

**EKSAMENSFORSIDER**

Hjemmeeksamen

Emnekode: <b>OBJ2000-1</b>	Emnenavn: Objektorientert programmering 1	
Ansvarlig faglærer: Trond Braadland	Campus: Ringerike	Fakultet: Handelshøyskolen
Utlev. dato og tidspunkt i WISEflow: 15. desember kl. 09.00	Innlev. dato og tidspunkt i WISEflow: 15. desember kl. 13.00	
Antall oppgaver: 4	Antall vedlegg: 0	Ant. sider inkl. forside og vedlegg: 3
<b>Hjelpebidrifter og samarbeid:</b>		
Tillatte hjelpebidrifter: Alle hjelpebidrifter er tillatt.		
Er det individuell eksamen?		Ja <input checked="" type="checkbox"/> Nei <input type="checkbox"/>
Er det tillatt å samarbeide med andre personer?		<input type="checkbox"/> <input checked="" type="checkbox"/>
<b>Kriterier for besvarelsen:</b>		
Skrifttype:	Skriftstørrelse:	Linjeavstand:
Antall ord (min/maks):	Maks antall sider ekskl. forside og vedlegg:	
Kildehenvisning:		
Andre viktige opplysninger:		

**Les gjennom hele eksamenssettet før du begynner. Begynn helst på en ny side for hver oppgave. Ikke skriv mer enn du blir bedt om. Du skal ikke skrive et helt program!**  
**Innlevering som pdf-fil.**

Tuppereiser AS er et reisebyrå som setter opp bussturer til forskjellige destinasjoner (bestemmelsessteder) innenlands og utenlands. De trenger nå en applikasjon for å holde rede på turer, deltagere og destinasjoner. Applikasjonen skal bestå av tre lag: et presentasjonslag i form av en grensesnittsklasse, et applikasjonslag i form av en kontrollklasse og et doménelag i form av klasser for reiser, deltagere og destinasjoner.

Du kan gå ut fra at du har tilgang til disse to klassene (du skal altså ikke skrive dem):

- Klassen Destinasjon skal ha et destinasjonsnummer av typen heltall, destinasjonens navn (eks. Strömstad) og avstand fra Hønefoss i hele kilometer (alle turer starter i Hønefoss). Klassen skal også ha en konstruktør som setter verdiene samt metoden `toString()`.
- Klassen Deltager med attributtene deltagnr, navn, adresse og telefonnummer. Denne klassen har også en konstruktør som setter attributtverdiene og metoden `toString()`. En deltager registreres på nytt for hver reise.

## Oppgave 1 (25 %)

Klassen Reise skal ha et reisenummer (heltall) og datoer for avreise og hjemkomst. Videre skal den ha en `ArrayList` med alle deltagerne på reisen og en referanse til destinasjonen. Imidlertid kan reiser være av to typer: firmareiser og reiser for privatpersoner. Firmareiser skal ha attributtene firmanavn og kontaktperson i tillegg til de felles attributtene.

- a) Skriv en abstrakt superklasse `Reise`, med `toString()` og konstruktør.
- b) Skriv klassen `Firmareise`, også denne med `toString()` og konstruktør.
- c) Skriv en metode `nyDeltager()`, som skal ha en deltager som parameter og sette denne inn i `ArrayList`'en med deltagere.
- d) Skriv en metode `getDeltagere()`, som returnerer alle deltagere på en reise.

## Oppgave 2 (30 %)

Applikasjonen skal ha en klasse kalt `Grensesnitt` som håndterer all kommunikasjon med brukeren, og en klasse `Kontroll` som kontrollerer objektene. I `Grensesnitt` skal det opprettes et objekt av klassen `Kontroll`, som vil ha de metodene som er angitt i oppgave 3. Bruk `JOptionPane` for å lage brukergrensesnitt. Her er metodene du skal programmere i klassen `Grensesnitt`:

- a) Skriv en metode `nyFirmaReise()` som spør brukeren etter data om en reise og sender disse over til kontrollobjektet.
- b) Skriv en metode `nyDeltager()` som skal spørre etter nummeret på en reise og deretter data om en deltager. Disse skal også sendes til kontrollobjektet.
- c) Skriv en metode `visReise()` som spør etter nummeret på en reise, kaller metoden `getReise()` fra kontrollobjektet og deretter lister ut dato for avreise og hjemkomst, samt data om alle deltagere. Test på om det returneres et objekt.
- d) Skriv en metode `alleReiser()` som skriver ut alle registrerte reiser.

### Oppgave 3 (30 %)

Skriv en klasse Kontroll som har en ArrayList med alle reiser og destinasjoner. Videre skal du skrive disse metodene:

- a) En metode nyFirmaReise( ) som mottar data fra metoden i punkt a) i forrige oppgave. Metoden skal selv tildele reisenummer i løpende rekkefølge. Videre skal den opprette et objekt av klassen FirmaReise og legge dette inn i ArrayList'en for reiser.
- b) Skriv en metode getReise( ) som mottar et reisenummer fra metoden i punkt c) i forrige oppgave og bruker linjært søk for å finne og returnere et objekt av klassen Reise (eller null).
- c) Skriv en metode nyDeltager( ) som mottar data fra metoden i punkt b) i forrige oppgave. Metoden skal finne riktig reise i ArrayList'en og deretter opprette et objekt av klassen Deltager og legge dette inn i ArrayList'en for deltagere.
- d) Skriv en metode getReisene( ) som returnerer en referanse til datastrukturen.

### Oppgave 4 (15 %)

I stedet for linjært søk skal vi bruke binærersøk. Igjen er det reisenummer som brukes til søket.

- a) Gjør de nødvendige endringene i klassen Reise for å kunne bruke Javas binærersøk-metode.
- b) Hvordan vil det se ut om reisenummer er en String i stedet for et heltall?
- c) I Kontroll skal du skrive en metode for binærersøk på reisenummer.

*Lykke til!*