Projekt 1 dagbok -Frederick Wennborg

**2023-01-05**

Inledde arbetet med projekt 1. Skapade ett nytt project i visual studio och synkroniserade med github. La till feature branch för meny. Fortsatte planeringen av projektet genom att modellera databasen i drawio med respektive tabells kolumner och datatyper.

Reflektion:

Behöver titta mer på branches för att förstå hur jag ska implementera dem effektivt i min arbetsprocess och för att kunna påbörja kodning av projektet. Funderar kring min menystruktur, ska jag göra på liknande sätt som i hotelappen eller går det att strukturera bättre?

**2023-01-06**

Studiedag. Läst på och kollat youtube om branches och hur jag kan använda dem. Gjorde en feature branch för application klass och mergade den med min master branch. Har läst på och kollat youtube om class libraries. Har gjort en skiss på menystruktur som jag färdigställt i drawio.

Reflektion:

Funderar kring hur jag kan använda class libraries för detta projekt. Samtliga delar använder sig av uträkningar, kanske kan någon form av Math/Calculationlibrary kunna vara användbart?

**2023-01-07**

Studiedag. Läst på om dependency injection och strategy pattern.

Reflektion:

Undrar hur jag ska implementera design patterns i detta projekt. Måste jobba vidare med själva kodningen för att stöta på behovet.

**2023-01-08**

Fortsatte läsa på om interfaces, dependency injection och strategy pattern. Färdigställde min mappstruktur i C# via olika meny klasser. Har nu skapat en meny som går att navigera fram och tillbaka.

Reflektion:

Har nu en funktionell meny, är dock inte helt klar med strukturen ännu. Använder just nu ”new”för att skapa nya submeny objekt i min main menu klass. Kanske kan jag använda mig av dependency injection för att undvika tight coupling mellan min Main Menu class och alla sub menu classes?

**2023-01-09**

Har färdigställt alla komponenter som krävs för att skapa min databas med EF. Har lagt till appsettings.json, nuget-packages, applicationdbcontext och alla dbsets. Seedar 4 shapes och lägger till i databasen. Har ändrat menystrukturen så att subklasser binder sig till ett IMenu interface som skickas in i main menu via dependency injection.

Reflektion:

Funderar på hur jag ska fortsätta min projektarkitektur för att undvika tight coupling och dålig underhållbarhet. Kanske att min dbcontext samt CRUD klasser behöver vara kopplade till interfaces för att undvika detta?

**2023-01-10**

Skapat IDbcontext interface och gjort om flödesstrukturen så att den passeras runt med dependency injection till behövande klasser. Skapat CRUD klasser för GeometricResult och implementerat individuella interfaces ICreateResult, IReadResult etc. Dessa matas in via dependecy injection vid initialiseringen av main menu.

Reflektion:

Har försökt få IDbContext att implementera Migrate() men får det inte att fungera. Kan dock utföra SaveChanges(). Interfacet får därmed användas efter DataInitializer är klar. Dependency injection flödet verkar fungera men funderar på om jag har implementerat detta på ett korrekt sätt?

**2023-01-11**

Har lagt till klassbiblioteket och skapat klass för att underlätta felhantering för input. Har också lagt till Controller för mina shapes result som ska inehålla metoder för att assistera skapande av ett result. Fortsatt studerat dependecy injection och strategy pattern.

Reflektion:

Generellt nöjd med processen såhär långt men funderar på mitt klassbibliotek och vad jag ska lägga in för kod där? Ska bara kod relaterat till matematik finnas där eller även andra klasser exempelvis mina Data-klasser? Hur ska strategy pattern implementeras med calculator?

**2023-01-12**

Lektion på Kyh. Har bestämt mig för att göra om strukturen och lägga till 3 nya projekt, ett för varje del i programmet. Skapat en ny refactor branch för att flytta runt all kod. Programmet består nu av ett Classlibrary, ett MainMenu app project, och 3 programApp projects. Fixat till menyflödet så man kan ta sig till varje projekt och dess submeny. Hanterat ett stort bugproblem efter att ha flyttat Data foldern till mitt Class library. Varenda namespace i projektet blev fel och var tvungen att korrigera detta manuellt, resharper kunde inte lösa det.

Reflektion:

Jag har bestämt mig för att tänka om lite och expandera strukturen av Project 1 genom att dela upp det i olika projekt. Detta påverkar då också min EF Code first, då jag behöver ha mina Data-klasser i ClassLibrary istället för i mitt gamla huvudprojekt som nu heter MainMenuApp. Känns mer logiskt att dela upp på detta sättet, men undrar hur jag ska lösa flödet av de nya projekten i min Dependency injection struktur?

**2023-01-13**

Fokuserat på att implementera CreateCalculationResult koden. Har gått igenom StrategyPattern och använt detta för att genomföra mina uträkningar. Jobbat med felhantering; det ska exempelvis inte gå att dela med 0.

Reflektion:

Den grundläggande miniräknaren fungerar bra och det går att spara resultat utan problem. Måste nu tänka igenom hur jag ska göra i update-delen, vilken kod kan jag återanvända eller stoppa i en metod? Har gjort en grundläggande felhantering, men måste utveckla detta mer så att programmet förhindrar ett för stort eller för litet tal(så att programmet kraschar) eller att det går att ta roten ur ett minustal.

**2023-01-14**

Fortsatt jobbat med felhantering, samt implementerat modulus och roten ur. Första version av CreateCalculationResult klar och även UpdateCalculationResult samt ReadCalculationResult. Flyttat min strategypattern till CalculationController så att jag kan använda denna både i Create och Update utan att behöva upprepa samma kod 2 gånger.

Reflektion:

Strategy pattern fungerar som det ska i Calculator. Ska titta på hur jag kan implementera liknande mönster i Create Geometry. I övrigt är Calculator nu nästan klar, bara små detaljer och finjusteringar.

**2023-01-15**

Gjort klart DeleteCalculationResult och förbättrat felhantering vid UpdateCalc. Lagt in strategy classes för GeometryResult. Skapade en DefineGeometryResult metod i min Controller. Flyttat runt Class folders till sitt relevanta projekt.

Reflektion:

Arbetet flyter på bra, känner att jag har god förståelse för hur jag ska få ihop resten av koden. Funderar bara på mina sista concretions; behöver nu skapa ett nytt objekt i Create i alla projekt som jag måste ha för att spara i min dbset, ex. behöver jag göra en new GeometryResult varje gång nån trycker på CreateGeometry. Där fungerar inte dependency injection då den bara skapar ett objekt en gång vid initialiseringen av main menu. Hur ska jag göra för att undvika detta? Kanske en klass som returnerar den klass jag behöver iställer för att använda mig av new? Factory pattern?

**2023-01-16**

Gjort klart min CreateGeometry med strategypattern. Använder mig av en AreaPerimeter class för att lagra area och omkrets efter uträkning och skickar tillbaks denna klass från strategin till huvuddelen av programmet. Fokuserat på att få ihop resten av CRUD-klasserna i Geometrydelen. Tittat mer generellt på programmet och refactored kod till bättre platser och fixat mindre buggar.

Reflektion:

ShapeApp och CalculatorApp mer eller mindre klara. Nu är det endast GameApp. Jag har skrivit ett Rock Paper Scissors program förut som jag tänkte använda som grundkod. Behöver dock anpassa det till min struktur och kanske göra små förändringar i själva spelet.

**2023-01-17**

Har implementerat Rock,Paper,Scissors till GameApp. Spelet fungerar som det ska och sparar till databasen efter använderen väljer att avsluta. Har fokuserat på att få in mer färg i programmet och skapat en ColorService för detta ändamål. Fixat små buggar efter att Malin testkört programmet och kommit med feedback.

Reflektion:

Nu är programmet nästan klart. Ska titta på att få in lite mer färg i de olika appdelarna. Undrade fortfarande över factorydelen. Ska försöka titta på det med imorgon, sista dagen innan muntlig presentation.

**2023-01-18**

Skrivit klart readme.md. Fokuserat på helheten och letat efter eventuella småbuggar. Har implementerat färg i hela projektet och sett till att menyer, inputs, felmeddelanden och resultat har olika färgscheman. Version 1.0 av applikationen klar.

Reflektion:

Hann inte implementera factory pattern i tid, får se över detta vid senare tillfälle. I övrigt nöjd med arbetet.