

Uppgift 3 - Förbättrad planering av programmering

Strategi:

Jag har valt att arbeta enligt Scrum för denna uppgift.

Min planeringsstrategi utöver det som jag har nämnt tidigare är då att arbeta enligt Lean tänkandet och att skriva bra kod från början. Nu ska jag dokumentera projektets gång och så ska vi se om vi kan få uppskattad tid att bli helt perfekt.

Jag kommer dessutom att göra en liten skiss av koden innan jag börjar med kodningen (så att jag har en smidigare "mall" att gå efter). I samband med skissen kommer jag att söka information på områden man inte är så säker på.

Uppgift 3A - Planering

- Uppskattad Tid: 30min
- Uppskattad tid för fel: 10min
- Språk: C#
- Hjälpmedel: Visual studio. StackOverflow. Gamla C# uppgifter.
- Total tid: 40 min

Sprint 1 - (10 Mintuer)

- While loop.
- Fixa fram en textrad som användaren kan skriva in en text.
- If-sats om en användare skriver små eller stora bokstäver.
- Skapa ett felmeddelande (om någon skriver med stora bokstäver)

Sprint 2 - (10 Mintuer)

- Ny chans (ny inmatning)
- Inga stora bokstäver
- Kontrollera om ordet är Palindrom
- If palindrom

Sprint 3 - (10 Mintuer)

- Meddela ett svar om det är ett Palindrom
- Meddelande om det inte är ett Palindrom
- Meddela svaret tebox om

Uppgift 3B – Planering

- Uppskattad Tid: 1h
- Uppskattad tid för fel: 10m
- Språk: C#
- Hjälpmedel: Visual studio. StackOverFlow. Gamla C# uppgifter.
- Total tid: 1h 10m

Scrum

Sprint 1 - (15 Mintuer)

- Skapa klass "Fraction"
- Skapa privata fält för täljare och nämnare (int)
- Skapa konstruktör för ett nytt bråktal
- Skapa metoden getNumerator
- Returnerar täljaren

Sprint 2 - (15 Mintuer)

- Skapa metoden getDenominator
- Om nämnaren är noll
- Felmeddelande
- Returnera nämnaren
- Skapa metoden isNegative
- Returnerar "true" om det är ett negativt bråktal
- Skapa metoden add
- Lägger till

Sprint 3 - (30 Mintuer)

- Skapa metoden multiply
- Skapa metoden isEqualTo()
- Jämför två Fraction-instanser och ser om de representerar samma bråktal.
- Använd toString()
- Returnerar "Täljaren T eller Nämnaren N"

Uppgift 3 - Reflektioner

Uppgift 3A

Vi sa från början att total tiden är 1 timme och 10 min. Då 10 minuter motsvara antal uppkomna fel osv. Då jag har jobbat med Scrum och jobbat inom projektform innan så ska jag ha hyffsat bra koll på vad jag gör. Dock så var detta för 2 år sedan men lite kunskap sitter kvar.

Eftersom jag ärligt talat inte visste vad en palindrom var så tog jag tid för att läsa om det och hur man kan kontrollera om en sträng är en palindrom eller inte. Det jag fick reda på att man kan konvertera den inmatade strängen "bakvänd" och sedan jämföra om den inmatade strängen och den bakvända strängen är lika.

Tidsuppskattning:

- Uppskattad Tid: 1h
- Uppskattad tid för fel: 10m
- Språk: C#
- Hjälpmedel: Visual studio. StackOverflow. Gamla C# uppgifter.
- Total tid: 1h 10m
- Verklig tid: 50min
- Sprint 1: 10min, Sprint 2: 10min, Sprint 3: 30min. Fel: 0min
- Diff: -20min

Problem:

Inga problem i denna uppgift. Uppgiften blev enklare när man strukturerade med upp med sprintar. Dock googlade jag lite på vad Palindrom är. Typ hur man läser kod baklänges?

// Google

Ett **palindrom** är en följd av [skrivtecken](#) som blir likadan oavsett om man läser den framlänges eller baklänges.

Uppgift B

Vi sa från början att total tiden är 1 timme och 20 min. Då 20 minuter motsvara antal uppkomna fel osv. Då jag har jobbat med Scrum och jobbat inom projektform innan så ska jag ha hyffsat bra koll på vad jag gör. Dock så var detta för 2 år sedan men lite kunskap sitter kvar.

Eftersom jag såg på alla metoder som skulle skrivas så visste jag att denna uppgift skulle vara lite mer tanke bakom det hela och lite mer kod som skulle skrivas.

Orsaker till avvikelser:

Inga direkta orsakar eller dylikt. Eftersom jag tyckte jag hade en bra struktur redan från start då jag jobbar enligt Scrum så kan jag gå igenom vad jag kan förbättra till nästa Sprint. Då jag har en Sprintbacklogg som jag tar från min Produktbacklogg. Så jag har stor koll från början.

