

# Eksamen kont PGR112 – Objektorientert programmering V2024

Individuell hjemmeeksamen. 24 timer.

**All kode må kodes av deg og deg alene.** Hvis du baserer koden din på eksisterende kode, så må du tydelig vise (i kommentarer) hva kilden var. Det samme gjelder kode generert av AI-verktøy som ChatGPT eller tilsvarende.

Du bør legge alle filene som leveransen inneholder i én mappe og zippe før du laster opp zip-en til WISEFlow.

Løsningen skal kunne kjøres i IntelliJ (dvs. trenger ingen annen spesiell teknologi/IDE)

Hvis du er usikker på noe angående eksamen, les instruksjonene på nytt og prøv å gjøre en beste antagelse. Gi tydelig uttrykk for hvilke forutsetninger du gjorde og hvorfor i rapporten (se nedenfor).

Leveransen vil inneholde to ting:

1. Prosjektmappen med alle relevante filer for å kunne kjøre og vurdere koden du har laget.
2. En liten rapport (pdf eller word-dokument) som kort beskriver hvordan du har håndtert
  - unntakshåndtering
  - innkapsling
  - arv

og eventuelle forutsetninger du har lagt til grunn for løsningen din (hvis du mener oppgaveteksten inneholdt uklarheter).

## Vurdering

Bestått/Ikke bestått.

For å oppnå bestått:

*Kandidaten har oppnådd kravene til læringsutbytte og viser nødvendige kunnskaper, ferdigheter og kompetanse.*

Emnebeskrivelsen viser læringsmålene i emnet. Blant temaene i emnebeskrivelsen vil eksamen særlig måle læringsutbytte innen følgende temaer:

- Arv og polymorfi
- Abstrakte klasser og interface (løsningen din bør inneholde et eksempel på minst en av dem)
- Unntakshåndtering
- Kopling, samhörighet og innkapsling (og generell kodekvalitet). Det er en klar forventning at løsningen viser eksempel på innkapsling.
- JDBC
- Filhåndtering

Sensor vil ta utgangspunkt i levert kode for å måle måloppnåelse innenfor disse temaene. I tillegg vil sensor benytte seg av rapporten der noen av temaene er utdypet. En bestått besvarelse vil vise måloppnåelse innen alle temaene ovenfor. Høy måloppnåelse innen flere læringsmål kan oppveie for svak måloppnåelse innen et fåtall temaer. Sensor vil foreta en totalvurdering av leveransen i sin vurdering av besvarelsen. Det er dermed ikke en spesifikk prosentvis score som må oppnås for å

bestå eksamen. Det er en totalvurdering fra sensor basert på læringsmålene som måles gjennom eksamen.

### Oppgaven

Caset omhandler samlerkort. En person (la oss kalle han Per) har en vesentlig samling med samlerkort. Per ønsker å få bedre oversikt over kortene sine. Han har laget en tekstfil med informasjon om kortene. Per ønsker å få hjelp til å lese inn informasjon fra filen og legge dataene inn i en database. For en beskrivelse av filen, se vedlegg 1 i dette dokumentet.

#### Del 1 - importere data til database:

Du skal lage et program som leser fra filen samlerkort.txt og legger dataene inn i en database. Et forslag til databasestruktur finner du i filen samlerkort.sql. Du kan velge å gjøre justeringer i samlerkort.sql, men det skal ikke være nødvendig.

Når du er ferdig med del 1 bør du ha:

- En velfungerende database som har informasjon om samlerkortene. Tabellene i databasen er fylt med data basert på filen samlerkort.txt.
- En eller flere klasser som kan kommunisere med databasen.
- En eller flere klasser som kan holde i data fra databasen. **Det er forventet at arv benyttes for klasser som omfatter ulike typer samlerkort ettersom de har mange felles egenskaper. Det er også forventet at disse klassene blir benyttet i kommunikasjon med databasen.**

OBS! Husk at eksamenstiden er begrenset. Hvis du ikke klarer å komme i mål med del 1 innen rimelig tid, så er du nødt til å gå videre til del 2. Ta i så fall vare på koden du skrev for del 1, slik at den kan bli vurdert.

#### Del 2 – bruke databasen:

Du skal lage et lite program som tilbyr brukeren en meny med noen menyvalg som omhandler samlerkortene som nå er plassert i databasen (etter at del 1 er gjennomført). Du kan selv velge om du vil ha to main-metoder (en for del 1 og en for del 2), eller om du vil kommentere ut koden for del 1 når du utvikler del 2. Eller du kan velge å ha del 1 og del 2 i samme program.

Programmet skal tilby følgende funksjonalitet:

- Se informasjon om alle samlerkort for en sport (brukeren oppgir sport).
- Få informasjon om antall samlerkort registrert.
- Se informasjon om alle samlerkort som er i «mint condition».
- Avslutte programmet.

Husk å inkludere filen som beskriver hvordan du har håndtert unntakshåndtering, innkapsling og arv før du zipper og leverer.

**Lykke til!**

## Vedlegg 1 – om filen samlerkort.txt

Overordnet sett inneholder filen informasjon om samlerkortserier og tre ulike typer samlerkort: Fotballkort, Basketballkort og Baseballkort. En samlerkortserie er en serie med kort som er gitt ut et spesifikt år.

Filen starter med informasjon om samlerkortseriene:

*Samlerkortserier:* -> Overskrift som viser at informasjonen som nå kommer omhandler samlerkortserier.

11 -> Hvor mange samlerkortserier som er listet opp i filen.

1 -> Id til samlerkortserie

*Topps* -> Utgiver av samlerkortserien

1975 -> Årstallet samlerkortserien ble utgitt

*Fotball* -> Hvilken sport samlerkortserien omfatter

220 -> Hvor mange ulike kort det er i serien

[Flere samlerkortserier]

Deretter kommer informasjon om samlerkort:

----- -> en strek som markerer skille mellom samlerkortseriene og samlerkortene

*Kort:* -> Overskrifter som viser at informasjonen som nå kommer omhandler samlerkort.

199 -> Hvor mange samlerkort (kort) som er listet opp i filen.

1 -> Id til kort

5 -> Id til hvilken samlerkortserie kortet tilhører

*Excellent* -> Tilstanden til kortet. Et kort har en av følgende tilstander: Mint, Near Mint, Excellent, Good, Light Played, Played og Poor.

*Dennis Adams* -> Navn på spilleren på kortet

*Everton* -> Navn på klubben der spilleren spiller

5 -> Hvor mange sesonger spilleren har spilt (da kortet ble utgitt)

125 -> Hvor mange kamper spilleren har spilt (da kortet ble utgitt)

*Fotball* -> Sporten spilleren utøver. Denne sporten er avgjørende for hvilken informasjon som kommer videre for kortet. For sporten for fotball er det informasjon om antall mål i serien og cupen:

61 -> Antall seriemål

18 -> Antall cupmål

[for baseball er det kun informasjon om antall homeruns spilleren har hatt i sin karriere (når kortet ble utgitt. Som for eksempel slik for spiller med id=2, Brian Mitchell:]

6 -> Antall homeruns.

[for basketball er det informasjon om «fieldgoal percentage» (hvor ofte de treffer på skuddforsøk), «free-throw percentage» (hvor ofte de treffer på straffekast) og gjennomsnittlig antall poeng i kamp. Som for eksempel slik for Frank Miller (id=5):]

265 -> Dette indikerer at Ryan treffer på 26.5% av skuddene sine (så langt i karrieren sin)

519 -> Han treffer på 51.9% av straffekastene sine

24.5 -> Han har scoret i snitt 24.5 poeng i kampene han har spilt.

-----> En separator som viser at informasjonen om kortet er ferdig.

[Flere kort]

Som du ser, er det altså en del felles informasjon om alle samlerkort. Og for de ulike sportene er det forskjellig tilleggsinformasjon.