|  |  |  |
| --- | --- | --- |
| HIST_logo_farge_liten150[1] | **HØGSKOLEN I SØR-TRØNDELAG**  **Institutt for informatikk og e-læring - IIE** | |
| **Målform:** | | Bokmål |
| **Eksamensdato:** | | 8. desember 2015 |
| **Varighet:** | | 4 timer |
| **Emnekode:** | | TDAT 1001 |
| **Emnenavn:** | | Programmering grunnkurs - S2 |
| **Klasse(r):** | | 1 ING |
| **Studiepoeng:** | | 10 |
| **Faglærer(e):** | | Bjørn Klefstad, Grethe Sandstrak |
| **Kontaktperson (adm.)** | | Ingrid Island (tlf. 73 55 95 64) |
| **Hjelpemidler:** | | Alle trykte hjelpemidler |
| **Oppgavesettet består av:** | | En oppgave og tre sider (inkludert forside og vedlegg) |
| **Vedlegg består av:** | | Ingen vedlegg |
| **Merknad:** NB! Les *hele* oppgaveteksten før du setter i gang, og planlegg tiden.  Dersom noe virker uklart i noen av oppgavene, må du selv gjøre antagelser og få med disse antagelsene i besvarelsen.  **Lykke til!** | | |

## Oppgave 1

Du skal i denne oppgaven lage et program for å administrere idrettsarrangementer. Et arrangement kan inkludere flere idrettsgrener (friidrett, fotball, bryting, sykling), hver idrettsgren kan ha flere øvelser (100 m hekk – Kvinner, 100 m – Menn, Spyd – Kvinner). Den enkelte øvelsen skal beskrives med et navn og hvilket kjønn (Kvinner/ Menn) øvelsen er for.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Ovelse |  | Idrettsgren |  | Arrangement |  |
| beskrivelse  kjønn |  | navn på gren  ovelser |  | navn på arr  idrettsgrener |  |
|  |  |  |  |  |  |

1. Les **hele** oppgavesettet og sett opp komplett utvidet klassediagram for alle klasser, som også viser forholdene mellom klassene. Begrunn kort valg av binding mellom objektene (aggregering/ komposisjon). Klassene som skal inngå er: Ovelse, Idrettsgren, Arrangement og Arrangement\_klient.
2. Lag den mutable klassen Ovelse, med konstruktør, tilgangsmetoder og toString-metode. Eksempel på utskrift av et Ovelse-objekt:

400m hekk Kvinner

1. For å unngå dobbeltlagring av øvelser trenger vi en metode for å sjekke likhet. Lag en metode i klassen *Ovelse* som sjekker likhet med et annet Ovelse-objekt. Objektene er like hvis både beskrivelse og kjønn er lik.
2. Vi går nå over til klassen Idrettsgren. Lag en konstruktør som i tillegg til navn, setter Maks størrelse på antall øvelser (maks størrelse skal ikke kunne endres senere i programmet). Lag også en konstruktør som i tillegg til navn, tar inn en utfylt tabell med øvelser. Sett opp passende objektvariabler og toString-metode for klassen. Eksempel på utskrift av et idrettsgren-objekt:

Friidrett

Spyd Kvinner

Spyd Menn

400m hekk Kvinner

400m hekk Menn

1. Lag en metode i klassen Idrettsgren som registrerer en ny øvelse. Ny øvelse kan kun registreres dersom det er plass og dersom øvelse med samme beskrivelse og kjønn ikke er registrert fra før. Metoden skal returnere true dersom registrering går bra, false hvis ikke.
2. Lag klassen Arrangement. Ett arrangement har et navn og kan ha flere idrettsgrener. Maks antall idrettsgrener settes i konstruktør og kan ikke endres senere. Lag klassen med passende objektvariebler, konstruktør og toString-metode. Eksempel på utskrift av et Arrangement-objekt:

IIE-lekene

Vektløfting

48kg Kvinner

56kg Kvinner

56kg Menn

77kg Menn

Friidrett

Spyd Kvinner

Spyd Menn

400m hekk Kvinner

1. Lag en metode i klassen Arrangement som registrerer en ny idrettsgren. Ny Idrettsgren kan kun registreres dersom det er plass og dersom Idrettsgren med samme navn ikke er registrert fra før. Metoden skal returnere true dersom registrering går bra, false hvis ikke.
2. Lag et menystyrt klientprogram (Arrangement\_klient) som gir bruker følgende valg:

* Opprette nytt Arrangement. Dersom arrangement er registrert fra før skal det gis en advarsel om at alle registrerte data vil bli slettet dersom en fullfører dette valget.
* Registrere ny Idrettsgren på arrangementet med gitt-antall øvelser. (Navn og Antall øvelser leses inn fra bruker, registrer så det oppgitt antall øvelser før nytt Idrettsgren-objekt opprettes). Det skal gis tilbakemelding på om registreringen gikk bra eller ikke. Dersom bruker taster inn feil datatype (for eksempel tekst i stedet for tall) skal programmet håndtere dette og gi en fornuftig tilbakemelding til bruker (Du kan bruke samme tilbakemelding på alle typer feilsituasjoner som oppstår). Dersom nytt arrangement ikke er opprettet skal bruker få beskjed om dette.
* Avslutt.

1. Alle data om ett arrangement skal lagres på fil mellom hver kjøring (arrangement.ser) og ved oppstart skal eventuelle lagrede data leses inn. Lag klassemetoden for å lese data fra fil (og kun fra fil):

* public static Arrangement lesFraFil(String filnavn)

Lag metoden på en slik måte at feilsituasjoner håndteres. Hvilke endringer må du evt. gjøre i koden fra oppgavene over?

1. Lag en JUnit-test for klassen Idrettsgren. Det holder at du lager JUnit-tester for metoden for å registrere en ny øvelse. Denne skal til gjengjeld testes på minst tre ulike måter for å fange ulike situasjoner som kan oppstå under kjøring.