



MALMÖ HÖGSKOLA

Malmö högskola

Programmering med C#, grundkurs

Övning – Mäklarna
Hitta objekt
Frivillig – redovisas ej

[Farid Naisan](#)

University Lecturer
Department of Computer Science

Övning – Arv

1. Målsättning

- 1.1. att identifiera objekt, attribut och operationer
- 1.2. att bestämma relationer mellan objekten.

2. Beskrivning

Ett mäklarbolag vill ha ett program för hantering av de fastigheter de förmedlar. Till att börja med, vill bolaget kunna registrera de fastighetsobjekt som kommer in till byråen. Deras fastigheter består av både bostäder och kommersiella lokaler. Bostäder kan vara av typ villa, radhus, kedjehus och bostadsrätter, fritidshus, medan kommersiella lokaler är kontorslokaler, butik och lager. De har också en del hyresverksamhet som omfattar endast lyxiga lägenheter.

Varje fastighet har en adress, ett objektnummer och en fastighetsbeteckning och ett pris.

Gör en analys av detta problem och rita ett klassdiagram där relationerna bestäms med hjälp av "is a" och "has a" relationer. Du kan använda Visual Studio för att rita klassdiagrammet. Klasserna behöver inte programmeras. Ta god tid på dig innan du jämför med den förslagna lösningen.

3. Vägledning

Utveckling av en mjukvara sker i olika faser. I varje fas produceras artefakter. Det är minst två steg före programmering av ett program.

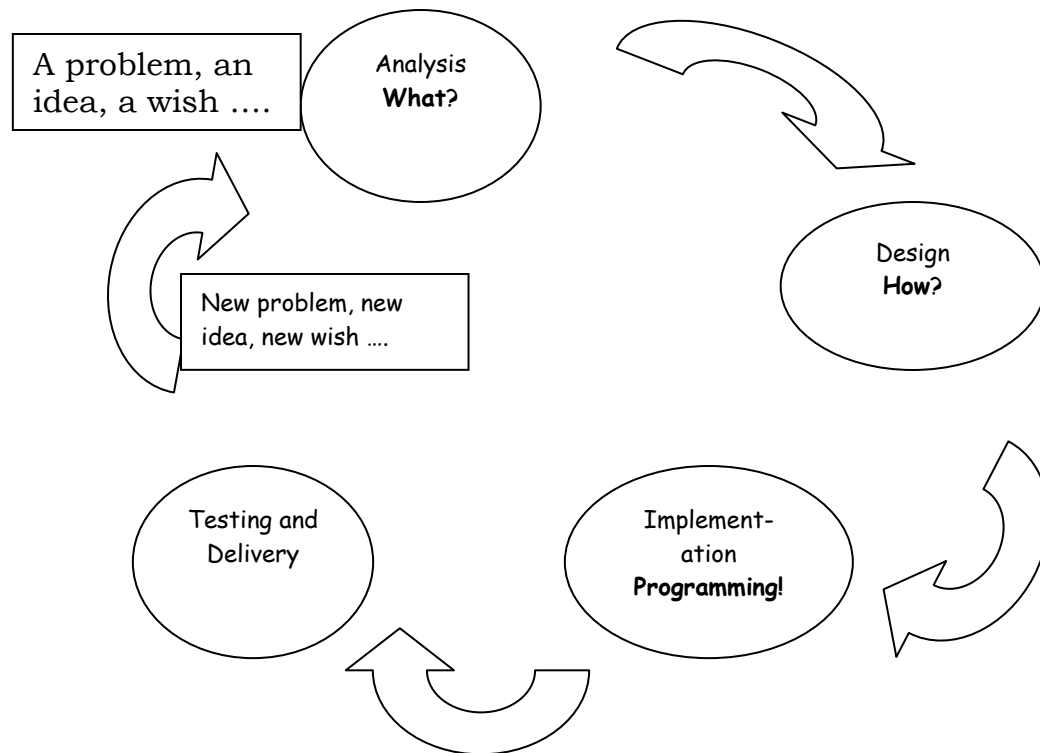
Analys - VAD? I denna fas undersöks uppdraget, och resultatet blir ett svar till VAD som skall göras. Här bestäms både funktionella och icke-funktionella krav. Arbetet resulterar i ett Kravspecifikations dokument (Requirement Specification).

Design - Hur? Under denna fas undersöks olika lösningar för det som har avtalats i kravspecifikationen. Fasen resulterar i ett Design dokument.

Implementation – Gör! Denna del är programmerings del som i bra projekt tar mindre tid än analys och design. Resultatet blir programmerings kod som dokument i denna fas.

Testning och avslutning: Leverera! I denna fas lämnas över projekt till uppdragsgivare efter testning. Som dokument produceras både ett testprotokoll.

Se bilden på nästa sida.



3.1. Identifiera objekt, attribut och operationer

Som en liten del i designfasen, behöver man planera sina klasser. Ofta är det inte så lätt att bestämma de objekt som skall ingå i lösningen och det kräver erfarenhet, men ju mer en jobbar på det desto lättare blir programmeringsarbetet.

En nybörjare har det ännu svårare och därför rekommenderas följande förfarande för att göra det lättare för dem. Utgå ifrån beskrivning av det som skall programmeras. Enkelt uttryckt, är substantiv kandidater för klasser och verb kandidater för metoder.

- 3.1.1. Läs igenom systembeskrivning (t ex beskrivningen ovan) eller ett användarfall, markera substantiven som objektkandidat alternativt attribut.
- 3.1.2. Stryk alla vaga och problemberoende objekt.
- 3.1.3. Ta bort dåliga objekt (alla substantiv kan ej bli objekt), och slå ihop redundanta objekt (synonymer)
- 3.1.4. Identifiera attributen – vissa av de **substantiv** som inte passat som objekt passar kanske bättre som attribut.
- 3.1.5. Läs igen och markera alla **verb** som beskriver operationer.
- 3.1.6. Stryk alla orelevanta operationer.
- 3.1.7. Bestäm vilken operation skall tillhöra vilket objekt.
- 3.1.8. Bestäm relationer mellan objekten; **"Is a"** och **"has a"** är grundprincipen.

3.2. Resultatet

Objekten blir klasser.

Attributen blir instansvariabler i klasser.

Operationerna blir metoder.

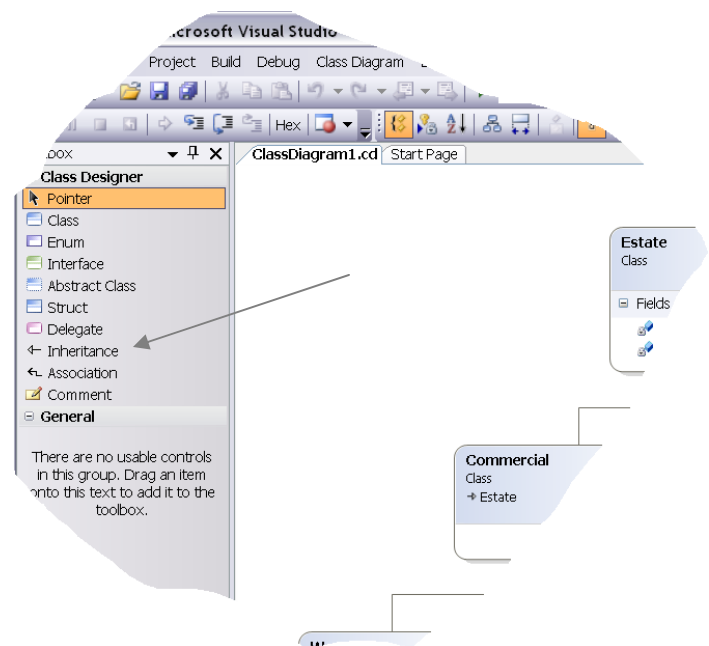
3.3. Skissa designen

- 3.3.1. Starta Visual Studio och skapa ett Windows applikation eller en tom solution. Skapa ett tomt ClassDiagram genom att lägga till en "New Item".
- 3.3.2. Skapa klasserna genom att högerklicka i klassdiagrammet.
- 3.3.3. Skapa relationer mellan dem med hjälp av verktygen i verktygslåda. Kom ihåg att:
 - 3.3.3.1. "är ett" –relation talar för **arv** och
 - 3.3.3.2. "har ett" för **aggregation**
- 3.3.4. Prova olika lösning och välj den bästa.

3.4. Programmera

- 3.4.1. Tänk på
 - 3.4.1.1. bra organisation av projektet,
 - 3.4.1.2. OOP, inkapsling, arv och polymorfism i högsta grad och i minsta detaljer dokumentation

Men i denna övning behöver du inte programmera . Försök att bestämma dig i hundra procent om klassdiagrammet innan du kollar det lösningsförslaget.



4. Lösningsförslag

Ett klassdiagram presenteras i separat fil.

Good Luck!

Programming is fun. Never give up. Ask for help!

Farid Naisan, Course Coordinator and Instructor