Laboration 3

Uppgift 3.a planering:

Steg 1: Kolla upp önskade metoder att ta bort mellanslag i en sträng och om toUpperCase()/ toLowerCase() finns i C#

Steg 2: skriv och granska planen

Steg 3: utför planen.

Steg 4: granska koden.

Do

//variabler (bool värde för att loopa)

// läs in en text rad, ge en ny sträng variabler och ta bort mellanslag

//Kolla så att det bara är små eller stora bokstäver, bool värde checkar

//om den övre radens krav genomgår så ska en for-loop initieras och göra en ny sträng som har det i den omvända

> //om det är ett palidrom så presentera engligt det, annars presentera att det inskriva inte var ett palidrom.

// While (loopa så länge de inte trycker ESC)

Misstänker att uppgiften kommer ta runt 1½ tim inkluderat med steg 1.

Motivering bakom tids besult: Jag känner att min planering är ganska solid och om jag väl stöter på nått problem så borde det inte vara stort.

Reflektion – Min strategi i denna planering var att göra en karta av programmet sen att ha steg därtill som jag följde.

I slutändan så gick min planering som planerat, jag använde samma princip som jag gjorde i 1.a och 1.b med input[i].

Jag är nöjd med min lösning men nu i efterhand har jag också sett lösningar som var enklare då de aldrig gjorde en ny sträng utan bara jämnförde varje tecken för sig, dvs. det första tecknet med det sista tecknet, det näst första tecknet med det näst sista osv.

Jag behåller givetvis min lösning för jag anser att det har med själva uppgiften att göra. Om jag hade suttit och planerat mer och tänkte lite mer på en optimal lösning

så hade kanske kommit på den med. Har inte mycket mer att säga om denna uppgiften. Den tog 1 tim och 25 min att utföra.

Uppgift 3.b planering:

Klassen:

// privata fält

// egenskaper

//konstruktor, en för 2 parametrar och en för 4.

// metoden add, ta 2 bråk tal och addera dem, ge 2 av egenskaperna de nya värderna.

//metoden multiply, , ta 2 bråk och multiplera dem, ge 2 av egenskaperna de nya värderna.

// isEqualTo, ska jämnaföra om 2 bråk är lika, teori om hur det ska göras läggs på notiser.

//toString, retunerar en stäng med det nya bråket som ska presenteras.

Main program:

Do

Try

Switch

~ ~ ~ ~

~~~~~

~~~~~

~~~~~

Catch

While

Optimerar under tiden jag gör det.

Notiser:

-isEqualTo

Teori: Multiplicera båda täljarna med den andra nämnaren och sen jämföra täljarna. Skriva ut lämpligt medelande? Retunera true / flase och sen ha if, else som fångar värdet?

Misstänkt tid för utförning : 4-5 tim.
~~~~~

Motivation bakom tids planeringen: känner att jag har koll på allt men att det är en stor uppgift med många trådar som ska knytas ihop.

Reflektion – 95% av planen gick som planerat. Hade lite problem när jag skulle hantera utskriften av uträkningen men löste det genom att lägga till en till konstruktor, en för inläsning av 2 parametrar och en för 4. Detta var inget jag fastnade på, utan jag hade bara en fel tänkande i min plan. Jag missade också att sätta set i egenskaperna som inte behöver set:a värderna inom ett visst intervall, blev dock påmind av detta av handledaren.

Stratergin jag hade var att göra en karta över min tankegång sen göra notiser om saker jag var osäker på, vilket i detta fall bara var en sak. Jag framsåg inga andra saker som skulle kunna bli ett stort problem. Hur räkne formerna såg ut dubbelkollade jag innan jag började med uppgiften.

Totalt så tog uppgiften att utföra 3 tim och 30 min.