

Emnekode: OBJ2100B	Emnenavn: Objektorientert Programmering 2	
Ansvarlig faglærer: Tor Lønnestad	Campus: Bø	Fakultet: Handelshøgskolen
Utleiv. dato og tidspunkt i WISEflow: 13.05.2024 kl. 09.00		Innlev. dato og tidspunkt i WISEflow: 16.05.2024 kl. 14.00
Antall oppgaver: 1	Antall vedlegg: 0	Ant. sider inkl. forside og vedlegg: 1 + 3 + 3
Hjelpemidler og samarbeid: Tillatte hjelpemidler: Alle hjelpemidler er tillatt. Eksamensform (kryss av): Individuell <input type="checkbox"/> Gruppe <input checked="" type="checkbox"/> Følgende samarbeid er tillatt både for individuell og gruppeeksamen: Det er tillatt å samarbeide med andre studenter/grupper om forståelse av oppgavene o.l., men eksamensbesvarelsen skal vise din/gruppens individuelle prestasjon. Tekst, fremgangsmåte for oppgaveløsning og faglige refleksjoner osv., skal være ditt/gruppens eget arbeid og derfor ikke utveksles. Tekstlikhet i besvarelser kan føre til mistanke om fusk.		
Kildehenvisning (dersom aktuelt): Vær tydelig, unngå setninger som f.eks. «som fortalt på forelesninger» osv. <ol style="list-style-type: none">1. Dere skal henvise løpende i teksten til alle kilder dere bruker.2. Dersom dere bruker egne, tidligere arbeider skal dere oppgi disse som kilde.3. Direkte sitater i teksten må markeres med anførselstegn, etterfulgt av kildehenvisning.4. Litteraturlisten skal ligge til slutt i besvarelsen og skal inneholde alle kildene som er brukt. Husk: Alle kilder som finnes i teksten skal også finnes i litteraturlisten og motsatt. Andre viktige opplysninger: <ol style="list-style-type: none">1. Programfiler (kildekode) skal leveres på tekstformat, ikke PDF/Word/bildeformater eller lignende.2. Dokumenter som ikke er programfiler skal leveres på PDF-format.3. Bildefiler skal være på JPEG-format. Alle besvarelser blir kontrollert for plagiat/ulovlig samarbeid/fusk.		

KANDIDATEN MÅ SELV KONTROLLERE AT OPPGAVESETTET ER FULLSTENDIG

Råd og retningslinjer. Les hele oppgaveteksten godt før dere går i gang med å løse oppgaven. Planlegg tidsbruken og fordel oppgaver dere imellom. Dersom dere mener noe ved oppgaveteksten er upresist eller uklart, så beskriv hvordan dere tolker oppgaven. Lag de klassene og metodene dere måtte ønske, men slik at det er i samsvar med oppgaveteksten. All programkode skal være i Java. Alt av grafisk brukergrensesnitt skal være basert på JavaFX, men det er tillatt å bruke FXML. Husk at all kode dere leverer skal være egenutviklet, uten å låne/hente kode fra andre. Det er heller ikke tillatt å generere kode med chatGPT eller lignende tjenester.

Introduksjon

I større organisasjoner, både private og offentlige, er det en utfordring å holde oversikt over alt inventar som er kjøpt inn. Eksempler på inventar er møbler, utsmykning og teknisk utstyr. Gruppa skal utvikle et system som bidrar til å holde denne oversikten.

Systemet skal bestå av to programmer, et klientprogram og et tjenerprogram. Brukere av systemet kjører klientprogrammet på egen maskin, dette kommuniserer med tjenerprogrammet ved hjelp av sockets, og tjenerprogrammet benytter en database til permanent lagring. Klientprogrammet skal ha grafisk brukergrensesnitt, tjenerprogrammet kan skrive eventuelle feilmeldinger til en loggfil.

Funksjonalitet

Bruker skal kunne få utført en rekke funksjoner. I grove trekk er det:

- Opprette et nytt element i inventar-databasen.
- Gjøre søk i, og få ut informasjon fra databasen.
- Etter at det er gjort søk, og søkeresultatet er presentert for bruker, skal bruker kunne velge ut et element til videre behandling.
- Videre behandling kan være å slette elementet, men det kan også være å endre informasjon i elementet.

Inventarelementer

Vi skiller mellom de tre inventartypene nevnt i innledningen: møbler, utsmykning og teknisk utstyr. Følgende skal som minimum lagres:

- Om møbler: Kategori (valg mellom faste alternativ som bord/stol/sofa/skap/hylle/tavle/annet), beskrivelse (fritekst), innkjøpsdato, innkjøpspris, forventet levetid (antall år), plassering (nærmere beskrevet senere).
- Om utsmykning: Kategori (valg mellom faste alternativ som maleri/grafikk/tekstil/bilde/skulptur/annet), beskrivelse (fritekst), innkjøpsdato, innkjøpspris, plassering.
- Om teknisk utstyr: Kategori (fritekst), beskrivelse (fritekst), innkjøpsdato, innkjøpspris plassering.

Dessuten skal det være mulig å registrere et antall, slik at eksempelvis innkjøp av 100 like stoler opptrer som ett element, ikke 100 elementer.

Plassering skal angis med et hierarkisk system: bygg/fløy/etasje/rom, adskilt med «/» som vist her. Auditorium 5-118 vil da representeres som «USN-Bø/5/1/18». Plassering skal også kunne registreres med en startdel av dette, med eksempelvis bare «USN-Bø/5» som tilsier at etasje og romnummer ikke er fastsatt.

Når inventar tas ut av bruk, skal det for alle inventartyper registreres:

- Dato tatt ut av bruk
- Årsak til at den er tatt ut av bruk: solgt/kassert/på lager/annet

Opprette inventarelement

Brukeren må kunne opprette inventarelementer med alle opplysningene drøftet over, og få dette lagt inn i databasen. Det er naturlig at man først oppgir inventartype, og at GUI-et deretter tilpasser seg aktuell type inventar.

Søke etter inventarelement

Det skal være mulig å søke på følgende opplysninger i databasen:

- Inventartype. For eksempel: Utsmykning
- Beskrivelse. For eksempel: Lenestol. Her bør det gi match dersom søkestrengen finnes som en del av beskrivelsen, selv om beskrivelsen er lengre
- Innkjøpsdato. Her holder det med å angi årstall
- Kategori. For eksempel: Grafikk
- Innkjøpspris. Angis som et intervall, for eksempel: 10000 – 20000 kr
- Forventet levetid (regnet fra innkjøpsdato). Angis som et intervall, for eksempel: 5 – 7 år.
- Man skal også kunne søke på forventet år for kassering, uansett hvor lenge elementet da har vært i bruk. Dette beregnes ut ifra feltene for dato tatt i bruk og forventet levetid. Angis som et intervall, for eksempel: 2025-2027
- Plassering. Bruker skal kunne oppgi hele eller bare en startdel av dette, eksempelvis «USN-Bø/5» som vil få treff på alt registrert i fløy 5.
- Antall, angitt som et intervall, for eksempel: 50 – 70
- Alle elementer som er tatt ut av bruk/som er i bruk
- Alle elementer som er tatt ut av bruk av en spesiell årsak (solgt/kassert/lagt på lager/...)
- Alle elementer som er tatt ut av bruk på et spesielt tidspunkt. Her holder det å angi årstall

Listen over inneholder 12 punkter. **Gruppen skal implementere minst seks av disse**, valgfritt hvilke, men minst ett av intervalltype og ett med faste alternativ. De valgte søkekriteriene skal kunne kombineres, eksempelvis Beskrivelse: lenestol, Innkjøpspris: 10000-20000, Plassering: USN-Bø/5.

Brukerens søkekriterier må oversendes tjeneren. Merk at det så er to måter å implementere søket på. Den ene er å «oversette» hele søket til SQL slik at man direkte får ut relevante poster fra databasen. Den andre måten er å hente ut større mengder data fra databasen over til en datastruktur, som man så kan traversere i Java-koden. Disse to teknikkene kan gjerne kombineres. Søk i databasen må utføres i tjeneren. Søk i datastruktur kan i prinsippet utføres i tjener eller klient, men bør helst utføres i tjener for å redusere socket-kommunikasjonen. Søkeresultatet overføres uansett metode til klienten og presenteres for brukeren.

Redigere/slette inventarelement

Basert på søkeresultat skal bruker kunne velge et element, og redigere eller slette dette. Alle opplysninger skal kunne endres, herunder opplysninger om at elementet er tatt ut av bruk.

Innlogging

Denne første versjonen av systemet inneholder ikke innlogging/brukerautentisering.

Database

Vurdér databasestrukturen, og beskriv den i rapporten som skal leveres. Det bør benyttes en SQLite-database. Dersom databasen ikke eksisterer i det tjeneren starter, skal den opprettes automatisk.

Leveranse

Skriv lett forståelig og lesbar kode, og bruk gode navn på klasser, metoder, variabler og konstanter. Bruk Javadoc til å dokumentere klasser og metoder (i det minste public metoder), med minimum taggene @param og @return. (Se f.eks. <https://www.oracle.com/technical-resources/articles/java/javadoc-tool.html>) Dere trenger ikke generere HTML-filer fra Javadoc. Dersom dere bruker et IDE kan gjerne hele prosjektkatalogen leveres. I tillegg til all programkode dere utvikler, skal det også leveres en rapport i PDF-format der dere viser med skjermdumper, og kort beskriver, klientprogrammet i ulike relevante situasjoner. I den samme rapporten skal dere beskrive databasestrukturen og tjenerapplikasjonen. Skriv også litt om hva systemet eventuelt mangler i forhold til oppgaveteksten. Levér PDF-rapporten som hoveddokument i WISEflow/FLOWassign, alt annet pakkes sammen i én zip-fil og leveres som et vedlegg.

Bruk god tid på å tenke og planlegge og samkjøre gruppa før dere starter å kode, det vil dere trolig spare mye tid på totalt sett. Planlegg klasser, klassehierarki, databasestruktur og kommunikasjonsmønster mellom klient og tjener. Skriv også ned valgene og vurderingene som gjøres underveis. Dette vil være et godt utgangspunkt for rapporten som skal leveres.

Oppgaven skal løses av gruppe på 2-5 personer.

Lykke til!

Råd og retningslinjer. Les heile oppgåveteksten godt før de går i gang med å løyse oppgåva. Planlegg tidsbruken og fordel oppgåver imellom dykk. Dersom de meiner noko ved oppgåveteksten er upresist eller uklårt, så skriv korleis de tolkar oppgåva. Lag dei klassene og metodane de måtte ynskje, men slik at det er i samsvar med oppgåveteksten. All programkode skal vera i Java. Alt av grafisk brukargrensesnitt skal vera basert på JavaFX, men det er tillate å bruke FXML. Hugs at all kode de leverer skal vera eigenutvikla, utan å låne/hente kode frå andre. Det er heller ikkje tillate å generere kode med chatGPT eller liknande tenester.

Introduksjon

I større organisasjonar, både private og offentlege, er det ei utfordring å halde oversikt over alt inventar som er kjøpt inn. Døme på inventar er møblar, utsmykking og teknisk utstyr. Gruppa skal utvikle eit system som bidreg til å halde denne oversikta.

Systemet skal bestå av to program, eit klientprogram og eit tenarprogram. Brukarar av systemet køyrer klientprogrammet på eigen maskin, dette kommuniserer med tenarprogrammet ved hjelp av sockets, og tenarprogrammet nyttar ein database til permanent lagring. Klientprogrammet skal ha grafisk brukargrensesnitt, tenarprogrammet kan skrive eventuelle feilmeldingar til ei loggfil.

Funksjonalitet

Brukar skal kunne få utført ei rekke funksjonar. I grove trekk er det:

- Opprette eit nytt element i inventar-databasen.
- Gjere søk i, og få ut informasjon frå databasen.
- Etter at det er gjort søk, og søkeresultatet er presentert for brukar, skal brukar kunne velge ut eit element til vidare handsaming.
- Vidare handsaming kan vera å slette elementet, men det kan også vera å endre informasjon i elementet.

Inventarelement

Vi skil mellom dei tre inventartypene nemnt i innleiinga: møblar, utsmykking og teknisk utstyr. Følgjande skal som minimum lagrast:

- Om møblar: Kategori (val mellom faste alternativ som bord/stol/sofa/skap/hylle/tavle/anna), omtale (fritekst), innkjøpsdato, innkjøpspris, forventa levetid (antal år), plassering (nærare beskrive seinare).
- Om utsmykking: Kategori (val mellom faste alternativ som maleri/grafikk/tekstil/bilde/skulptur/anna), omtale (fritekst), innkjøpsdato, innkjøpspris, plassering.
- Om teknisk utstyr: Kategori (fritekst), omtale (fritekst), innkjøpsdato, innkjøpspris, plassering.

Dessutan skal det vera mogeleg å registrere eit antal, slik at til dømes innkjøp av 100 like stolar opptre som eitt element, ikkje 100 element.

Plassering skal ein gje med eit hierarkisk system: bygg/fløy/etasje/rom, skilt med «/» som vist her. Auditorium 5-118 vil da bli representert som «USN-Bø/5/1/18». Plassering skal også kunne registrerast med ein startdel av dette, til dømes berre «USN-Bø/5» som tyder at etasje og romnummer ikkje er fastsett.

Når inventar vert teken ut av bruk, skal det for alle inventartypar registrerast:

- Dato tatt ut av bruk
- Årsak til at den er tatt ut av bruk: seld/kassert/på lager/anna

Opprette inventarelement

Brukaren må kunne opprette inventarelement med alle opplysningane drøfta over, og få dette lagt inn i databasen. Det er naturleg at ein fyrst gjev inventartype, og at GUI-et deretter tilpassar seg aktuell type inventar.

Søke etter inventarelement

Det skal vera mogeleg å søke på fylgjande opplysningar i databasen:

- Inventartype. Døme: Utsmykking
- Omtale. Døme: Lenestol. Her bør det gje match dersom søkestrengen finst som ein del av omtalen, sjølv om omtalen er lengre
- Innkjøpsdato. Her held det med å gje årstal
- Kategori. Døme: Grafikk
- Innkjøpspris. Gjevast som eit intervall, døme: 10000 – 20000 kr
- Forventa levetid (rekna frå innkjøpsdato). Gjevast som eit intervall, døme: 5 – 7 år.
- Ein skal også kunne søke på forventa år for kassering, same kor lenge elementet da har vore i bruk. Dette reknast ut ifrå felte for dato tatt i bruk og forventa levetid. Gjevast som eit intervall, døme: 2025-2027
- Plassering. Brukar skal kunne gje heile eller berre ein startdel av dette, døme: «USN-Bø/5» som vil få treff på alt registrert i fløy 5.
- Antal, gjeve som eit intervall, døme: 50 – 70
- Alle element som er tatt ut av bruk/som er i bruk
- Alle element som er tatt ut av bruk av ein spesiell årsak (seld/kassert/lagt på lager/...)
- Alle element som er tatt ut av bruk på eit spesielt tidspunkt. Her held det å gje årstal

Lista over inneheld 12 punkt. **Gruppa skal implementere minst seks av desse**, valfritt kva for nokre, men minst eitt av intervalltype og eitt med faste alternativ. Dei valde søkekriteriane skal kunne kombinerast, til dømes Beskrivelse: lenestol, Innkjøpspris: 10000-20000, Plassering: USN-Bø/5.

Brukaren sine søkekriterium må sendast til tenaren. Merk at det så er to måtar å implementere søket på. Den eine er å «oversetje» heile søket til SQL slik at ein direkte får ut relevante postar frå databasen. Den andre måten er å hente ut større mengder data frå databasen over til ein datastruktur, som ein deretter kan traversere i Java-koden. Desse to teknikkane kan gjerne kombinerast. Søk i databasen må utførast i tenaren. Søk i datastruktur kan i prinsippet utførast i tenar eller klient, men bør helst utførast i tenar for å redusere socket-kommunikasjonen. Uansett metode vert søkeresultatet overført til klienten og presentert for brukaren.

Redigere/slette inventarelement

På grunnlag av søkeresultat skal brukar kunne velgje eit element, og redigere eller slette dette. Alle opplysningar skal kunne endrast, også opplysningar om at elementet er tatt ut av bruk.

Innlogging

Denne fyrste versjonen av systemet inneheld ikkje innlogging/brukarautentisering.

Database

Vurdér databasestrukturen, og omtal den i rapporten som skal leverast. Det bør nyttast en SQLite-database. Dersom databasen ikkje eksisterer i det tenaren startar, skal den opprettast automatisk.

Leveranse

Skriv kode som er lett å lesa og lett å forstå, og nytt gode namn på klasser, metodar, variablar og konstantar. Bruk Javadoc til å dokumentere klasser og metodar (i det minste public metodar), med minimum taggane @param og @return. (Sjå t.d. <https://www.oracle.com/technical-resources/articles/java/javadoc-tool.html>) De treng ikkje generere HTML-filer frå Javadoc. Dersom de nyttar eit IDE kan gjerne heile prosjektkatalogen leverast. I tillegg til all programkode de utviklar, skal det også leverast ein rapport i PDF-format der de viser med skjermdumpar, og kort omtalar, klientprogrammet i ulike relevante situasjonar. I den same rapporten skal de omtale databasestrukturen og tenarapplikasjonen. Skriv også litt om kva systemet eventuelt manglar i høve til oppgåveteksten. Levér PDF-rapporten som hovuddokument i WISEflow/FLOWassign, alt anna pakkast saman i ei zip-fil og leverast som eit vedlegg.

Nytt god tid på å tenke og planlegge og samkøyre gruppa før de startar å kode, det vil de truleg spare mykje tid på totalt sett. Planlegg klasser, klassehierarki, databasestruktur og kommunikasjonsmønster mellom klient og tenar. Skriv også ned dei val og vurderingar som gjerast undervegs. Dette vil vera eit godt utgangspunkt for rapporten som skal leverast.

Oppgåva skal løysast av gruppe på 2-5 personar.

Lykke til!