Konteeksamen IN1010

Sensorveiledning

??. august 2025

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 én gang.

Se også «Generelt om oppgaven» i oppgavesettet.

Poengene danner grunnlaget for den endelige bokstavkarakteren som settes til slutt etter en helhetlig vurdering.

Oppgave 1 Klasser og klassestruktur (vekt 10 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 de nevnte klassene samt interface-et Brukerkommunikasjon.
- Viser korrekt hva som er subklasse av hva.
- Pilene må gå riktig vei.
- Viser sammenhengen mellom klassen AutomatrettetOppgave og interface-et Brukerkommunikasjon.
- Angir Oppgave, AutomatrettetOppgave og ManueltrettetOppgave som **abstract** (teller 1 poeng).

Oppgave 2 **Programmering av klassene** (vekt 10 poeng)

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

De må ha med alle de åtte nevnte klassene: Oppgave, AutomatrettetOppgave, ManueltrettetOppgave, Flervalgsoppgave, Tallsvaroppgave, Ordsvaroppgave, Programmeringsoppgave og Tekstsvaroppgave. Interface-et Brukerkommunikasjon 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. De abstrakte klassene (se forrige oppgave) må være deklarert som **abstract**.
- Alle instansvariabler nevnt i oppgaveteksten må være med. Det er greit om instanvariable deklareres som **final**, men studentene får ingen bonus for det.
- Initieringen i konstruktøren må være riktig (dvs kalle **super**(. . .) for parametre til superklassen).

Oppgave 3 **Sammenligning** (vekt 6 poeng)

I denne oppgaven skal studentene vise at de kan implementere et Comparable-grensesnitt.

- Klassen må angi «implements Comparable<Oppgave>» (1 poeng for «implements Comparable» og 1 poeng til for «<Oppgave>»).
- Metoden compareTo må implementeres (3 poeng). Gal impelmentasjon (f eks beregne $b - a$ i stedet for $a - b$) gir 1–2 poeng trekk.
- 0,5 poeng ekstra om compareTo er deklarert **public** og 0,5 poeng til om @Override er med.

Oppgave 4 **En toppliste** (vekt 12 poeng)

Her skal studentene få demonstrere at de behersker lenkelister og spesielt sorterte lister.

- Selve det å sette inn i listen skal telle 4 poeng.
- Riktig sortering gir inntil 5 poeng til.
- Fjerning av overflødige elementer teller 3 poeng.

Oppgave 5 **Oppgavesamling** (vekt 12 poeng)

I denne oppgaven skal studentene vise at de kan implementere en ikke helt triviell datastruktur.

- Klassen Oppgavesamling med riktige instansvariabler alleOppgaver og topp5 og korrekt instansiering av disse teller 4 poeng.
- Metoden registrerOppgave teller 4 poeng.
- Metoden antallAutomatiske teller også 4 poeng.

Om noen velger å la disse metodene kalle på andre metoder som skal skrives senere i oppgavesettet, er det bare positivt.

Oppgave 6 Sorteringssjekk (vekt 10 poeng)

Det viktigste i denne oppgaven er å se om studentene kan skrive rekursiv kode. En korrekt ikke-rekursiv løsning vil derfor bare gi 3 poeng.

Oppgave 7 Endringer i topplisten (vekt 10 poeng)

Denne oppgaven er en test på om studentene kan løse et programmeringsproblem uten at det er spesifisert hvordan de skal løse det. Det er gitt et hint, men studentene er ikke forpliktet til å følge det. Det viktigste er at de levere kode som er oversiktlig og lettforståelig, og selvfølgelig riktig.

Oppgave 8 Brukergrensesnitt terminal (vekt 10 poeng)

Her skal studentene vise at de kan programmere en enkel brukerdialog, men sammen med oppgave 10 skal de også vise at de forstår hvordan et interface kan implementeres helt ulikt i forskjellige sammenhenger. Nøyaktig hvordan brukerdialogen ser ut er ikke viktig – alle fornuftige forslag blir godtatt.

Oppgave 9 Beholder for utsatt henting (vekt 10 poeng)

I denne oppgaven skal de skrive en enkel generisk monitor.

- Selve klassen med generisk parameter og en instansvariabel med plass til ett element teller 3 poeng.
- Opprettelse av lås og låsing av de to metodene teller 4 poeng.
- Deklarasjon av Condition-variabel og riktig bruk av den teller 3 poeng.

Oppgave 10 Brukergrensesnitt GUI (vekt 10 poeng)

Her skal studentene skrive en enkel GUI-dialog. Det spiller ikke så stor rolle nøyaktig hvordan dialogen ser ut, men den skal ha funksjonalitet slik det er beskrevet i oppgaveteksten.

- Riktig håndtering av brukersvaret med komminikasjon via et Utsatt-objekt teller 4 poeng.

Om koden får problemer hvis metoden kalles flere ganger, skal det ikke trekkes for det.