

# Prosjektoppgave

i «**IMT1082 - Objekt-orientert programmering**» våren 2017

**Frister:** **Fredag 7.april 2017 kl.12:00**

(Gi emnelærer klar beskjed om dere trenger ekstra frist til over påske - dvs. tirsdag 18.april kl.09:00.)

**NB:** Fredag 17.mars kl.10:00 (1.delinnlevering)

Onsdag 29.mars 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 noe større program som holder orden på ulike nasjoner, deltagere, grener, øvelser, deltagerlister, resultatlister, medaljestatistikk og poengstatistikk ifm. et større idrettsarrangement (f.eks. vinter-OL).

## Globale variable, klasser (og litt datastrukturen)

Programmet skal *kun* inneholde fem globale objekter av klassene: Nasjoner, Deltagere, Grener, Medaljer og Poeng. Inni disse objektene ligger til enhver tid den aktuelle datastrukturen.

Programmet skal (i hvert fall) inneholde de ti klassene med minst datamedlemmene:

1. **Nasjoner** – inneholder datastrukturen med Nasjon'er.
2. **Nasjon** – inneholder en unik nasjonsforkortelse (tre tegn, sortert på dette), fullt/reelt navn, antall deltagere i nasjonens tropp, navn og telefon til nasjonens kontaktperson, samt andre generelle data om nasjonen.
3. **Deltagere** – inneholder datastrukturen med Deltager'ne.
4. **Deltager** – inneholder et unikt nummer (sortert på dette), fullt/reelt navn, nasjonstilhørighet, kjønn, samt andre generelle data om deltageren.
5. **Gren** – inneholder datastrukturen med Gren'ene.
6. **Gren** – inneholder et unikt navn på grenen (sortert på dette), hvordan prestasjoner i grenen generelt registreres (tid eller poeng), samt alle øvelsene i grenen (max 20 stk).
7. **Øvelse** – inneholder en unik ID (nummer eller navn), fullt/reelt navn, klokkeslett, dato, antall deltagere, deltagerliste og resultatliste.
8. **Statistikk** – inneholder opptil 200 nasjonsforkortelser.
9. **Medaljer** – er avledet klasse fra **Statistikk**. Den inneholder i tillegg en sortert oversikt over hvor mange medaljer (gull, sølv og bronse) de ulike nasjonene har fått.
10. **Poeng** – er også avledet klasse fra **Statistikk**. Den inneholder i tillegg en sortert oversikt over hvor mange poenger de ulike nasjonene har fått.

**NB:** Tegn opp, og bli ordentlig sikker på hvordan datastrukturen må være (ser ut), og hvordan den fungerer ifm. de ulike funksjonene dere skal lage (angitt nedenfor).

# Menyvalg / funksjoner

Programmet skal (i hvert fall) håndtere følgende 27 menyvalg:

|   |   |  |
|---|---|--|
| N | N | <b>Registrer en ny nasjon</b><br><i>Alle dataene om en ny og unik nasjon leses inn.</i>  |
| N | E | <b>Endre en nasjon</b><br>Alle data om en nasjon (unntatt dets forkortelse og antallet i troppen) kan endres.  |
| N | A | <b>Skriv hoveddataene om <i>alle</i> nasjoner</b><br>Forkortelsen, hele navnet og antallet i troppen skrives ut for <i>alle</i> nasjonene.   |
| N | T | <b>Skriv en nasjons deltagertropp</b><br>Nummer, navn og kjønn om <i>alle</i> deltagerne i aktuell nasjons tropp skrives ut.   |
| N | S | <b>Skriv <i>alle</i> data om en gitt nasjon</b>  |
| D | N | <b>Registrer en ny deltager</b><br><i>Alle dataene om en ny og unik deltager leses inn.</i><br>(Det må selvsagt sjekkes at nasjonen virkelig eksisterer.)  |
| D | E | <b>Endre en deltager</b><br>Alle data om en deltager (unntatt vedkommendes nummer) kan endres.   |
| D | A | <b>Skriv hoveddataene om <i>alle</i> deltagere</b><br>Nummer, navn og kjønn om <i>alle</i> deltagerne skrives ut.  |
| D | S | <b>Skriv <i>alle</i> data om en gitt deltager</b><br>Brukeren kan velge å angi deltageren vha. vedkommendes unike nummer eller vha. (en entydig del av) navnet.  |
| G | N | <b>Registrer en ny gren</b><br><i>Alle dataene om en ny og unik gren leses inn.</i> (Det skal <i>ikke</i> leses inn noen øvelser her – det gjøres vha. kommandoen O N )                                      |
| G | E | <b>Endre en gren</b><br>Kun grenens unike navn tilbys å bli endret.  |
| G | A | <b>Skriv hoveddataene om <i>alle</i> grener</b><br>Navn, registreringen av prestasjoner (tid/poeng) og antall øvelser om <i>alle</i> grenene skrives ut.   |
| G | S | <b>Skriv <i>alle</i> data om en gitt gren</b><br>Grenens navn, registreringen av prestasjoner (tid/poeng), samt nummer, fullt navn, dato, klokkeslett og antall deltagere i <i>alle</i> øvelsene skrives ut. |

Ifm. *alle* valgene under 'O', så spørres det *alltid* først om aktuell gren ( <navn> ):

- O <navn> N Registrer en ny øvelse**  
Leser inn (om plass og *ikke finnes allerede under aktuell gren*) alle data om en ny øvelse. (Leser *ikke* inn deltager-/startliste og resultatliste. Dette utføres vha. kommandoene "L N" og "R N".)
- E Endre en øvelse**  
Navn, dato og klokkeslett kan endres (ikke alt det andre).
- F Fjerne/slette en øvelse**  
Ønsket øvelse fjernes helt.  
**NB:** Dette er litt tricky, for: Husk å fjerne involverte filer, samt å oppdateres medalje- og poengstatistikken.
- A Skriv hoveddataene om alle øvelser**  
Det samme som ved kommandoen "G S" skrives.

Ifm. *alle* valgene under 'L' og 'R', så spørres det *alltid* først om aktuell øvelse ( <id> ):

- L <id> S Skriv deltager-/startliste**  
*Om deltagerliste finnes, så leses denne inn fra fil, og skrives på skjermen (startnummer, samt deltagerens nummer, navn og nasjon).*
- N Ny deltager-/startliste**  
*Om deltagerliste ikke finnes, så leses en ny inn fra tastaturet, og skrives deretter til fil. Husk å sjekke at deltagerne virkelig finnes.*
- E Endre deltager-/startliste**  
*Om resultatliste ikke allerede finnes, så tilbys brukeren å endre på deltager-/startlisten. Den blir først lest inn fra fil, og skrevet tilbake igjen etterpå.*
- F Fjerne/slette deltager-/startliste**  
*Om resultatliste ikke allerede finnes, så tilbys brukeren å slette/fjerne hele deltager-/startlisten (altså dets fil).*
- R <id> S Skriv resultatliste**  
*Om resultatliste finnes, så leses denne inn fra fil, og skrives på skjermen (startnummer, samt deltagerens nummer, navn og nasjon).*  
**NB:** *Om listen som leses fra fil ikke er sortert, så gjøres dette først, og skrives så til fil igjen. Statistikken for medaljer og poeng oppdateres da også.*
- N Ny resultatliste**  
*Om resultatliste ikke finnes, så leses den fra tastaturet (ut fra deltagerlista), den sorteres, skrives til skjermen, skrives til fil og statistikkene oppdateres. Hvordan håndtere deltagere som har brutt, ikke møtt, blitt disket?*
- F Fjerne/slette resultatliste**  
*Om denne (filen) finnes så slettes den. Men, husk først å oppdatere statistikkene.*

|          |   |
|----------|---|
| <b>M</b> | <b>Medaljeoversikt</b><br>Skriver hele medaljeoversikten/-statistikken på skjermen. |
| <b>P</b> | <b>Poengoversikt</b><br>Skriver hele poengoversikten/-statistikken på skjermen.     |
| <b>X</b> | <b>eXit / avslutt</b><br>Avslutt hele programmet.                                   |

Når brukeren har skrevet hovedkommandoen 'N', 'D', 'G' eller 'O', så må vedkommende gjerne tilbys en undermeny med de neste valgene/mulighetene.

*Alle slags feilsituasjoner (f.eks. ulovlige kommandoer, ikke-eksisterende navn og numre), og dertil egnede meldinger, er bare litt 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 *aktuelle* valg/kommandoer. En del annet også, f.eks: at tall virkelig *er numeriske* verdier, at datoer, klokkeslett, poeng og tider er i lovlige og aktuelle intervaller, bygge filnavn (deltager- og resultat-filer) eller å ”strippe” alle tekster for blanke foran og bak.

## Data til/fra filer

I programmet er det involvert syv ulike (typer) filer (*alle* filformat bestemmer dere selv):

Alle dem fem globale objektene lagrer sine data på hver sine filer: **NASJONER.DTA**, **DELTAGERE.DTA**, **GRENER.DTA**, **MEDALJER.DTA** og **POENG.DTA**.

I tillegg lagres *alle* deltager-/startlister (*en* pr. øvelse) på filene **OVxxxx.STA** og øvelsenes resultatlister (*en* pr. øvelse) på filene **OVxxxx.RES**

I *alle* tilfeller bør det automatisk skrives til fil hver gang en datastruktur er oppdatert.

## Prosjekt / multifil-program

Dere *skal* utvikle hele dette programmet som et prosjekt, der programmet er splittet opp i flere ulike filer. Følgende (minst 13) .h-filer må lages:

- en med *alle* const'er (og evt. en med *alle* enum'er)
- en med deklarasjon av *alle* 'globale' funksjonsheadinger
- en *pr.klasse* med deklarasjon av dets innhold (datamedlemmer og funksjonsheadinger)
- ListTool2B.h (ligger allerede ferdig på PROSJEKT-katalogen)

Følgende (minst 13) .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)
- ListTool2B.cpp (ligger allerede ferdig på PROSJEKT-katalogen)

**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 'ListTool2B.H' og 'ListTool2B.CPP'* (se rett ovenfor).
- Definer de fem globale objektene på samme fil som main. Når dere trenger å bruke disse på/i andre filer, så refererer dere til dem vha. `extern .....` i disse filene.
- Noen aktuelle `const`'er *kan* være: `NVNLEN`, `STRLEN`, `NASJONLEN`, `MAXNASJONER`, `MAXOVELSER`, `MINDELTAGERE`, `MAXDELTAGERE`, .....
- Medalje- og poengstatistikken inneholder *ikke* alle registrerte nasjoner, men *kun* de som har tatt medaljer/fått poeng.
- Poeng gis til de seks første i en øvelse: Gull: 7 poeng, så 5, 4, 3, 2 og 1 til de fem neste.
- Prestasjonene registreres som en av fem: Min:Sek:Tidel (skiskyting, langrenn)  
Min:Sek:Hundredel (skøyter, alpin, bob, skeleton) Min:Sek:Tusendel (aking, kortbane)  
Poeng.x (hopp) Poeng.xx (kunstløp). Alt dette må dere ta hensyn til ved håndtering av resultater.  
Kan være lurt å lagre f.eks. det første og andre som: `MMSST` og `MMSSHH`
- Det er *kun* deltager- og/eller resultatlisten for den aktuelle øvelsen man for tiden ser på/behandler, som er innlest i memory. Alle de andre ligger "bare" på fil.
- Programmet skal f.eks. *ikke* håndtere slikt som (noe av det er selvsagt en svakhet):
  - Fjerne nasjon, deltager eller gren.
  - Endre en resultatliste.
  - Grener som består av øvelser med ulike typer registreringer (f.eks. kombinert, snøbrett eller fristil – alle med poeng og tid) eller seriespill (f.eks. ishockey eller curling).
  - Fortløpende resultatservice (dvs. oppdatering av resultater og lister *underveis* under arrangementet. Det er *kun* sluttresultatet som skal oppgis, når alle er ferdig/i mål).
  - Skrive *alle* en deltagers plasseringer/medaljer.Mye av dette gir for mye "tukkel" med masse fillesing og -endring og oppdateringer av medalje- og poengstatistikkene.
- Det *skal* sikres at alle tall som leses inn virkelig er *numeriske*, samt at de ligger i *fornuftige* intervaller.
- Det er lurt å tidlig planlegge hvilken rekkefølge dere bør implementere dette i, og hva som kan gå parallelt, eller forutsetter at annet er laget/fungerer allerede.
- 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.

## Grovt forslag til rekkefølge på implementasjonen

1. Main m/lesKommando, skrivMeny og omrisset av *alle* klassene m/datamedlemmer.
  2. Bestem formatet for NASJONER.DTA, DELTAGERE.DTA og GRENER.DTA.  
Legg inn noen testdata på disse, og les dette inn i datastrukturen.
  3. Bestem flest mulig const'er, skriv pseudokode og kod (globale) hjelpefunksjoner  
(jfr. siste avsnittet *rett* over tittelen «Data til/fra filer» ovenfor).
  4. Implementer kommandoene: 'N', 'D', 'G', 'M', 'P' og 'X'  
(Hva som må gjøres i rekkefølge eller kan gjøres parallelt, må dere selv finne ut/bestemme.)
  5. Implementer (den mest avanserte) kommandoen: 'O'
- Når formatet for OVxxxx.STA, OVxxxx.RES, MEDALJER.DTA og POENG.DTA må bestemmes, og når det skal leses fra/skrives til disse, må dere selv også finne ut/bestemme.

## Gjøremål 1.arbeidsuka (13.-17.mars)

1. Gjøre pkt 1.1 og punktene 2.1-2.5 på websiden om prosjektet.
2. Sette seg inn i/lese nøye oppgaveteksten (jfr. pkt.5 på websiden om prosjektet).  
Analyse av problemstillingen og datastrukturen.
3. Delta på de tre forelesningene i uke 11 (mandag, tirsdag *og fredag*).
4. Lage grupperegler (jfr. pkt.1.2 på websiden om prosjektet).  
**NB:** Dokumentet som ligger ute er *innspill*, *ingen ferdig* kontrakt.
5. Gjøre pkt.3 og 4 på websiden om prosjektet.
6. Bestemme datoene for når de ulike punktene under «Forslag til rekkefølge på implementasjonen»  
(rett ovenfor) skal være ferdig.
7. Utføre og avkrysse alt på sjekklisten (jfr. pkt.6 på websiden om prosjektet).
8. Overholde fristen for og innholdet i «Delinnlevering nr.1» (se rett under).

## Delinnlevering nr.1

**Innen fredag 17.mars 2017 kl.10:00** skal dere ha gjort følgende:

Levert (*på papir*) til emnelærer:

1. navnet på gruppedeltagerne, inkl. kontaktinfo (mail og mobil) for alle i gruppen.
2. *ett* A4- eller A3-ark med *detaljert* tegning av datastrukturen.
3. gruppereglene, signert av alle gruppens medlemmer.
4. individuelt signert bekreftelse fra *hver* av gruppedeltagerne  
(jfr. pkt.1.3 på websiden om prosjektet).
5. utført, avkrysset og signert sjekkliste (jfr. pkt.6 på websiden om prosjektet).

Dataene i pkt.1 legges også inn i README-filen for prosjektet på Bitbucket  
(fjern all den andre teksten Bitbucket foreslår skal være med der).

Dokumentene i pkt.2-3 ovenfor skal også (innscannet) legges inn i GruppeXX-mappen deres på Google Docs.

## Delinnlevering nr.2

**Innen onsdag 29.mars 2017 kl.09:00** skal dere ha lastet opp (committed) deres siste versjon av koden i prosjektet. Dere skal *minst* ha gjort ferdig (kodet og fungerer korrekt) t.o.m. pkt.4 under «Forslag til rekkefølge på implementasjon» (ellers vil dere ligge veldig dårlig an ..... ☹). *I tillegg skal alle aktuelle testfiler i GruppeXX-mappen på Google Docs være oppdatert* (jfr. pkt.2.6 på websiden om prosjektet). *Disse lages/fylles ut etter hvert som dere skriver pseudokode for hver kommando (altså før dere begynner å skrive selve koden).*

## Sluttinnlevering

**Innen fredag 7.april 2017 kl.12:00** skal dere ha:

- lastet opp (committed) deres fungerende, endelige og siste versjon av koden i prosjektet.
- lagt inn *minst* de obligatoriske testdataene i alle filene (jfr. pkt.7 på websiden om prosjektet).
- *Testkjørt prosjektet i god tid før fristen ved å clone det hele ned til en helt ny katalog, og kjørt det derfra.* (Dette blir en simulering av hvordan fagassene vil teste/kjøre og oppleve programmet.)
- i GruppeXX-mappen (på Google Docs):
  - oppdatert og ferdigstilt (fylt ut komplett) alle testfilene.
  - lagt *en fil* med beskrivelse av *alle* filformatene og eksempler på filenes utseende.
  - evt. ha lagt *en* egen fil («readme.doc/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 best ved å sette opp klare og konkrete grupperegler (se pkt.1.2 på websiden om prosjektet, som er *innspill* til dette). *Jobb mye, effektivt og målrettet allerede fra første stund* (dvs. start «langspurten» tidlig, se «Gjøre mål 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://folk.ntnu.no/frh/grprog/obliger#Gen\\_reg](http://folk.ntnu.no/frh/grprog/obliger#Gen_reg) (spesielt det åttende punktet)

**Lykke til!**

**FrodeH**