

Eksamen PGR112 – Objektorientert programmering juni 2025

(For nynorsk, sjå side 6)

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. Å referere til kilde gjelder også for bruk av KI. Vedlegg 2 viser retningslinjer for bruk av KI.

Du bør legge alle filene som leveransen inneholder i én mappe og zippe før du laster opp zip-en til WISEFlow. En leveranse som IKKE inneholder kildekoden til prosjektet ditt, vil bli vurdert til Ikke bestått.

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, gjerne rundt 300 ord) som kort beskriver hvordan du har håndtert
 - a. Innkapsling
 - b. arv og polymorfi

I rapporten, fortell også om en eller to utfordringer du opplevde underveis. Eller er det noe du har lyst å dele som du synes du har løst veldig bra? Du kan også forklare 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
- Kodekvalitet (f.eks kopling, samhörighet, innkapsling, navngivning). Det er en klar forventning at løsningen viser eksempel på innkapsling.
- JDBC (du kan altså **ikke** benytte et ORM-rammeverk i løsningen din)
- Filhåndtering

Sensor vil ta utgangspunkt i levert kildekode 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

Det nyopprettede selskapet Scrapyard med filialer i tre norske byer tar imot biler og motorsykler. De ønsker å strukturere informasjonen de har om de ulike kjøretøyene. De har laget en fil med informasjon (vehicles.txt), og denne informasjonen skal overføres til en database. For en beskrivelse av filen, se vedlegg 1.

Del 1 - importere data til database:

Du skal lage et program som leser fra filen vehicles.txt og legger dataene inn i en database. Et forslag til databasestruktur finner du i filen scrapyard.sql. Du kan velge å gjøre justeringer i scrapyard.sql, men det skal ikke være nødvendig. Det er heller ikke anbefalt å gjøre endringer i databasen da det vil stjele tid. Du trenger ikke å kjøre databasescriptet via Java. Det kan du like gjerne kjøre i MySQL Workbench, så er databasen din på plass.

Når du er ferdig med del 1 bør du ha:

- En velfungerende database som har informasjon om skraphandlerstedene (scrapyards) og kjøretøy (vehicles). Tabellene i databasen er fylt med data basert på filen vehicles.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 kjøretøy 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. Du kan velge å legge inn data direkte i databasen (utenfor Java-programmet ditt), hvis du ikke får til del 1. Skriv i så fall om dette i rapporten.

Del 2 – bruke databasen:

Du skal lage et lite program som tilbyr brukeren en meny med noen menyvalg som omhandler skraphandlerstedene og kjøretøyene 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 kjøretøy.
- Se informasjon om hvor mye drivstoff som befinner seg i fossilbilene totalt.
- Se informasjon om alle kjøretøy som er kjørbare.
- En valgfri funksjonalitet som du selv velger. Valgt funksjonalitet må medføre kommunikasjon med databasen.
- Avslutte programmet.

Husk å inkludere rapporten før du zipper og leverer.

Lykke til!

Vedlegg 1 – om filen vehicles.txt

Overordnet sett inneholder filen informasjon om filialer (de ulike fysiske skraphandlerstedene) og kjøretøyene. Det er tre ulike typer kjøretøy: fossile biler (FossilCar), elektriske biler (ElectricCar) og motorsykler (Motorcycle).

Filen starter med informasjon om skraphandlerstedene (scrapyards):

3 -> Hvor mange skraphandlersteder som er listet opp i filen.

1 -> Id til skraphandlerstedet

Oslo Scrapyard -> Skraphandlerstedets navn

Gulbrusvingen 3, Oslo -> Skraphandlerstedets adresse

+47 123 456 789 -> Skraphandlerstedets telefonnummer

--- -> Et skille mellom hvert skraphandlersted

[Flere skraphandlersteder]

[Deretter kommer informasjon om kjøretøy. Det starter med et tall som angir hvor mange kjøretøy det er informasjon om, og deretter informasjon om de ulike kjøretøyene. Det er tre forskjellige typer kjøretøy: fossile biler (FossilCar), elektriske biler (ElectricCar) og motorsykler (Motorcycle). De har en del felles egenskaper, og noen egenskaper som er forskjellige.]

1 -> Id til kjøretøyet.

1 -> Id til skraphandlerstedet kjøretøyet tilhører (altså i dette tilfellet: Oslo Scrapyard som har id=1)

FossilCar -> Hva slags kjøretøy (FossilCar, ElectricCar eller Motorcycle).

Toyota -> Merke.

Corolla -> Modell.

2010 -> Årsmodell.

AB12345 -> Registreringsnummer da kjøretøyet ble levert inn.

XYZ987654321 -> Chassisnummer.

true -> Om kjøretøyet er kjørbart (true/false).

4 -> Hvor mange hjul som er salgbare (fine nok til å bli solgt).

Petrol -> Hva slags drivstoff (**særegen egenskap for en fossilbil**)

20 -> Hvor mange liter drivstoff det er igjen på tanken (**særegen egenskap for en fossilbil**)

--- -> Det er også et slikt skille mellom de ulike kjøretøyene

[Her er et eksempel på en annen type kjøretøy – ElectricCar]:

2 -> Id til kjøretøyet.

1 -> Id til skraphandlerstedet kjøretøyet tilhører (også dette tilfellet: Oslo Scrapyard som har id=1)

ElectricCar -> Hva slags kjøretøy (FossilCar, ElectricCar eller Motorcycle).

Tesla-> Merke.

Model S -> Modell.

2013 -> Årsmodell.

CD67890 -> Registreringsnummer da kjøretøyet ble levert inn.

ABC123456789-> Chassisnummer.

true -> Om kjøretøyet er kjørbart (true/false).

4 -> Hvor mange hjul som er salgbare (fine nok til å bli solgt).

100 -> Batterikapasitet (**særegen egenskap for en elektrisk bil**)

80 -> Ladenivå (**særegen egenskap for en elektrisk bil**)

[Her er et eksempel på en annen type kjøretøy – motorsykel]:

3 -> Id til kjøretøyet.

1 -> Id til skraphandlerstedet kjøretøyet tilhører (også i dette tilfellet: Oslo Scrapyard som har id=1)

Motorcycle-> Hva slags kjøretøy (FossilCar, ElectricCar eller Motorcycle).

Harley-Davidson-> Merke.

Street 750 -> Modell.

2015 -> Årsmodell.

EF34567 -> Registreringsnummer da kjøretøyet ble levert inn.

DEF987654321-> Chassisnummer.

false -> Om kjøretøyet er kjørbart (true/false).

2 -> Hvor mange hjul som er salgbare (fine nok til å bli solgt).

false -> om motorsykkelen har sidevogn (**særegen egenskap for en motorsykel**)

750-> Motorkapasitet (**særegen egenskap for en motorsykel**)

true-> Om motorsykkelen er modifisert (**særegen egenskap for en motorsykel**)

2-> Antall hjul (**særegen egenskap for en motorsykel**)

Som du vil se vil informasjonen om kjøretøyene først beskrive egenskaper som er felles for alle kjøretøy (og informasjon om hva slags kjøretøy det er). Deretter vil det komme informasjon som er spesifikk for kjøretøyet.

Vedlegg 2 Om bruk av generativ KI på eksamen i PGR112

Eksamen i PGR112 (Objektorientert programmering) er en individuell hjemmeeksamen med alle hjelpemidler tillatt. Her følger en beskrivelse angående bruk av generativ KI på eksamen.

Eksamensoppgavene er utformet slik at det ikke skal være nødvendig å benytte generativ KI (som ai.kristiania.no og liknende) for å løse dem. Oppgavene er utformet for å teste dine kunnskaper og ferdigheter tilknyttet læringsmål i emnet. Hvis du allikevel velger å benytte generativ KI som en kilde må du referere til kilden. Ved bruk av en AI-chatbot gjøres det på følgende måte:

- Angi hva slags generativ KI som er brukt (type verktøy og versjon). For eksempel ai.kristiania.no versjon GPT-4o.
- Angi **nøyaktig** hvilke kodelinjer i besvarelsen der generativ KI er brukt som et hjelpemiddel. Dette skal fremkomme i kildekoden.
- Angi *hvordan* verktøyet er brukt som hjelpemiddel for hver enkelt oppgave det har vært i bruk. For en KI-chatbot **må** dette inkludere prompten (instruksjonen) brukt for den spesifikke oppgaven. Det **må** også inkludere svar fra KI-chatboten.

Det anbefales ikke å benytte en KI-assistent innebygget i kodeverktøyet. Kravet til kildereferanse gjelder nemlig også her. Du må dokumentere hva slags KI-assistent som er benyttet, og bruken må dokumenteres i kildekoden for hver eneste kodelinje der KI-assistenten har generert kodelinjen for deg.

Bruk av et KI-verktøy som resulterer i en fullstendig løsning av en oppgave eller deloppgave vil ikke bli regnet som fusk så lenge kildebruken er oppgitt som punktene over beskriver, men gir heller ingen uttelling.

Bruk av KI-verktøy uten henvisning til kilde vil bli oppfattet som fusk og mistanken vil bli meldt videre til Eksamenskontoret for vurdering.

En besvarelse som inkluderer en generell beskrivelse av bruk av KI på eksamen, uten å presisere for hvilke oppgaver, og på hvilken måte, vil bli vurdert til Ikke bestått, men vil ikke bli oppfattet som fusk.

Eksamen PGR112 – Nynorsk

Individuell skriftleg heimeeksamen, 24 timar

All kode må kodast av deg og deg åleine. Om du baserer koden din på eksisterande kode, må du tydeleg vise (i kommentarar) kva kjelda var. Å referere til kjelde gjeld også for bruk av KI. Vedlegg 2 viser retningslinjer for bruk av KI.

Du bør leggje alle filene som leveransen inneheld i éi mappe og zippe før du lastar opp zip-en til WISEflow. Ein leveranse som IKKJE inneheld kjeldekoden til prosjektet ditt, vil bli vurdert til Ikkje bestått.

Løysinga skal kunne køyrast i IntelliJ (dvs. treng ikkje nokon annan spesiell teknologi/IDE)

Om du er usikker på noko som gjeld eksamen, les instruksjonane på nytt og prøv å gjere det beste ut frå føresetnadene. Gje tydeleg uttrykk for kva slags føresetnader du gjorde og kvifor i rapporten (sjå nedanfor).

Leveransen vil innehalde to ting:

1. Prosjektmappa med alle relevante filer for å kunne køyre og vurdere koden du har laga.
2. Ein liten rapport (pdf- eller word-dokument, gjerne rundt 300 ord) som kort skildrar korleis du har handtert
 - a. innkapsling
 - b. arv og polymorfi

I rapporten, fortel også om ei eller to utfordringar du opplevde undervegs. Eller er det noko du har lyst til å dele som du tykkjer du har løyst veldig bra? Du kan også forklare eventuelle føresetnader du har lagt til grunn for løysinga di (om du meiner oppgåveteksten var uklår på nokre punkt).

Vurdering:

Bestått / Ikkje bestått.

For å oppnå bestått:

Kandidaten har oppnådd krava til læringsutbyte og viser nødvendige kunnskapar, ferdigheiter og kompetanse.

Emneskildringa viser læringsmåla i emnet. Blant temaa i emneskildringa vil eksamen særleg måle læringsutbyte innan følgjande tema:

- Arv og polymorfi
- Abstrakte klassar og interface (løysinga di bør innehalde eit eksempel på minst ein av dei)
- Unntakshandtering
- Kodekvalitet (f.eks. kopling, samhøyrslø, innkapsling, namngjeving). Det er ei klår forventning at løysinga viser eksempel på innkapsling.
- JDBC (du kan altså **ikkje** bruke eit ORM-rammeverk i løysinga di)
- Filhandtering

Sensor vil ta utgangspunkt i levert kjeldekode for å måle måloppnåing innanfor desse temaa. I tillegg vil sensor bruke rapporten der nokre av temaa er utdjupa. Eit bestått svar vil vise måloppnåing innan alle temaa ovanfor. Høg måloppnåing innan fleire læringsmål kan oppvege for svak måloppnåing innan eit fåtal tema. Sensor vil gjere ei totalvurdering av leveransen i vurderinga si av svaret. Det er såleis ikkje ein spesifikk prosentvis score ein må oppnå for å stå på eksamen. Det er ei totalvurdering frå sensor basert på læringsmåla som blir målte gjennom eksamen.

Oppgåva

Det nyoppretta selskapet Scrapyard med filialar i tre norske byar tar imot bilar og motorsyklar. Dei ønskjer å strukturere informasjonen dei har om dei ulike køyretøya. Dei har laga ei fil med informasjon (vehicles.txt), og denne informasjonen skal overførast til ein database. For ei skildring av fila, sjå vedlegg 1.

Del 1 - importere data til database:

Du skal lage eit program som les frå fila vehicles.txt og legg dataa inn i ein database. Eit framlegg til databasestruktur finn du i fila scrapyard.sql. Du kan velje å gjere justeringar i scrapyard.sql, men det skal ikkje vere nødvendig. Det er heller ikkje tilrådd å gjere endringar i databasen då det vil stele tid. Du treng ikkje å køyre databasescriptet via Java. Det kan du like gjerne køyre i MySQL Workbench, så er databasen din på plass.

Når du er ferdig med del 1, bør du ha:

- Ein velfungerande database som har informasjon om skraphandlarstadene (scrapyards) og køyretøy (vehicles). Tabellane i databasen er fylte med data basert på fila vehicles.txt.
- Ein eller fleire klassar som kan kommunisere med databasen.
- Ein eller fleire klassar som kan halde i data frå databasen. **Det er forventa at arv blir brukt for klassar som omfattar ulike typar køyretøy ettersom dei har mange felles eigenskapar. Det er også forventa at desse klassane blir brukte i kommunikasjon med databasen.**

OBS: Hugs at eksamenstida er avgrensa. Om du ikkje greier å kome i mål med del 1 innan rimeleg tid, så er du nøydd til å gå vidare til del 2. Ta i så fall vare på koden du skreiv for del 1, slik at han kan bli vurdert. Du kan velje å leggje inn data direkte i databasen (utanfor Java-programmet ditt), om du ikkje får til del 1. Skriv i så fall om dette i rapporten.

Del 2 – bruke databasen:

Du skal lage eit lite program som tilbyd brukaren ein meny med nokre menyval som handlar om skraphandlarstadene og køyretøya som no er plasserte i databasen (etter at del 1 er gjennomført). Du kan sjølv velje om du vil ha to main-metodar (ein for del 1 og ein for del 2), eller om du vil kommentere ut koden for del 1 når du utviklar del 2. Eller du kan velje å ha del 1 og del 2 i same program.

Programmet skal tilby følgjande funksjonalitet:

- Sjå informasjon om alle køyretøy.
- Sjå informasjon om kor mykje drivstoff som er i fossilbilane totalt.
- Sjå informasjon om alle køyretøy som er køyrbare.

- Ein valfri funksjonalitet som du sjølv vel. Vald funksjonalitet må medføre kommunikasjon med databasen.
- Avslutte programmet.

Hugs å inkludere rapporten før du zippar og leverer.

Lykke til!

Vedlegg 1 – om fila `vehicles.txt`

Overordna sett inneheld fila informasjon om filialar (dei ulike fysiske skraphandlarstadene) og køyretøya. Det er tre ulike typar køyretøy: fossile bilar (FossilCar), elektriske bilar (ElectricCar) og motorsyklar (Motorcycle).

Fila startar med informasjon om skraphandlarstadene (scrapyards):

3 -> Kor mange skraphandlarstadar som er lista opp i fila.

1 -> Id til skraphandlarstaden

Oslo Scrapyard -> Namn på skraphandlarstaden

Gulbrussvingen 3, Oslo -> Adressa til skraphandlarstaden

+47 123 456 789 -> Telefonnummeret til skraphandlarstaden

--- -> Eit skilje mellom kvar skraphandlarstad

[Fleire skraphandlarstader]

[Deretter kjem informasjon om køyretøy. Det startar med eit tal som gjev opp kor mange køyretøy det er informasjon om, og deretter informasjon om dei ulike køyretøya. Det er tre ulike typar køyretøy: fossile bilar (FossilCar), elektriske bilar (ElectricCar) og motorsyklar (Motorcycle). Dei har ein del felles eigenskapar, og nokre eigenskapar som er ulike.]

1 -> Id til køyretøyet.

1 -> Id til skraphandlarstaden køyretøyet tilhøyrer (altså i dette tilfellet: Oslo Scrapyard som har id=1)

FossilCar -> Kva slags køyretøy (FossilCar, ElectricCar eller Motorcycle).

Toyota -> Merke.

Corolla -> Modell.

2010 -> Årsmodell.

AB12345 -> Registreringsnummer då køyretøyet vart levert inn.

XYZ987654321 -> Chassisnummer.

true -> Om køyretøyet er køyrbart (true/false).

4 -> Kor mange hjul som er salbare (fine nok til å bli selde).

Petrol -> Kva slags drivstoff (**særeigen eigenskap for ein fossilbil**)

20 -> Kor mange liter drivstoff det er igjen på tanken (**særeigen eigenskap for ein fossilbil**)

--- -> Det er også eit slikt skilje mellom dei ulike køyretøya.

[Her er eit eksempel på ein annan type køyretøy – ElectricCar]:

2 -> Id til køyretøyet.

1 -> Id til skraphandlarstaden køyretøyet tilhøyrer (også dette tilfellet: Oslo Scrapyard som har id=1)

ElectricCar -> Kva slags køyretøy (FossilCar, ElectricCar eller Motorcycle).

Tesla -> Merke.

Model S -> Modell.

2013 -> Årsmodell.

CD67890 -> Registreringsnummer då køyretøyet vart levert inn.

ABC123456789 -> Chassisnummer.

true -> Om køyretøyet er køyrbart (true/false).

4 -> Kor mange hjul som er salbare (fine nok til å bli selde).

100 -> Batterikapasitet (**særegen eigenskap for ein elektrisk bil**)

80 -> Ladenivå (**særegen eigenskap for ein elektrisk bil**)

[Her er eit eksempel på ein annan type kjøretøy – motorsykkle]:

3 -> Id til køyretøyet.

1 -> Id til skraphandlarstaden køyretøyet tilhøyrer (også i dette tilfellet: Oslo Scrapyard som har id=1)

Motorcycle -> Kva slags køyretøy (FossilCar, ElectricCar eller Motorcycle).

Harley-Davidson -> Merke.

Street 750 -> Modell.

2015 -> Årsmodell.

EF34567 -> Registreringsnummer då køyretøyet vart levert inn.

DEF987654321 -> Chassisnummer.

false -> Om køyretøyet er køyrbart (true/false).

2 -> Kor mange hjul som er salbare (fine nok til å bli selde).

false -> om motorsykkelen har sidevogn (**særegen eigenskap for ein motorsykkle**)

750 -> Motorkapasitet (**særegen eigenskap for ein motorsykkle**)

true -> Om motorsykkelen er modifisert (**særegen eigenskap for ein motorsykkle**)

2 -> Antal hjul (**særegen eigenskap for ein motorsykkle**)

Som du vil sjå vil informasjonen om køyretøya først skildre eigenskapar som er felles for alle køyretøy (og informasjon om kva slags køyretøy det er). Deretter vil det kome informasjon som er spesifikk for køyretøyet.

Vedlegg 2 Om bruk av generativ KI på eksamen i PGR112

Eksamen i PGR112 (Objektorientert programmering) er ein individuell heimeeksamen med alle hjelpemiddel tillatne. Her følgjer ei skildring som gjeld bruk av generativ KI på eksamen.

Eksamensoppgåvene er utforma slik at det ikkje skal vere nødvendig å bruke generativ KI (som ai.kristiania.no og liknande) for å løyse dei. Oppgåvene er utforma for å teste kunnskapane og ferdigheitene dine knytte til læringsmål i emnet. Om du likevel vel å bruke generativ KI som ei kjelde må du referere til kjelda. Ved bruk av ein KI-chatbot gjer du det på følgjande måte:

- Gje opp kva slags generativ KI som er brukt (type verktøy og versjon). For eksempel ai.kristiania.no versjon GPT-4o.
- Gje opp **nøyaktig** kva slags kodelinjer i svaret ditt der generativ KI er brukt som eit hjelpemiddel. Dette skal kome fram i kjeldekoden.
- Gje opp *korleis* verktøyet er brukt som hjelpemiddel for kvar enkelt oppgåve der det har vore i bruk. For ein KI-chatbot **må** dette inkludere prompten (instruksjonen) brukt for den spesifikke oppgåva. Dette **må** også inkludere svar frå KI-chatboten.

Det er ikkje tilrådd å bruke ein KI-assistent bygd inn i kodeverktøyet. Kravet til kjeldereferanse gjeld nemleg også her. Du må dokumentere kva slags KI-assistent som er brukt, og bruken må dokumenterast i kjeldekoden for kvar einaste kodelinje der KI-assistenten har generert kodelinja for deg.

Bruk av eit KI-verktøy som resulterer i ei fullstendig løysing av ei oppgåve eller deloppgåve vil ikkje bli rekna som fusk så lenge kjeldebruken er gjeven opp som punkta over skildrar, men gjev heller inga utteljing.

Bruk av KI-verktøy utan tilvising til kjelde vil bli oppfatta som fusk og mistanken vil bli meld vidare til Eksamenskontoret for vurdering.

Eit svar som inkluderer ei generell skildring av bruk av KI på eksamen, utan å presisere for kva slags oppgåver, og på kva måte, vil bli vurdert til Ikkje bestått, men vil ikkje bli oppfatta som fusk.