

## Beskrivning och reflektion över min kod i uppgift "Left To Do"

Projektet innehåller fem klasser, tre av klasserna är **NormalTask**, **DateTask** och **CheckListaTask** de tar hand om att lägga till ett objekt till olika listor. **NormalKlass** är en **Superklass** som innehåller tre listor av objekt (**TaskUndone**, **TaskDone** och **TaskArchive**) och tre andra variabler (**NumberOfTask**, **Status** och **NameOfTask**). Det finns också tre olika funktioner **AddToDone**, **AddToArchive** och **AddTaskToList**. De andra klasserna **DateTask** och **CheckListTask** ärver alla listor och variabler samt två av funktioner **AddToDone** och **AddToArchive** förutom **AddToTaskList**. Den här funktionen är **Overloading**. Att man lägger till en ny uppgift oavsett vilken typ av uppgift kommer den att lägga till den till **TaskUndone** list. Samma sak gäller om man vill lägga till uppgiften till färdig eller archive. Men det finns en variabel **DateForDays** som finns i **DateTask** klass för att ta hand om resterande dagar. Det finns en annan skillnad som handlar om att när man lägger till en uppgift med **Checklista** kommer programmet att lägga till de till en **Dictionary** som tar nyckeln som string (namn av huvuduppgift) och list av objekt. Det hjälper mig när jag vill lägga till några uppgifter av den här typen av uppgift. Samt hjälper mig när jag vill skriva ut uppgifter.

I samma klass **CheckListTask** finns två viktiga funktioner. En funktion **CheckStatus** som returnerar **True** om alla element i checklistan är färdiga och den andra funktioner tar hand om att flytta alla element till **TaskDone** list. Det finns inte många funktioner i själva klasserna för att jag inte ville använda **System.Console** som Viktor tipsade om. Men jag vill säga att det här är inte min stil. Jag undviker **System.Console** bara när jag använder MVC men inte till exempel i console applikationer.

**Menu** klass är bara en klass som tar hand om olika operationer och beror mest på **System.Console**. **MenuManagement** är såsom **Repository** till **Menu**. För att **Menu** klass var lång och jag är en person som inte gillar att skriva mycket i **Program** klass. Jag har delat upp menyn till små delar och några funktioner som tar hand om olika operationer.

**Encapsulation** beror bara på att skapa privata variabler som man kan nå genom **getter** och **setter**. Funktionen som lägger till **TaskUndone** är **Overloading**. **Inheritance** typ är **Hierarchical** och jag såg ingen anledning till att skapa någon **interface klass** eller **Abstrakt**.

Jag har skapat flera tester. En test där jag skapar objekt av alla typer av uppgifter och tester att lägga till i **TaskUndone** listan och räknar hur

många antal objekt av en specifik uppgift som finns i **TaskUnDone** listan. Jag har skapat en specifik test där jag kontrollerar min funktion **CheckStatus**. Om statusen för alla element för huvuduppgift är **X** så returnerar **True**, annars **False**. Jag har testat den i både situationer **True** och **False**. Jag har skapat andra tester för att kontrollera **addToDoneCheckLista**. Den här funktionen beror på **CheckStatus** funktion, om den returnerar **True** så flyttar elementen till **Done** annars inte. Jag har testat i två olika situationer också.

En annan test var att jag ville veta om huvudnamn för uppgiften sparas i **Dictionary** som **Key**. Jag har fokuserat på den här delen för att både funktionerna **CheckStatus** och **addToDoneCheckLista** var nyckeln för att lösa den här delen av uppgift.