

Refaktoriserings rapport

Det första jag gjorde var att lyfta ur kod från program.cs till lämpliga klasser som jag har skapat. Skapade även interfaces till klasserna för testning och dependency injection.

Result

Result skapade jag för att sköta all statistik. I den klassen så delade jag ShowTopList till två metoder, Fileformatting och Show för att få bättre skalbarhet.

GameLogic

GameLogic skapade jag för att sköta all logik i spelet och det var två metoder som skötte det. Ändrade string returnen för CheckBC enligt Microsofts förslag.

UserInterface

UserInterface skapade för att hand om det som användaren kommer interagera med. Här använder jag mig av Dependency injection för att injecta GameLogics och Results interfaces.

Delade upp koden i Userinteraction, CheckUserGuess och LengthCheck.

UserInteraction är metoden som hanterar flödet och CheckUserGuess tar hand om användarens input samt tar hand om dess felhantering.

GameController

GameController skapade jag för att skapa separation från Userinterface.

Med IUserinterface injectat i GameController så den får tillgång till UserInterface metoder som kontrollerar flödet.

Gjorde en run-metod som tar instansen av userinterface och kör userinteraction.

Dependency Injection

Dependency injection logiken har jag i program.cs som jag hittade med hjälp av youtube och microsoft dokumentation. Körde allt i en console.app så det fanns inget inbyggt för att använda.

Unit Testing

Unit Testing så använde mig av MSTest som rekommenderades av vår lärare.

Där så testade jag metoderna som tar hand om logiken för programmet dvs CheckBc och MakeGoal.

För CheckBC testade jag ett positivt, negativt och hälften rätt, hälften fel.

För MakeGoal så testade jag att den genererade fyra siffror.