

Prosjektoppgave

i PROG1003 - Objekt-orientert programmering våren 2025

Frist: *Mandag 7.april 2025 kl.11.00*

Arbeidsform: *1-3 personer*

Arbeidsinnsats: *Mye*

Innledning

Dere skal i denne prosjektoppgaven lage et litt større program som holder orden på oppdrag/boliger (dvs. leiligheter og eneboliger) til salgs i ulike soner i en by/et geografisk område, samt potensielle kunder/kjøpere som har interesse for boliger i en eller flere av disse sonene.

I hovedsak skal programmet håndtere følgende operasjoner:

- **Legge inn / skrive / slette / endre kunder/kjøpere**
- **Legge inn / skrive soner**
- **Legge inn / skrive / slette oppdrag/boliger (leiligheter/eneboliger)**
- **Skrive (til fil) oversikt over alle aktuelle boliger for en kunde**
- **Hele datastrukturen leses fra/skrives til filer**

Globale variable, klasser og datastrukturen

Programmet *skal i hvert fall* inneholde to globale objekter av klassene `Kunder` og `Soner`.

Se filene: **datastruktur.pdf** og **mainV25.cpp**

Programmet *skal minst* inneholde de seks klassene med (minst) sine datamedlemmer:

1. **Kunder:**

- `int sisteNr`: fortløpende økende automatisk nummerering av alle nye kunder
- `list <Kunde*>`: *sortert* liste med de *unikt* nummererte kundene

2. **Kunde:**

- `int`'er: kundens unike nummer (sortert på dette) og telefon
- `string`'er: navn, gateadresse (gate + nr), postadresse (nr + sted) og mail
- `enum boligtype`: med 'Leilighet' eller 'Enebolig' – etter kundens interesse
- `vector <int>`: *Sortert* vector med sonenumrene som kunde er interessert i (ene)bolig i

3. **Soner:**

- `int sisteNr`: fortløpende økende automatisk nummerering av alle nye oppdrag/boliger
- `map <int, Sone*>`: map med *alle* aktuelle *unikt nummererte* Sone'r (trenger *ikke* å være i fullstendig nummerrekkefølge)

4. **Sone:**

- `int`: sonens *unike* nummer
- `string`: en rimelig kort og generell tekstlig beskrivelse av sonen/området/bydel
- `vector <Bolig*>`: `Bolig/Enebolig`'ene til salgs i sonen (usortert)

5. **Bolig** (=leilighet):

- `int`'er: et *unikt* oppdrags-/bolignummer, dato innlagt i systemet (på formen ÅÅÅÅMMDD), byggeår, bruttoareal, antall soverom og angitt pris
- `string`'er: navnet på den interne saksbehandleren hos eiendomsfirmaet, nåværende eiers navn, gateadresse (gate+nr), postadresse (nr+sted), og en smålang beskrivelse av boligen ellers (med slikt som: innhold, standard, beliggenhet, tomten, byggemåte, overtagelse, kommunikasjon, avgifter/omkostninger, skole/barnehage, fritidstilbud, ligningsverdi)
- `enum boligtype`: med 'Leilighet' eller 'Enebolig'

6. **Enebolig** (subklasse av `Bolig`):

- `int`: tomtens areal (i kvadratmeter).
- `bool`: om tomten er selveiet eller ei (festetomt)

NB: En «leilighet» vil være et objekt av typen `Bolig`. Mens et hus/enebolig vil være av typen `Enebolig`.

Menyvalg / funksjoner

Det skal lages et fullverdig program som har følgende muligheter/menyvalg/funksjonalitet:

(**NB:** *Hvilken kode som må ligge inni i ulike klassene er det opp til dere å designe/bestemme*)

1. **K N**

Kunde Ny

Det opprettes en ny kunde, og vedkommende tildeles automatisk et nummer som er *en* høyere enn siste brukte kundenummer. Alle kundens andre data leses inn, inkludert nummeret på alle soner vedkommende initielt er interessert i.

Husk å sjekke at sonenumrene virkelig finnes!

Husk også å få lagt både kunden og sonenumrene sortert inn i hver sin datastruktur.

2. **K 1 <knr>**

Skriv alt om den *ene* kunden <kundenr>

Alle kundens data skrives ut på skjermen. ('1' er et ett-tall.)

3. **K A**

Kunder Alle-skrives

Skriv *hoveddatene* om *alle* kunder ut på skjermen. Dvs. for *hver* kunde skrives: nr, navn, boligtype, antall soner vedkommende er interessert i og nummeret på alle disse. Stans utskriften, og vent på ENTER/CR fra brukeren for hver 10.kunde.

4. **K E <knr>**

Kunde Endre <kundenr>

Alle kundens nåværende data skrives ut på skjermen, og brukeren tilbys kun å legge til nye/ta vekk eksisterende soner hos kunden. Husk å holde `vector`'en sortert!

5. **K S <knr>** Kunde Slett <kundenr>
Brukeren må bekrefte at *virkelig* ønsker å foreta denne handlingen. Om så er tilfelle slettes/fjernes kunden totalt fra datastrukturen.

6. **K O <knr>** Kunde Oversikt <kundenr>
Alle data om *alle* boliger i *alle* kundens interessesoner skrives på et *lesbart og forståelig* format til filen Kxxxxx.DTA. Der «xxxxx» er kundens unike nummer.

7. **S N <snr>** Sone Ny <sonenr>
Det opprettes (om ikke finnes allerede) og legges inn i datastrukturen en ny sone med <snr>. *Kun* dens beskrivelse leses inn, da nye oppdrag/boliger/eneboliger opprettes vha. kommandoen «O N».

8. **S 1 <snr>** Skriv *alt* om den *ene* sonen <sonenr>
Alt om *alle* oppdragene i vedkommende sone skrives ut på skjermen. Stans utskriften, og vent på ENTER/CR fra brukeren for hvert 5.oppdrag. ('1' er et ett-tall.)

9. **S A** Soner Alle-skrives
Skriv *hoveddatene* om *alle* soner ut på skjermen. Dvs. for *hver* sone skrives: nr, beskrivelse og antall oppdrag/boliger/eneboliger i sonen.

- (**S E <snr>** (Sone Endre) og **S S <snr>** (Sone Slett) - skal begge *ikke* lages)

10. **O N <snr>** Oppdrag Nytt <sonenr>
Et nytt oppdrag (*Bolig* (=leilighet) eller *Enebolig* legges inn i datastrukturen ifm. aktuelt sonenummer (om det finnes!). Oppdraget tildeles automatisk et nummer som er *en* høyere enn siste brukte oppdragsnummer. *Absolutt alle* andre datamedlemmer til aktuelt objekt blir lest inn fra brukeren.

11. **O 1 <onr>** Skriv *alt* om det *ene* oppdraget <oppdragsnr>
Alt om det ene oppdraget (*Bolig*/*Enebolig*) skrives ut på skjermen.
NB: Det er litt jobb å finne igjen oppdraget blant alle sonene!

12. **O S <onr>** Oppdrag Slett <oppdragsnr>
Brukeren må bekrefte at *virkelig* ønsker å foreta denne handlingen. Om så er tilfelle slettes/fjernes oppdraget fra den sonen det befinner seg.
NB: Også her litt jobb med å finne ut hvor det befinner seg!

- (**O E <onr>** (Oppdrag Endre) - skal *ikke* lages)

Dere må selv finne og bestemme slikt som: rimelige verdier for **const**'er, intervallene for **int**'er (som leses inn fra brukeren), hvordan håndtere evt. **enum**'er, og hvilke funksjoner/tjenester de ulike objektene må tilby hverandre (interface).

De aller fleste feilsituasjoner, f.eks. ulovlige kommandoer, ikke-eksisterende navn/numre, tomme containere, ...m.m, og dertil egnede meldinger, er stort sett ikke bemerket ovenfor. Dette må også selvsagt gjøres/kodes.

Forslag til **mainV25.cpp** kan hentes ned fra prosjektets hjemmeside. Det må i tillegg *minst* lages en global funksjon med utskrift av lovlige kommandoer.

*Tips: Kommandoene **KE**, **KS**, **KO** og **OS** bør være noe av det siste dere lager/koder, når alt annet virker og er ferdig laget/testet. For koden for disse er det mest utfordrende.*

Data til/fra filer

I programmet er det totalt involvert tre ulike (typer) filer:

KUNDER.DTA og **SONER.DTA**

Hele datastrukturen inni henholdsvis både Kunder og Soner ligger lagret på disse.

Formatene: Dette må dere selv bestemme

Kxxxxx.DTA

Format og utseende slik at blir leselig/forståelig for brukeren: Også selvbestemt ...

Prosjekt / multifil-program

Dere *skal* utvikle hele dette programmet som et prosjekt, der programmet er splittet opp i mange ulike filer. Følgende (minst 10) .h-filer må lages:

- en med *alle* const'er
- en med *alle* enum'er
- en med deklarasjon av *alle* 'globale' funksjonsheadinger
- en *pr.klasse* med deklarasjon av dets innhold (datamedlemmer og funksjonsheadinger)
- LesData3.h (ligger ferdig på katalogen «EKSEMPLE»)

Følgende (minst ni) .cpp-filer må lages:

- en som inneholder main og definisjon av de globale variablene
- *minst* en fil som inneholder definisjon (innmaten) av *alle* de 'globale' funksjonene
- en *pr.klasse* med definisjon av klassens funksjoner (deres innmat)
- LesData3.cpp (ligger ferdig på katalogen «EKSEMPLE»)

Hjelp: Se og lær av filene EKS27*.* på EKSEMPLE-katalog.

Annet (klargjørende?)

- Ordet «oppdrag» er i teksten brukt som en felles betegnelse for et eiendomssalg. I praksis er dette i koden enten en Bolig eller en Enebolig. En «leilighet» er en Bolig.
- For at datamengden som skal håndteres/lagres i dette programmet ikke skal «gå over alle støvleskaft», så tar vi ikke hensyn til bl.a: budgivning, selve salgsoppgaven (med masse tekst/info, bilder o.l), takst på eiendommen, egenerklæring om eiendommens status, bruks- og gårdsnummer, nåværende eiers personalia, visningstid.
- **NB:** Programmets kun to globale objekter er definert på filen **mainV25.cpp**. Når disse trengs brukt i andre filer, så refereres de til vha. `extern` (jfr. EKS_27*.*).
- Denne oppgaveteksten er nok ikke helt entydig og utfyllende på alle punkter/måter. Derfor er det mulig at dere må gjøre deres egne klargjøringer/presiseringer/forutsetninger (se pkt.2e under «Innlevering» på neste side).

NB NB NB

1. **Behold norske navn på klassene (og deres da respektive h- og cpp-filer).**
(Det er da så mye enklere for læringsassistentene (LA) å rette mange prosjekter hver, når filer og klasser har samme navn som angitt i dette dokumentet.)
2. **La absolutt alt av h-, cpp- og andre filer ligge i en og samme (topp)katalog på GitLab.**
(Dette gjør også alt rettelarbeidet for LAene mye enklere.)
3. **Lever tidlig en lenke til prosjektet i Blackboard** (se pkt.1 under «Innlevering» nedenfor).
4. **Testkjør prosjektet i god tid før fristen ved å clone det hele ned til en helt ny katalog, og kjør det derfra.** (Blir en simulering av hvordan LAene vil kjøre/oppleve programmet.)
5. **Meld fra (via mail) til emnelærer om grupper består av bare Mac- eller Linux-brukere.**

Innlevering

Innen mandag 7.april 2025 kl.11:00 skal dere ha:

1. **levert en lenke/adresse til prosjektet på GitLab via Blackboard.**
Dette gjøres ved å kopiere/klippe ut SSH-adressen for cloning i GitLab.
Den limes inn i Blackboard via «Skriveinnsending»
(og ikke via «Legg ved filer» - som hittil ved oblig-innleveringer).
2. **lastet opp (committed) på GitLab:**
 - a. deres fungerende, endelige og siste versjon av koden i prosjektet
 - b. lagt inn *minst* de obligatoriske testdataene i alle filene
(jfr. pkt.7 på websiden om prosjektet)
 - c. de tre stk. ferdig utfylte testskjemaene (pdf) for KN, ON og OS
 - d. *en* beskrivelse (pdf) av DTA-filenes format og eksempler på utseende
 - e. evt. *en* egen fil (pdf) med egne presiseringer/forutsetninger

Gruppe(sam)arbeid

- Sørg for at alle ytre rammer er lagt til rette for et godt og konstruktivt samarbeide.
Dette gjøres bl.a. best ved å sette opp klare og konkrete grupperegler.
- *Jobb mye, effektivt og målrettet allerede fra første stund* (dvs. start «langspurten» straks).
- *Gjør et selvstendig gruppearbeide. Dvs. ikke samarbeid med/kopier fra andre grupper.*
Det kalles plagiat, og vil medføre stryk på prosjektet for de involverte gruppene.
- Dere velger selv antallet (1-3 stk) i gruppen. Men, uavhengig av gruppeantallet, så forventes det at dere kommer i mål med prosjektet (*alt* kodes/gjøres og virker).
- Et gruppearbeid er et gruppearbeid. Enten får *alle* godkjent, eller så får *alle* det ikke.
- Det er *ingen reinnlevering på prosjektet* («second chance»). Enten så holder det man har kodet/laget, eller så gjør det ikke det.

Løkke tæll!

FrodeH