BOKMÅL

Universitetet i Sørøst-Norge

EKSAMENSFORSIDE

Skoleeksamen med tilsyn

Generell informasjon om eksamen:

Emnekode: PB1090

Emnenavn: Programmering 2 - Objektorientert programmering

Dato: 13.05.2024

Start klokkeslett: 09:00-13:00

Antall timer: 4

Emneansvarlig: Lars Erik Opdal & Joakim Bjørk

Campus:
Vestfold

Fakultetet: TNM

Antall oppgaver: 5

Antall vedlegg: 0

Totalt antall sider (inkludert forside og vedlegg): 6

Tillatte hjelpemidler (jfr. emneplan):

Ingen

Opplysninger om vedlegg:

Click or tap here to enter text.

Merknader:

Studenter på Campus Bakkenteigen skriver Java syntaks
Studenter på Campus Kongsberg skriver C++ syntaks

Tips.
I Wiseflow; bruke vedlegg -> kode og velge Java eller C++ og evt. et farge-tema for syntaks

Ved eksamen på papir kryss av for type eksamenspapir:
Ruter ☐ Linjer ☐

KANDIDATEN MÅ SELV KONTROLLERE AT OPPGAVESETTET ER FULLSTENDIG

Oppgave 1 (20%)

Klassehierarki

Implementer klassehierarkiet med alle datafelter/medlemsvariabler som *Figur 1 - UML* beskriver, lag også konstruktører til klassene. Klassene *Student* og *Ansatt* arver fra *Person*

- Om du skriver Java skal du implementere toString-metode for Person-klassen.
- Om du skriver C++ skal du overlaste << operatoren for Person-klassen.

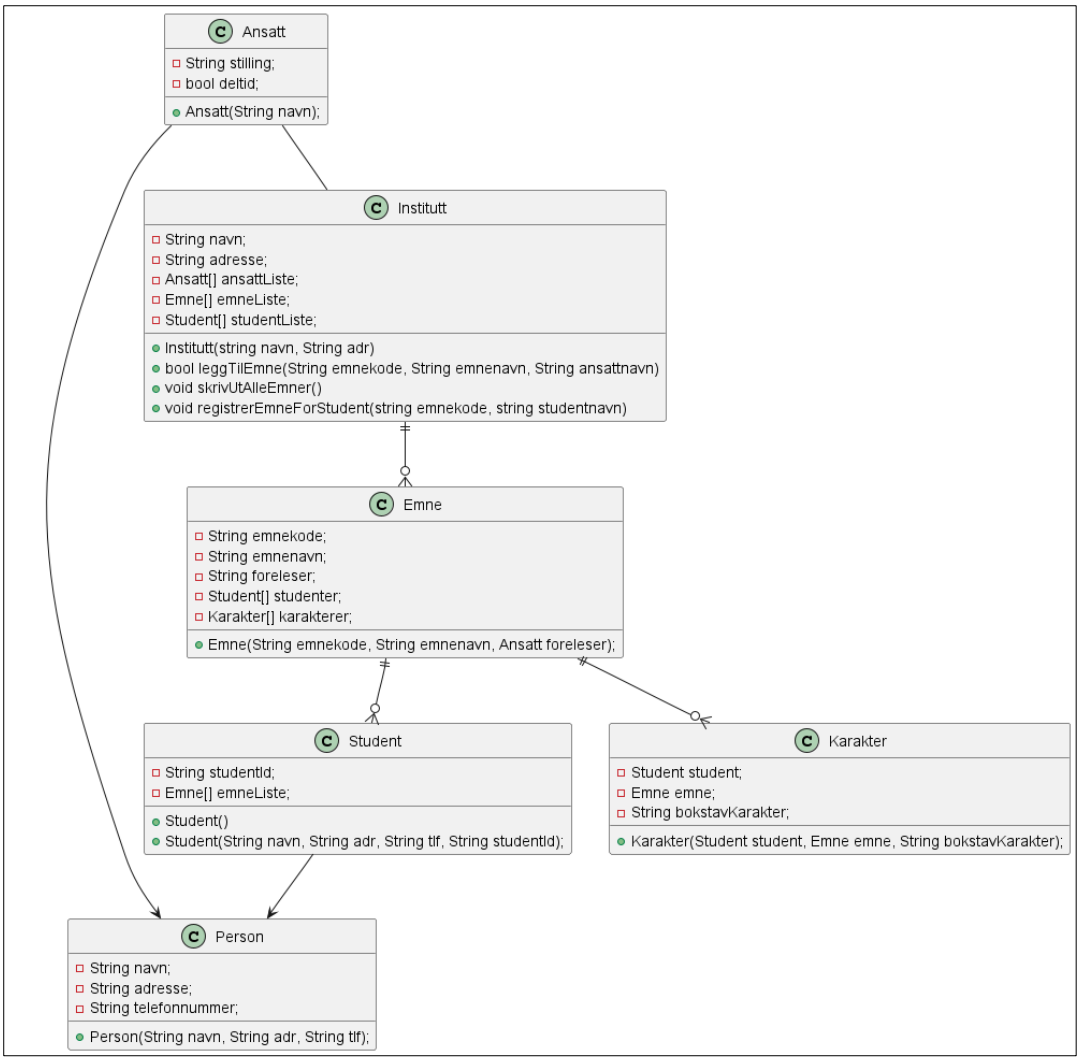
Du kan skrive inn signatur for resten av metodene/funksjonene for klassene, men de skal ikke ha noen implementasjon før i de neste oppgavene.

Det skal også være mulig å registrere nye Studenter med og uten personopplysninger. Lag to konstruktører for Student-klassen. Ansatt-klassen skal bare ha en konstruktør. Konstruktørene som tar parametere skal gi verdier til alle datafelt/medlemsvariablene.

Datafelt/medlemsvariablene skal være private, mens metoder/funksjoner skal være offentlig (public).

Du kan velge hvor du vil bruke referanser/pekere selv om det ikke kommer frem av diagrammet. Der variablene som skal ha referanser/pekere til flere elementer kan du velge om du vil bruke array for objekter eller ArrayList/vector.

NB. Du trenger ikke implementere gettere og settere, men anta at de finnes om du trenger dem.



Figur 1 - UML

Oppgave 2 (20%)

Oppretting av objekter og bruk av datastruktur

Lag følgende metoder for disse i Institutt-klassen:

a) *leggTilEmne* skal ta en emnekode, et emnenavn og et ansattnavn som parametere og opprette et nytt emne objekt. Det nye objektet skal legges inn i emnelisten.

b) *skrivUtAlleEmner* skal skrive ut emnekode og emnenavn på alle emner i emnelisten.

c) *registrereEmneForStudent* skal ta en emnekode og et studentnavn som parametere og legge emnet til i studentens *emneliste* og legge studenten til i emnets liste for *studenter.* Metoden skal returnere true om alt gikk bra og false om noe gikk galt.

Oppgave 3 (25%)

Array & lister: Søk og filtrering av studenter

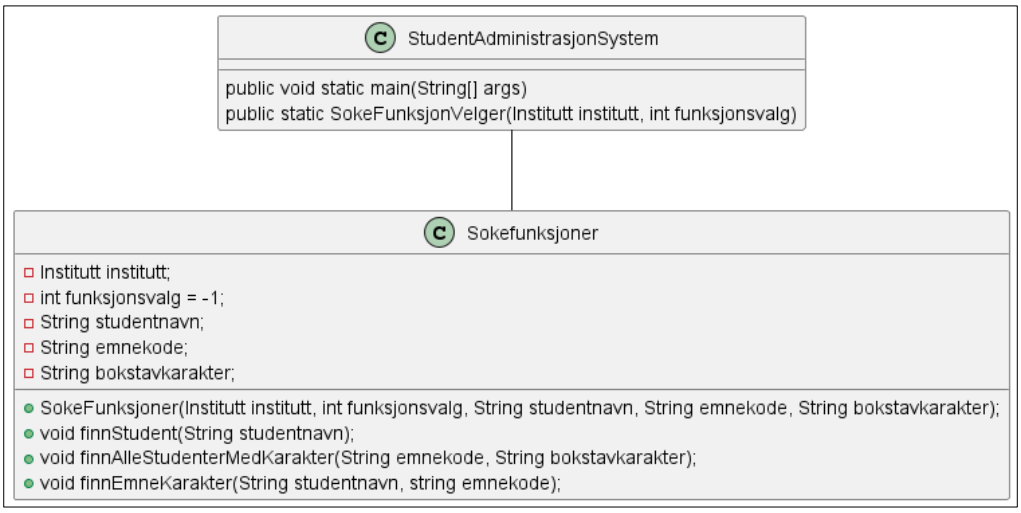
a) Implementer klassen *SokeFunksjoner*. Du skal implementere en konstruktør som *Figur 2 – UML* beskriver.

Det skal implementeres tre metoder/funksjoner for å søke etter studenter og for å kunne filtrere etter emne og karakter. Metodene som er beskrevet under her skal implementeres i klassen:

b) Lag en metode som heter *finnStudent(String studentnavn)*. Den skal søke gjennom alle student-objektene. Skriv ut studentobjektet som har samme navn som parameteren. Dersom flere har samme navn skal alle skrives ut.

c) Lag en metode som heter *finnAlleStudenterMedKarakter(String emnekode, String karakter).* Den skal skrive ut alle studentene som har denne karakteren i dette emnet.

d) Lag en metode som heter *finnEmneKarakter(String studentnavn, String emnekode).* Den skal skrive ut hvilken karakter denne studenten fikk i dette emnet.



Figur 2 - UML

Oppgave 4 (25%)

Brukerinteraksjon: main-metode og oppstart av tråd

Start med å opprette et objekt av *Institutt*.

- Om du skriver Java skal du også opprette en klasse som heter StudentAdministrasjonSystem.
- Om du skriver Java skal du også opprette objekt et for bruker-input.

I main skal du skrive ut en meny som ser slik ut:

Velkommen til hovedmenyen

Du kan velge følgende:

1. Skriv ut alle emner
2. Søk etter en student med navn
3. Finn alle studenter i et emne med en karakter (A,B,C,D,E,F)
4. Finn karakter til en student i et emne
0. for å avslutte

Brukeren skal kunne gjøre valg helt til tallet 0 blir registrert.

Hint: *while(valg !=0)*

Om du skriver Java: For å utføre et av valgene fra menyen, skal det lages en statisk metode som tar inn et heltall som parameter. Bruk signatur fra *Figur 2 - UML* for å lage metoden/funksjonen: *sokeFunksjonVelger(…)*

Om du skriver C++ skal *sokeFunksjonVelger(…)* være en funksjon som ikke ligger i noen klasse.

Metodene/funksjonene som hører til et av meny-valgene fra 1-4 skal det brukes her, det vil si:

skrivUtAlleEmner()
finnStudent(String studentNavn)
finnAlleStudenterMedKarakter(String emnekode, String bokstavkarakter)
finnEmneKarakter(String studentnavn, String emnekode)

Metoden/funksjonen *sokeFunksjonVelger(…)* skal startes opp av en egen tråd.

Oppgave 5 (10%)

Egendefinert unntak for feil inntasting

Lag et egendefinert unntak for feil inntasting av tallverdi for ordreløkken. Dersom bruker av system skriver inn feil tall skal det bli skrevet ut en melding der det står: "Du har tastet feil, velg et tall mellom 0-4". For at ordreløkken skal kunne avsluttes med feilmelding etter eventuell feil inntasting, skal unntak bli kastet dersom tallet som velges ikke er mellom 0-4.

- Om du skriver Java skal Main-metoden ha med unntaket i sin signatur.