

Bike rental system



Alle skriftlige materialer, pc'er, bærbare computere og internet ressourcer er tilladt til denne obligatoriske aktivitet.

Mobiltelefoner og kommunikation med andre personer (inkl. chatbots), bortset fra kommunikation med Underviserne, er forbudt.

Du må ikke gemme dine løsninger på eksterne netværksdrev/værter som GitHub, Facebook o. lign.

Ved denne obligatoriske aktivitet afslutning skal du lægge din løsning op i Wiseflow.

Denne obligatoriske aktivitet varer 4 timer og efterfølges af 1 times evaluering.

Eksamenssættet består af 9 opgaver.

Ud over disse opgaver, kan du kan blive bedt at besvare nogle ekstra spørgsmål omkring dine svar og eventuelle andre valg.

Kontroller at dette sæt indeholder 4 sider inklusive forsiden.

Opgave Beskrivelse

Zealand - Sjællands Erhvervsakademi ønsker at støtte den grønne omstilling. De har derfor anskaffet flere cykler, der kan blive lejet ud til de studerende. Således ønsker Zealand at medvirke til at reducere CO2-udledningen og efterkomme FN's verdens mål. For at lette administrationen af leje af cykler har Zealand brug for et cykeludlejningssystem (**BikeRentalSystem**).

Ved at bruge systemet skal det være muligt at administrere udlejning af cykler til de studerende. I systemet skal det være muligt at oprette og administrere cykler, der kan udlejes (CRUD på cykler). For hver cykel skal **anskaffelsesår** (yearOfPurchase), **modellen**, **stelfarven** (frame color) og **stelnummer** (chassisnumber) registreres

Det skal også være muligt at administrere oplysninger om cykeludlejning. I forbindelse med udlejen skal **startdato** og **slutdato** registreres, ligesom **start** og **slut kilometertal** skal registreres.

Hver lejer skal registreres med **navn**, **adresse**, **telefonnummer** og **studerende nummer** (entydigt). Det skal være muligt at administrere lejere (CRUD lejere) i systemet.

Første udkast af domæne modellen ser således ud:

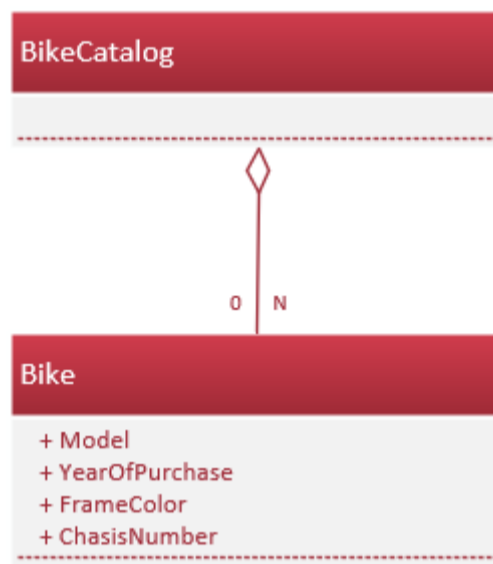


Fig.1 First draft of the domain model

Første udkast af design klasse diagrammet ser således ud:

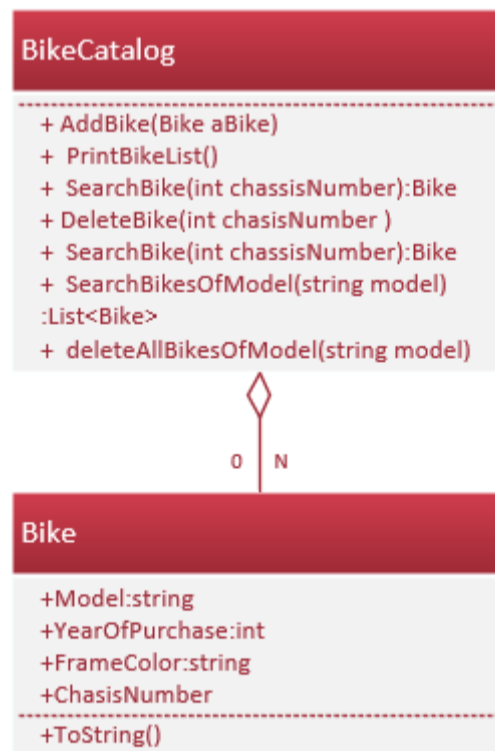


Fig.2 First draft of the design class diagram.

Opgave 1. User stories (SWD)

Skriv 3 user stories med acceptance criteria for BikeRentalSystem'et.

Opgave 2. Domain model (SWD)

Udvid den første domæne model i fig. 1. Du skal modellere den centrale del af BikeRentalSystem'et. Modellen bør højst indeholde 6 konceptuelle klasser.

Du må selv vælge hvilket tegne værktøj du vil benytte (Visio, DrawIO eller papir).

Opgave 3. Design klasse diagram (SWD)

Udvid det første design klasse diagram i fig. 2. Tegn et nyt design klasse diagram baseret på din domæne model du lavede i opgave 2. Du må selv vælge hvilket tegne værktøj du vil benytte (Visio, DrawIO eller papir).

Opgave 4. Opret en console application og Implementer klassen Bike (SWC)

Opret en ny console application (.net core) for **BikeRentalSystem'et**. I projektet skal du implementere klassen **Bike** i overensstemmelse med din klasse i design klasse diagrammet.

- Det skal være muligt at hente og sætte værdier (get og set) for properties **Model**, **YearOfPurchase** og **FrameColor**.
- Det skal være muligt at hente værdien (get) for **ChassisNumber**, men det skal kun være muligt at sætte værdien (set) ChassisNumber gennem konstruktøren.
- Implementer en konstruktør der kan initialisere alle instans felterne i objektet.
- Implementer en **ToString** metode i klassen. Den skal returnere en *string* med information om alle instans felterne i objektet.
- Opret 3 instanser af klassen Bike i Main metoden (eller tilhørende worker klasse) og udskriv dem i Console-window.

Opgave 5. Implementer klassen BikeCatalog (SWC)

Klassen BikeCatalog indeholder en *collection* af Bike objekter som skal håndteres. Du vælger om det skal være i en *List* eller en *Dictionary*. I BikeCatalog må der kun være én Bike med det samme **ChassisNumber** i collection, dvs. **ChassisNumber** er unikt.

- Opret en **konstruktør** for klassen BikeCatalog
 - Benyt konstruktøren til at initialisere den valgte collection.
- **public void AddBike(Bike aBike)**
 - Denne metode tilføjer et enkelt Bike objekt til collection'en.
 - I Main metoden, Opret en instans af BikeCatalog, samt tilføj de 3 objekter du lavede i Opgave 4.
- **public void PrintBikeList()**
 - Denne metode skal udskrive alle Bike objekterne i collection'en til et console window.
 - Hver Bike objekt skal udskrives på hver sin linje.
 - Til slut skal metoden udskrive antallet af Bike objekter i collection'en og den aktuelle dato og tid.
 - Afprøv din metode i Main metoden.
- **public Bike SearchBike(int chassisNumber)**
 - Denne metode returnerer et Bike objekt svarende til det angivne ChassisNumber i collection'en.
 - Hvis der ikke findes et Bike objekt med det angivne ChassisNumber returneres null.
 - Afprøv din metode i Main metoden.

Opgave 6. Exceptions (SWC)

Alle Bike objekter i Zealand's BikesSystem skal have unikke Chassis number. Hvis administrationen prøver at tilføje et bike objekt med et ChassisNumber der allerede findes i collection'en er det en fejl.

- Ændre implementeringen af BikeCatalog, således at der kastes (throws) en *ArgumentException* hvis du prøver at tilføje et bike objekt med et ChassisNumber, som allerede findes i BikeCatalog.
- Tilføj kode i Main metoden til at håndtere (catch) at *ArgumentException* kan opstå.
- Prøv at tilføje to Bike objekter med samme ChassisNumber og udskriv en passende fejlmeddelelse i console window.

Opgave 7. Implementering af inheritance (SWC & SWD)

Zealand ønsker tillige at udleje elektroniske bikes (EBike). En EBIke **is-a** Bike. EBIke skal have en ekstra property (i forhold til Bike) til at angive motoren **Motor** fx. 750 W.

- Implementer klassen EBIke.
 - Den skal have en property **Motor**, en konstruktør og en ToString metode.
 - Det skal være muligt at initialiserer alle properties gennem konstruktøren.
 - ToString metoden skal returnere en *string* med information om EBIke objektet.
- Opdater design klasse diagrammet med EBIke klassen (SWD).
- Afprøv din implementering i Main metoden:
 - Opret 2 instanser af klassen EBIke.
 - Tilføj de 2 objekter til BikeCatalog.
 - Udskriv alle Bike objekter fra BikeCatalog ved brug af PrintBikeList() metoden i BikeCatalog.
- Forklar med dine egne ord hvad der forstås ved begrebet "inheritance" (arv).

Opgave 8. Udvid BikeCatalog (SWC)

Implementer følgende metoder:

- **public void DeleteBike(int chasisNumber)**
 - Denne metode sletter et Bike objekt fra collection'en med det angivne chasis number. Hvis dette ikke findes bliver ingen objekter slettet.
- **public List<Bike> SearchBikesOfModel(string model)**
 - Denne metode søger igennem Bike objekter collection'en og finder alle dem der svarer til den angivne model.
 - Metoden returnerer en Liste af alle de Bike, der er den angivne model.
 - Hvis der ikke findes nogen Bike objekter med den angivne model returneres en tom liste.
- **public void deleteAllBikesOfModel(string model)**
 - Denne metode sletter alle Bike objekter af den angivne model.

Opgave 9. Implementering af klasse diagram(SWC)

- Implementer klasserne fra dit design klasse diagram, du lavede i opgave 3.