Prosjektoppgave

i "IMT1082 - Objekt-orientert programmering" våren 2014

Frister: Tirsdag 22.april 2014 kl.10.00

NB: Fredag 21.mars kl.09.00 (1.delinnlevering)
Tirsdag 8.april kl.09.00 (2.delinnlevering)

Arbeidsform: Gruppe (tre (evt. to) personer – flere er ikke relevant)

Arbeidsinnsats: Mye

Innledning

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

I hovedsak skal programmet håndtere følgende operasjoner:

- Legge inn / slette / endre / skrive en eiendom (bolig/tomt)
- Legge inn / slette / endre / skrive en potensiell kunde/kjøper
- Skrive alle eiendommer i en gitt sone
- Lage filer med navn/adresse på kunder som snarlig skal ha tilsendt salgsoppgaver
- Lage (ukentlig) liste over *alle* aktuelle eiendommer for en kunde

Globale variable, klasser og datastrukturen

Programmet trenger kun å inneholde to globale objekter av klassene Soner og Kunder.

Programmet *skal* (i hvert fall) inneholde de syv klassene med (minst) sine datamedlemmer:

- 1. **Kunder** inneholder:
 - int: nummeret til første nåværende kunde i datastrukturen
 - int: nummeret til sist innlagte kunde (*ikke* nødvendigvis lik den *siste* nåværende i datastrukturen, men større om de(n) bakerste/siste er slettet)
 - Datastrukturen med alle Kunde'ne
- 2. **Kunde** inneholder:
 - int'er: kundens unike nummer (sortert på dette) og telefon
 - char-pekere: navn, gateadresse (gate+nr), postadresse (nr+sted) og mail
 - Datastruktur med info om *alle* sonene (IntrSone) kunden er interessert i eiendom i (*ikke* bruk en 100 lang array).

3. **IntrSone** – inneholder:

- int'er: et sonenummer (sortert på dette), max.pris (kunden er villig til å betale for en eiendom i denne sonen), min.areal (boareal *eller* tomteareal) og min.antall soverom (kunden krever for en bolig i denne sonen)
- enum'er:
 - tilsending av boliginfo: ukentlig hovedoversikt eller info (salgsoppgave) om hver nye eiendom i sonen umiddelbart/snarest
 - interessert i om: til salgs, leie eller begge deler
 - eiendomstype: tomt, enebolig, rekkehus, leilighet eller hytte

4. **Soner** – inneholder:

- int: nummeret til sist innlagte eiendom/oppdrag i datastrukturen
- Datastruktur med max.100 Sone'r

5. **Sone** – inneholder:

- char-peker: rimelig kort og generell tekstlig beskrivelse av sonen
- Datastrukturen med Eiendom'ene til salgs/leie i sonen

6. **Eiendom** – inneholder:

- int'er: et unikt eiendoms-/oppdragsnummer (sortert på dette), dato innlagt i systemet (på formen ÅÅÅÅMMDD), bruksnummer (fire siffer), nummeret for den interne saksbehandleren hos eiendomsfirmaet, pris for boligen/tomten og tomtens areal (i kvadratmeter).
- char-pekere: gateadresse (gate+nr), postadresse (nr+sted), nåværende eiers navn, kommunenavnet der eiendommen ligger og *en* smålang beskrivelse av eiendommen ellers (med slikt som: innhold, standard, beliggenhet, tomten, byggemåte, overtagelse, kommunikasjon, avgifter/omkostninger, skole/barnehage, fritidstilbud, ligningsverdi)
- enum: "eiendomstype" (se enum'ene ovenfor)

7. **Bolig** – er en subklasse av Eiendom og inneholder:

- int'er: byggeår, bruttoareal, boareal og antall soverom
- enum: "interessert i om" (se enum'ene ovenfor)

NB: En "tomt" vil være et objekt av typen Eiendom. Mens de ulike andre eiendomstypene (se enum'ene ovenfor) vil være av typen Bolig.

Tips: Tegn opp, og bli ordentlig sikker på hvordan datastrukturen er/ser ut, og hvordan den fungerer ifm. de ulike funksjonene dere skal lage (angitt i "Innledning" og "Menyvalg/funksjoner"). Se også avsnittet "Delinnlevering nr.1" (nest siste side).

Menyvalg / funksjoner

Dere skal lage et fullverdig program som har følgende muligheter/menyvalg:

1. **E D <oppdnr> | <pnr>** Eiendom Display <oppdragsnr> eller <postnr> Skriver ut (på skjermen) eiendommen/oppdrag med et gitt oppdragsnummer eller *alle* eiendommene/oppdragene med et gitt postnummer i postadressen sin.

2. **E** N **<sonenr>** Eiendom Ny **<**sonenr>

Legger inn en ny eiendom/oppdrag i en gitt sone. Om sonen ikke finnes fra før, så opprettes denne og dets generelle beskrivelse leses inn. Uansett leses *alle* data om en ny eiendom/oppdrag (brukeren må velge om dette er en Eiendom (dvs. tomt) eller en Bolig) og denne legges så inn i datastrukturen. Oppdragets nummer telles automatisk opp med +1, og det nye nummeret skrives til den *første* linjen på filen 'SISTE.DTA'. *Alt* om den aktuelle sonen skrives til filen 'SONExxx.DTA'.

Det gås så gjennom *alle* kundene. De kundene:

- som er interessert i denne sonen
- og som har sine krav/kriterier tilfredsstilt/matched (eiendomstype, til salgs/leie, pris, tomte-/boareal og antall soverom)
- og som umiddelbart/snarest ønsker tilsendt salgsoppgave får sine navn, gate- og postadresser fortløpende skrevet/tillagt bakerst på filen 'Exxxxxx.DTA'.
- 3. **E S <oppdnr>** Eiendom Slett <oppdragsnr> Eiendommen/oppdraget med gitt nummer slettes (etter at brukeren har bekreftet dette) fra datastrukturen. Aktuell sone skrives til filen 'SONExxx.DTA'.
- 4. **E E <oppdnr>** Eiendom Endre <oppdragsnr>
 Brukeren gis mulighet til å endre på alle relevante verdier for en gitt eiendom/oppdrag.
 Aktuell sone skrives nå også til filen 'SONExxx.DTA'.
- 5. **S D <sonenr>** Sone Display <sonenr> Skriver ut (på skjermen) data om *alle* eiendommene/oppdragene i en gitt sone.
- 6. **K D <knr> | <knavn>** Kunde Display <kundenr> eller <kundenavn> Skriver ut alle data om en gitt kunde (nr) eller om *alle* kundene som har et gitt navn. Dette inkluderer også de dataene som er lagt inn angående kundens "føringer" for de sonene (IntrSone) vedkommende er interessert i.

7. **K** N Kunde Ny

Det opprettes en ny kunde, og vedkommende tildeles automatisk et nummer som er *en* høyere enn siste brukte kundenummer. Dette tallet skrives til den *siste* linjen på filen 'SISTE.DTA'. Alle kundens andre data leses inn, inkludert data om alle soner vedkommende er interessert i. De dataene som skrives inn om den første sonen lar vi automatisk bli duplisert for *alle* de andre sonene også. Om man *egentlig* ønsker dette litt annerledes for noen av sonene, så brukes kommandoen K E senere for dette. *Alle* data om vedkommende kunde (inkl. sone-info) skrives til filen 'Kxxxxxxx.DTA'. For *alle* de sonene og deres eiendommer/oppdrag kunden umiddelbart/snarest vil ha tilsendt info om, og som tilfredsstiller/matcher vedkommendes krav/kriterier, så skrives hans/hennes navn og adresse bakerst på filen 'Exxxxxxx.DTA'.

8. **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, og begge filene om kunden ('Kxxxxxxx.DTA' og 'Kxxxxxxx.INF') blir slettet. *Om* det var kunden med det laveste kundenummeret som ble slettet, oppdateres den aktuelle variabelen med den nye førstekundens nummer og dette tallet skrives til den *midterste* linjen på filen 'SISTE.DTA'.

9. **K** E **<knr>** Kunde Endre **<**kundenr>

Brukeren tilbys å endre brukerens data, inkludert de om brukerens interesse for ulike soner. *Alle* data om vedkommende kunde skrives nå også til filen 'Kxxxxxxx.DTA'. For de sonene kunden skifter *til* umiddelbar/snarest boliginfo om, vil det også være aktuelt å legge til info om kunden bakerst på filene 'Exxxxxxx.DTA'.

10. U Ukentlig utskrift/status/sending

En eller annen fast dag(?) i løpet av uken utføres denne kommandoen. Det gås da gjennom *alle* kundene. For hver kunde lages filen: 'Kxxxxxxx.INF'. På denne filen blir det skrevet *alle* kundens hoveddata (nr, navn, adr, mail og tlf) og *hoveddataene* om *alle* de eiendommene/oppdragene som er aktuelle (tilfredsstiller kravene/kriteriene) for kunden.

Alle slags feilsituasjoner (f.eks. ulovlige kommandoer, ikke-eksisterende kundenavn / kunde-, sone-/ post-/oppdragsnumre, ingen eiendommer/oppdrag i en sone, **m.m.**), og dertil egnede meldinger, er det ikke bemerket ovenfor. Dette må også selvsagt gjøres/kodes.

I tillegg må selvsagt main lages (som "styrer hele butikken"), samt funksjoner for å lese brukerens valg/kommando og en lengre utskrift med liste over lovlige valg/kommandoer. Og en del annet også (f.eks. for å lese: *numeriske* verdier (også i et visst intervall), at valg/tegn er bare *ett* av flere lovlige).

Data til/fra filer

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

1. SISTE.DTA

Format: <siste brukte oppdragsnummer> (opptil syv sifre)

<første nåværende brukte kundenummer> (opptil syv sifre)

<totalt siste kundenummer som har blitt brukt> (opptil syv sifre)

Dette er altså verdiene til enkelt-variablene som ligger inni klassene Kunder og Soner. De blir oppdatert flere ulike steder i programmet (se ovenfor) ved *direkteposisjonering* på filen (husk at *alle* linjene er ni lange (i Windows) inkl. '\n').

2. SONExxx.DTA

Format: Dette må dere selv bestemme

xxx = 001-100 **NB:** En slik fil pr.sone.

3. Kxxxxxxx.DTA

Format: Dette må dere selv bestemme

xxxxxxx = 0001000 - 9999999 **NB:** *En* slik fil pr.kunde.

4. Exxxxxxx.DTA

5. KxxxxxxxxINF

Prosjekt / multifil-program

Dere *skal* utvikle hele dette programmet som et prosjekt, der programmet er splittet opp i flere ulike filer. Følgende (minst 12) .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)
- listtool2.h (ligger allerede ferdig under "Filer" "PROSJEKT")
- timer3.h (ligger allerede ferdig under "Filer" "PROSJEKT")

Følgende (minst 11) .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)
- listtool2.cpp (ligger allerede ferdig under "Filer" "PROSJEKT")
- timer3.cpp (ligger allerede ferdig under "Filer" "PROSJEKT")

Hjelp: Se og lær av filene E19*.* på EKSEMPEL-katalog.

Annet (klargjørende?)

- LISTTOOL skal brukes ifm. løsningen av denne prosjektoppgaven. Legg merke til og bruk 'LISTTOOL2.H' og 'LISTTOOL2.CPP' som ligger på PROSJEKT-katalogen.
- Det er *ikke* meningen at dere skal trenge å bruke noe fra STL-biblioteket/string-klassen.
- **NB:** Definer globale variable (av klassene Soner og Kunder) på samme fil som main. Når dere trenger å bruke disse på/i andre filer, så refererer dere til dem vha. extern
- Noen aktuelle const'er kan være: MAXSONE, NVNLEN og STRLEN
- Ordet 'eiendom' kan ovenfor kanskje ha blitt brukt litt tvetydig. 'Eiendom' kan både være det samme som '(salgs)oppdrag' og selve klassen Eiendom (som i praksis er en Bolig eller en tomt). Forhåpentlig kommer det fram av konteksten hva som menes.

- Numrene som brukes er: Sonenr: 1-100, Kundenr: 1000 →, Postnr: 100-9999 og Oppdragsnummer: 10000 →
- Det *skal* sikres at alle tall som leses inn virkelig er *numeriske*, samt at de ligger i *fornuftige* intervaller.
- **Hint:** Lag funksjonen
 - void lagNavn(char*t, char* s1, char* s2, int n, const int LEN) Virkning: 't' blir inneholdende 's1' etterfulgt av tallet 'n' som *alltid* opptar 'LEN' plasser (blir evt. fylt på med innledende 0'er) samt avsluttes med 's2'. Denne funksjonen brukes *alle* ganger det er behov for å konstruere et av filnavnene (med x'er i) beskrevet ovenfor.
- Filene 'TIMER3.H' og 'TIMER3.CPP' (på PROSJEKT-katalogen) *kan* inneholde funksjoner dere har bruk for (Dataene på disse filene er uinteressante. Det aktuelle er å lage et Timer-objekt (når ønskelig), og tilkalle en av dets tilgjengelige funksjoner.)
- Kommandoene 'K S' og 'K E' har *kun* <knr> som parameter (*ikke* <knavn>). ('K D' tar begge.) Dette skyldes at det blir litt uklart ("dramatisk") å skulle kunne angi et et <knavn> som flere kunder kan ha. (En aktuell kundes <knr> får man jo vite ved å bruke kommandoen 'K D <knavn>' på vedkommende.)
- Poenget med kommandoen 'U' er at kunden ukentlig vil få tilsendt en oversikt over alle relevante eiendommer, og dermed være sikret at vedkommende er oppdatert om hvilke eiendommer som for øyeblikket er til salgs. Kan så via SMS/mail/telefon be om å få tilsendt relevante salgsoppgaver som vedkommende er ekstra interessert i.
- For at datamengden som skal håndteres/lagres i denne prosjektoppgaven ikke skal "gå over alle støvleskaft", så tar vi ikke hensyn til bl.a. slikt som: selve salgsoppgaven (med *masse* tekst/info, bilder o.l), takst på eiendommen, egenerklæring om eiendommens status, gårdsnummer, nåværende eiers personalia, visningstid, lånetakst, ligningtakst.
- 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. Angi dette i så fall på et ark først i besvarelsen deres.

Forslag til rekkefølge på implementasjon

- 1. Main m/lesKommando, skrivMeny og omrisset av klassene m/datamedlemmer.
- 2. Lag funksjonen lagNavn(...).
- 3. Bestem formatet for 'SONExxx.DTA' og 'Kxxxxxxx.DTA'. Lag 2-3 "dummy" filer av hver av disse, samt 'SISTE.DTA'.
- 4. Les fra/skriv til 'SONExxx.DTA' og 'SISTE.DTA' (oppdragsnr).
- 5. Les fra/skriv til 'Kxxxxxxx.DTA' og 'SISTE.DTA' (forstenr og sistenr).
- 6. Implementer kommandoene 'S D', 'E D' og 'K D'.
- 7. Implementer kommandoene 'E N' (vent med data til filen 'Exxxxxxx.DTA') og 'E S'.
- 8. Implementer kommandoene 'K N' (vent med data til filen 'Exxxxxxx.DTA') og 'K S'.
- 9. Lag skrivingen til filen 'Exxxxxx.DTA' ved kommandoene 'E N' og 'K N'.
- 10. Implementer kommandoene 'U'.
- 11. Implementer kommandoene 'E E' og 'K E'.

Innspill til gjøremål 1.arbeidsuka (17.-21.mars)

- 1. Sette seg inn i/lese nøye oppgaveteksten. Analyse av problemstillingen og datastrukturen.
- 2. Delta på de tre forelesningene i uke 12 (mandag, tirsdag og fredag).
- 3. Lage grupperegler (dokumentet som ligger ute er *innspill*, ingen *ferdig* kontrakt).
- 4. Bestemme hvordan konfigurasjonsstyring/versjonskontroll skal foregå. (og da evt. installere og bli kjent med Git).
- 5. Sette datoer/tidsfrister for pkt.1-11 nederst forrige side.
- 6. Være godt i gang med design/pseudokode for relevant kode.
- 7. Kode/implementer pkt.1og 2 nederst forrige side.
- 8. Overholde fristen for og innholdet i "Delinnlevering nr.1" (se rett under).

Delinnlevering nr.1

Innen fredag 21.mars 2014 kl.09.00 skal dere levere følgende (på papir) til emnelærer:

- Ett A4- eller A3-ark med detaljert tegning av datastrukturen
- Navnet på gruppedeltagerne, inkl. kontaktinfo (navn, mail, mobil) for alle i gruppen
- Gruppereglene, signert av alle gruppens medlemmer. Se *innspill* til kontrakt på: http://www.hig.no/~ooprog/obliger/regler.pdf
- Individuelt signert bekreftelse fra *hver* av deltagerne, se: http://www.hig.no/~ooprog/obliger/bekreftelse.pdf

Delinnlevering nr.2

Innen tirsdag 8.april 2014 kl.09.00 *skal* dere sende (på mail til emnelærer) *en* zip-fil innholdende *alle* prosjektets (minst) 23 .h- og .cpp-filer, relevante .DTA-filer og .exe-filen for programmet. Filen *skal* hete «gruppexx.zip», der «xx» er gruppenummeret deres). Dere skal da *minst* ha gjort ferdig (kodet og fungerer korrekt) t.o.m. pkt.7 under "Forslag til rekkefølge på implementasjon" (nederst forrige side). Ellers vil dere ligge dårlig an 🕾

Sluttinnlevering

Følgende *skal* leveres (**innen tirsdag 22.april 2014 kl.10.00**):

- Utskrift av programmet (*alle* .h- og .cpp-filer) i ett eksemplar.
- USB-stick (evt. CD/DVD) med *alle* filene som trengs for å kjøre programmet fra Visual Studio C++. Dvs. *alle* .dta-, .h- og .cpp-filer, *hele* katalogen og dets underkatalog "debug" der alle VS-filene ligger, samt en kjørbar (.exe-fil) versjon av programmet. **Testkjør dette selv fra der filene ligger, på en maskin/konto som** *ikke* **allerede inneholder prosjektet men Visual Studio, før dere leverer.**
- Filene som det skrives til/leses fra (.dta) sitt *format* og eksempel på deres utseende.
- Egne presiseringer/forutsetninger (om dere har måttet foreta dette).

Gruppe(sam)arbeid

Sørg for at alle ytre rammer er lagt til rette for et godt og konstruktivt samarbeide. Dette gjøres best ved å sette opp klare og konkrete grupperegler (se eget notat med *innspill* til dette). *Jobb mye, effektivt og målrettet allerede fra første stund* (dvs. start "langspurten" tidlig, se "Innspill til gjøremål 1.arbeidsuka").

Og: sørg for å være "i rute" ved delinnlevering nr.2 – ellers får dere en knallhard avslutning.

Generelle krav til obligatoriske arbeider

Se: http://www.hig.no/~grprog/obliger#Gen_reg (åtte punkter nederst på siden)

Lykke til!

FrodeH