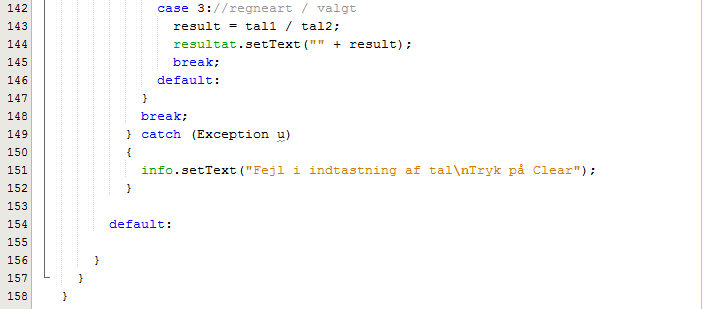


Her er koden for panelet









**20.6** ***JavaxSwing i Netbeans***

## Her er link til tutorial: <https://www.youtube.com/watch?v=6IJr5JIY9Yo>

## Leg med det, prøv det af!

## Prøv f.eks. at tilføje nye knapper der kan beregne f.eks. kvadratrod osv.

## Prøv at arbejde med Layout. Se f.eks. java doc på Oracles hjemmeside og se hvad og hvordan man kan gøre.

## God arbejdslyst Just