

# Indledning:

**Mål:**

* at kunne arbejde med objektorienteret programmering (OOP) i C# herunder klasser og metoder, analysere og oversætte virkelighed (OOA), benytte de datastrukturer som er relevante for løsning af opgaven og at kunne arbejde sammen om en programmeringsopgave.
* at opnå et bredt kendskab til C#
* Udvide kendskabet til og bruge database normalisering, SQL, relationer, tabeller etc.
* at kunne lave færdig statisk klasse med databasekommunikation der kan bruges på de følgende moduler.
* at kunne arbejde med integration mellem database SQL og C#
* at kunne planlægge og følge op på sit arbejde. At være bevidst omkring forskellige løsninger og deres fordele og ulemper. At kunne diskutere på et fagligt niveau.
* at kunne formidle viden mundtligt og visuelt
* at afprøve versionsstyring og samarbejde om fælles løsninger

# Forudsætninger:

Kendskab til SQL, design af database, tabeller og relationer, C# grundlæggende begreber og kodestrukturer og repræsentation (klasser, metoder, arv, indkapsling, lister, overload, override osv.)

# Arbejdsform:

Gruppe med 2 max 3 medlemmer

# Støtte og input undervejs:

læreren vil, hvis det er nødvendigt, komme med relevante oplæg omkring generelle problemstillinger. Ud over det, skal I bruge tidligere kode I har lavet, Google og hinanden.

# Evaluering:

I skal fremlægge løsningen over for læreren torsdag og fredag i anden uge af forløbet. Den eksakte fremlæggelsesform vil blive gennemgået.

# Opgaveoplæg

**Sydvest-Bo** er et mindre firma, der udlejer **sommerhuse** og lejligheder.

Sydvest-bo ejer ikke selv ferieboligerne, men indgår aftaler med **private sommerhusejere** og større lejlighedskomplekser.

Sydvest-Bo har inddelt deres distrikt i **områder**. Til hvert område er der knyttet en ansvarlig **udlejningskonsulent**, som tager sig af kommunikationen med sommerhusejerne.

Udlejningskonsulenterne skal blandt andet godkende husene, **klassificere** dem og sørge for udlejningskontrakterne med ejerne.

Når det drejer sig om ferielejligheder, forhandles udlejningskontrakterne med lejlighedskompleksets inspektør. Udlejningskontrakterne gælder hele komplekset. Lejlighederne er kategoriseret i lejlighedstyper. Typerne angiver størrelse, antal senge, kvalitet m.m.

Udlejningskonsulenterne skal også sørge for, at der bliver engageret **opsynsmænd**. En opsynsmand er en løst tilknyttet person, der mod timebetaling påtager sig aflæse strøm og efterse huse om lørdagen. Det er kun sommerhuse, der har tilknyttet en opsynsmand. Lejlighederne efterses og aflæses af inspektørerne.

Kunderne henvender sig telefonisk, pr. mail eller personligt for at indgå et lejemål af et af sommerhus eller lejlighed. De bliver betjent af kundekonsulenterne, som har ansvar for lejemålene. Der **udlejes kun i hele uger** – fra lørdag til lørdag. (denne politik kan firmaet ikke forestille sig at ændre, da omkostningerne så vil stige uforholdsmæssigt.) Udlejningsenhederne er således en uge, men **lejemålet kan** selvsagt **dække flere uger**.

**Priserne** afhænger både af **boligernes standard** og af **sæsonkategori**. Der er i øjeblikket 4 sæsonkategorier: super, høj. mellem og lav. En sæsonkategori dækker nogle ganske bestemte uger. F.eks. dækker supersæson ugerne 28, 29, 30 og 52. Hver bolig har altså tilknyttet 4 priser.

**Der er følgende ønsker til systemet:**

1. Man skal kunne oprette, rette og slette en sommerhusejere
2. Man skal kunne oprette, rette og slette et sommerhus
3. Man skal kunne oprette, rette og slette en reservation af sommerhus
4. Man skal kunne få listet et sommerhus reservationer, samt få vist en specifikation af den enkelte udlejning.
5. En ejer kan godt have flere sommerhuse
6. Man skal kunne administrere ”områder”
7. Man skal kunne administrere ”sæsonkategori og tilhørende priser”
8. Man skal kunne administrere Inspektører

**Fremtid for systemet**

Det skal være muligt at udbygge dette system, så lejlighedskomplekser også bliver en del af systemet, som I kan arbejde videre med og udbygge på efterfølgende skoleophold. Samtidig skal I være opmærksom på, at der måske er elementer i beskrivelsen, der ikke er en del af ovennævnte ønsker, det kunne være en fordel hvis jeres løsning, i størst mulig omfang, er forberedt for disse eventuelle fremtidige udvidelser.

**TIP!**

I skal arbejde objektorienteret, dvs. en kunde er et objekt, et sommerhus er et objekt osv. Som regel siger man at navneord er objekter og udsagnsord giver os metoderne. (En menu kan også være et objekt, med metoder til at oprette nyt punkt etc.)

Al tilgang til databasen skal ske gennem en statisk klasse som kan håndtere databasekald som oprette, viser, (redigere) og slette poster (CRUD).

I koder i C# Console programmering. Selve databasen (etablering, tabeller, relationer, felter, navne definitioner) må gerne oprettes uden for C# miljøet i f.eks. SQL server management Studio.

**Forslag til overordnet arbejdsplan**:

* Plan over hvad I skal lave de enkelte dage.
* Design og etablering af databasen med testdata.
* Kodning og test af statisk klasse til databaseadministration.
* Analyse hvilke objekter og metoder skal benyttes, brug UML diagrammer hvor det giver mening. Hvilke klasser, hvilken beskyttelse (private osv.)
* Hvilke skærmbilleder får I brug for (brugergrænseflade), Wireframe?
* Hvilke datastrukturer skal benyttes (f.eks. lister, arrays osv.)
* Kodning i C#.
* Test og rettelser (kan systemet nemt udbygges med de foreliggende strukturer)
* Fremlæggelse (demo af system og forklaring af valgte løsninger). Brug diagrammer, C# kode, foto, video etc.

**Følgende skal afleveres på ItsLearning onsdag i uge 2:**

* + Overordnet tidsplan
  + Logbog over daglige aktiviteter og udfordringer
  + UML diagrammer, f.eks. class diagram, E/R diagrammer
  + Kildekode – C# kode, sql-script
  + Test dokumentation der viser at I har opfyldt væsentlige krav, og som viser at de er teste af, både ’negativ’ og ’positiv’ testforløb.