

Reflektionsprotokoll – Labb 2

Vilka problem stötte ni på och hur löste ni dem?

Ett av de största problemen jag stötte på var att programmet kraschade när användaren matar in felaktig data, som bokstäver istället för siffror. Jag hade använt `Convert.ToInt32` och `Convert.ToDouble`, vilket inte hanterade särskilt bra. För att lösa det började jag använda `TryParse` istället, vilket gjorde programmet mer stabilt. Jag hade också skrivit för mycket logik direkt i `Main`-metoden, vilket gjorde koden svår att läsa. Det löste jag genom att bryta ut delar till en egen metod för att göra programmet mer strukturerat och lättare att förstå.

Vilka erfarenheter/lärdomar har ni fått?

Jag har lärt mig hur viktigt det är att dela upp koden i metoder, särskilt när man vill att programmet ska vara överskådligt och lätt att felsöka. Jag har även fått en större förståelse för varför felhantering är viktig, och att man aldrig kan lita på att användaren skriver in rätt saker. `TryParse` var nytt för mig, men jag insåg snabbt hur användbart det är för att göra programmet säkrare. Jag lärde mig också hur `Math.Pow` fungerar och hur man kan spara återkommande beräkningar i variabler för att göra koden effektivare.

Vad har ni använt för att lösa uppgiften?

Jag har använt mig av Rickards föreläsningar och mina egna anteckningar från hans föreläsningar. Utöver det använde jag YouTube och andra online-resurser som t.ex. FreeCodeCamp för att lära mig grunderna i C# och hur man jobbar med metoder och användarinmatning. Jag har även läst olika guider och forumtrådar för att förstå specifika delar, som `TryParse`. Jag pratade också med en kompis som också läser programmering – han hjälpte mig att förstå hur man tänker steg för steg i kod. Jag har jobbat själv med uppgiften, men använt dessa resurser för att lösa problemen på egen hand. Jag använde också ChatGPT ibland när jag fastnade eller behövde enklare förklaringar till nya begrepp.

Om ni har använt generativ AI, som till exempel ChatGPT eller liknande: Vilka frågor ställde ni till AI-verktyget?

Jag använde AI som ett komplement när jag behövde hjälp att förstå vissa saker mer grundläggande, till exempel hur man strukturerar ett program i metoder eller hur man validerar användarinmatning. Det hjälpte eftersom jag fick det lättförklarat. Jag frågade bland annat hur man använder `TryParse` i praktiken, och hur man kan skriva en metod som hanterar användarens input på ett säkert sätt. Jag ställde också frågor om hur `Math.Pow` fungerar.

Hur använde ni det svar ni fick från AI-verktyget?

Jag använde svaren för att först förstå koncepten på en enklare nivå och sedan testade jag själv i Visual Studio för att se om jag kunde få det att fungera. Informationen jag fick stämde även överens med andra källor som FreeCodeCamp. Det blev ett slags stöd för att tänka logiskt och kunna gå vidare när jag kört fast.

Vilken nytta fick ni av svaren från AI-verktyget?

Jag hade nytta av att få en enklare och tydlig genomgång av hur klasser fungerar och hur man kan använda dem i koden på ett mer strukturerat och lättbegripligt sätt, vilket hjälpte mig att bygga vidare på de grundkunskaper jag redan hade med mig.