Eksamen IN1010 våren 2025

Sensorveiledning

Generelt

I utgangspunktet får man full uttelling for det enkelte delspørsmålet om

- koden er riktig
- den gjør det oppgaven ber om
- er oversiktlig og lettlest

Effektivitet er ikke viktig i dette emnet.

Rene skrivefeil ignoreres vanligvis, men feil som kan tyde på manglende forståelse (for eksempel kode som ikke passer inn eller svarer på oppgaven), skal gi trekk.

Dette ignoreres:

- mangler enkelte «`;`» , «`{`» eller «`}`» om koden ellers gir mening
- småfeil i navn

Studentene har fått beskjed om at **`import`**-setninger, kommentarer eller tilgangsmodifikatorer (**`public`**/**`private`**/**`protected`**) ikke er viktig på eksamen.

Feil i kopiert kode trekkes kun en gang.

Se også «Generelt om oppgaven» i oppgavesettet.

Poengene er *veiledende* og endelig karakter settes etter en *helhetlig vurdering*.

Oppgave 1 Klassen Samling (10 poeng)

Her skal studentene vise at de kan implementere en klasse med en gitt datastruktur og at de behersker enkel lenkelisteprogrammering.

Det er viktig i denne oppgaven at studenten

- bygger på koden gitt i oppgaven
- implementerer Node-klassen korrekt
- implementerer konstruktøren korrekt (4 poeng)
- implementerer `settInn`-metoden korrekt (4 poeng)
- bruker typeparameteren E riktig

Det er ikke så viktig om de bruker navn i figuren som ikke er nevnt i kodeskjelettet (dvs `neste` og `data`).

Oppgave 2 Iterator (10 poeng)

Her skal studentene vise at de kan implementere en iterator.

Dette må de ha med:

- Implementerer `Iterator`-klasse med `next()` og `hasNext()` korrekt

Dette punktet er spesielt viktig og teller 8 poeng av 10. Å kode slik at iteratoren hopper over alle elementer som er **`null`**, teller 5 av de 8 poengene.

- Implementerer `iterator`-metoden korrekt
- endrer signaturen av `Samling`-klassen (dvs ha med «`implements Iterable<E>`»)

Oppgave 3 Klassediagram (8 poeng)

Her skal studentene vise at de forstår klassehierarkier med klasser og subklasser, og at de kan uttrykke dette i et klassediagram tegnet slik det er vist i forelesningene og annet undervisningsmateriale.

Dette må de ha med:

- Alle klassene `Ansatt`, `Kelner`, `Vinkelner` og `Lærling` samt interface-et `Sommelier`
- Viser korrekt hva som er subklasse av hva
- Viser sammenhengen mellom klassen `Kelner` og interface-et `Sommelier`
- Angir `Ansatt` som **`abstract`** (teller 1 poeng)

Oppgave 4 Klassedeklarasjoner (10 poeng)

Her skal studentene vise at de kan programmere klasser og subklasser i Java.

De må ha med alle de 8 nevnte klassene: `Restaurant`, `Ansatt`, `Kelner`, `Vinkelner`, `Lærling`, `Bord`, `Gjest` og `Gjestegruppe`. Interface-et `Sommelier` er gitt i oppgaveteksten, så det trenger de ikke ha med.

Dette må være i orden:

- Klassestrukturen (dvs **`extends`** og **`implements`**) må være som beskrevet i oppgaven.
- Alle instansvariabler nevnt i oppgaveteksten må være med. Pass spesielt på at `Samling`-er deklareres og initieres riktig.
- Initieringen i konstruktøren må være riktig (dvs kalle **`super`**`(...)` for parametre til superklassen). Det er ikke nødvendig å opprette bordene i restauranten, men det trekkes ikke dersom det gjøres.
- Det skal ikke brukes andre beholdere enn `Samling`.
- Merk at oppgaven ikke krever at metoden `spesialitet` implementeres.

Oppgave 5 Søking (8 poeng)

Her skal studentene vise at de kan søke gjennom en datastruktur ved å bruke en iterator.

- Metoden `ønskerVin()` er triviell og teller bare 1 poeng av 8.
- Hvis de ikke bruker iterator i løsningen, skal det trekkes 4 poeng.
- Det trekkes 1 poeng hvis de ikke avslutter søket når metoden har funnet en vinelsker.

Oppgave 6 Finn beste (8 poeng)

Her skal også studentene vise at de kan lete seg frem til et svar, men nå er kriteriet litt vanskeligere (men ikke mye).

- En løsning som sjekker antall bord i stedet for antall gjester, trekker 4 poeng.

Oppgave 7 Rekursjon (10 poeng)

Her skal studentene demonstrere sin evne til å programmere en rekursiv løsning. Det er mange måter å gjøre dette på, og det er det samme hvilken de velger så lenge den er rekursiv.

- En riktig ikke-rekursiv løsning gir bare 2 poeng.
- Trekkes 3 poeng dersom tomme noder telles med i antallet.

Oppgave 8 Ta imot gjester (12 poeng)

Her skal studentene vise at de kan programmere en metode som er godt forklart, men ikke fullspesifisert. De må selv finne ut hva som må gjøres i noen tilfeller.

Oppgave 9 GUI (10 poeng)

Her skal studentene vise at de kan opprette et GUI-vindu med tekst og trykknapp.

- Brukerinteraksjonen med OK-knappen (dvs `ActionListener`-klassen og `actionPerformed`-metoden) er viktigst og teller 5 av 10 poeng.
- Innholdet i `actionPerformed`-metoden er ikke viktig, og en kommentar er tilstrekkelig.
- 5 av 10 poeng gis til å få riktig struktur på GUI-oppsettet, som inkluderer
    - Et vindu (`JFrame`)
    - En tekst (`JLabel`)
    - En knapp (`JButton`)
    - At vinduet pakkes og synliggjøres
- Vær raus med ulike navn på klasser og metoder.

Oppgave 10 Tråder (14 poeng)

Her skal studentene vise at de behersker trådprogrammering.

- Å lage en monitor av `Restaurant` med riktig låsing av metodene `taImotGjester` og `forlatBordet` teller 4 poeng.
- Riktig bruk av synkronisering med `Condition`-en teller 6 poeng.
- Klassen `Restaurantbesøk` med `run`-metoden teller 3 poeng.
- Riktig håndtering av unntak teller 1 poeng.