

Obligatorisk oppgave i OBJ2000

Objektorientert programmering 1

Høsten 2022

Vi skal bruke mye av tiden i uke 42 og 43 til obligatorisk oppgave, men du bør begynne nå.
LEVERINGSFRIST ONSDAG 2. NOVEMBER KL 23.59. EMNEOMTALEN SIER EGENTLIG AT DETTE ER
INDIVIDUELL OPPGAVE, MEN JEG VIL I ÅR ÅPNE FOR GRUPPE PÅ 2 – TO – STUDENTER.

Formelle krav

Det skal utvikles et tegneprogram i JavaFX som lar brukeren tegne enkle geometriske figurer. Bruker skal kunne velge hvilken figurtype hen ønsker å opprette, og deretter tegne valgt figurtype på en tegneflate i programmet ved hjelp av musepeker, og dermed bestemme posisjon, størrelse og form, og velge linjefarge og fyllfarge.

Hen skal også kunne selekttere en tidligere tegnet figur på skjermen og flytte den med musepeker, og forandre linjefarge og fyllfarge. Programmet skal støtte følgende figurtyper:

- Linjetyper:
 - Rett linje
 - Ellipsebue (arc) – frivillig ekstraoppgave
 - Polyline – frivillig ekstraoppgave
- Flatetyper:
 - Rektangel
 - Sirkel
 - Ellipse – frivillig ekstraoppgave
 - Polygon – frivillig ekstraoppgave
- Tekst

Når bruker selekterer et figurobjekt, skal det dukke opp et panel på høyre side med opplysninger om valgt figur: figurtype, farge, evt. linjelengde, evt areal og evt. tekst. I det panelet kan bruker endre fargene. Du finner nok ferdig komponent for å velge farge, ellers kan du lage selv.

Ved overlapp mellom figurer er det avgjørende hvilken rekkefølge de tegnes ut i. I utgangspunktet tegnes de i samme rekkefølge som de ble opprettet, men bruker skal kunne endre dette, flytte et objekt bakerst, forrest, ett trinn bakover, ett trinn framover.

Programmet behøver ikke støtte rotasjon av figurene (frivillig ekstraoppgave), så all tekst står rett veg, og rektangler er rettvendt. Linjer skal selvsagt kunne gå fra/til vilkårlig punkt.

Programmet skal tillate at det opprettes vilkårlig mange figurobjekter av vilkårlige typer.

Leveransen skal bestå av:

1. Programkode lastet opp i Canvas, evt hele prosjektkatalogen. Om du komprimerer så bruk zip, ingen ting annet.
2. Minimum to skjermbilder av applikasjonen i bruk, som del av Canvas-leveransen.
3. Demonstrasjon av applikasjonen torsdag 3. november. Alle må stille med egen PC. Dersom dette er vanskelig må du ta kontakt pr epost senest ved innlevering, for å avtale alternativt opplegg.

Tips/drøfting

Oppgaven kan løses på mange måter, men her er noen råd.

- Det er trolig hensiktsmessig å bruke et BorderLayout som hovedstruktur, der tegneflaten er i sentrum og andre GUI-komponenter «rundt omkring». Selve tegneflaten kan være et Pane, som eksemplene i boka.
- Det er lurt at man lager en hjelpemetode for hvert panel, som bygger opp og returnerer panelet. Dermed blir hver metode ganske kort og oversiktlig. Alternativt kan du lage en subklasse av relevant panel.
- For å tegne figurene på tegneflaten, er det naturlig å bruke passende «Shapes». Men det trengs noe mer, for de har ikke metoder for å beregne areal og linjelengde. DU KAN ALTSÅ IKKE BARE BRUKE SHAPE-KLASSENE. Vurdér hva du foretrekker – om dine figurklasser skal ha *referanse* til tilhørende shape, eller om dine figurklasser er *subklasser* av tilsvarende shape, eller andre løsninger, kanskje basert på grensesnitt. Det er ingen klar fasit på dette, tenk og gjør et valg.
- Det er ganske sikkert lurt å legge alle figurene i én felles struktur, gjerne ArrayList eller LinkedList.
- Legg vekt på polymorfi. Dersom programmet ditt inneholder mye kode på formen:

```
if (sirkel){
    masse kode...
}
else if (rektangel) {
    masse kode...
} osv.
```

-da har du **ikke** brukt polymorfi, da har du ikke forstått objektorientert programmering. Sørg for at alle klassene - så langt det er mulig - har de samme metodene, men med ulik "innmat".

- Hvem skal lytte til hva? Panelet må lytte etter «pressed», opprette rett objekt, og sende koordinatene til det nye objektet. Så er det nok to muligheter for hvem som skal lytte etter «dragged». Enten panelet, som hele tiden forteller det nye objektet hva brukeren gjør, eller det nye objektet selv. Helst slik at det visuelle oppdateres – man ser objektet endre seg mens man «dragger».
- Ved «selektering» av eksisterende objekt er det to muligheter: panelet eller hver enkelt figur lytter. Fordeler og ulemper ved begge løsninger.
- Skriv ryddig kode! Bruk forklarende navn på klasser, metoder og variabler! Bruk Javadoc!

Oppdatert 04.10.2022

Tor Lønnestad